

Journal of Advanced Transportation

Intelligent Autonomous Transport Systems Design and Simulation

Lead Guest Editor: Cheng S. Chin

Guest Editors: Xionghu Zhong, Mohammad Hamdan, Rongxin Cui,
Juan C. Cano, and Jose R. Martinez-De-Dios





Intelligent Autonomous Transport Systems Design and Simulation

Journal of Advanced Transportation

Intelligent Autonomous Transport Systems Design and Simulation

Lead Guest Editor: Cheng S. Chin

Guest Editors: Xionghu Zhong, Mohammad Hamdan,
Rongxin Cui, Juan C. Cano, and Jose R. Martinez-De-Dios



Copyright © 2018 Hindawi. All rights reserved.

This is a special issue published in "Journal of Advanced Transportation." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

Francesco Bella, Italy
Abdelaziz Bensrhair, France
María Calderon, Spain
Juan C. Cano, Spain
Giulio E. Cantarella, Italy
Anthony Chen, USA
Steven I. Chien, USA
Antonio Comi, Italy
Emanuele Crisostomi, Italy
Luca D'Acierno, Italy
Andrea D'Ariano, Italy
Alexandre De Barros, Canada
Luigi Dell'Olio, Spain
Cédric Demonceaux, France
Sunder Lall Dhingra, India
Yuchuan Du, China
Juan-Antonio Escareno, France

David F. Llorca, Spain
Jérôme Haïri, France
Serge Hoogendoorn, Netherlands
Ángel Ibeas, Spain
Lina Kattan, Canada
Victor L. Knoop, Netherlands
Alain Lambert, France
Ludovic Leclercq, France
Seungjae Lee, Republic of Korea
Zhi-Chun Li, China
Yue Liu, USA
Jose R. Martinez-De-Dios, Spain
Monica Menendez, USA
Rakesh Mishra, UK
Andrea Monteriù, Italy
Giuseppe Musolino, Italy
Jose E. Naranjo, Spain

Aboelmaged Noureldin, Canada
Dongjoo Park, Republic of Korea
Paola Pellegrini, France
Luca Pugi, Italy
Nandana Rajatheva, Finland
Prianka N. Seneviratne, Philippines
Wai Yuen Szeto, Hong Kong
Richard S. Tay, Australia
Martin Trépanier, Canada
Pascal Vasseur, France
Antonino Vitetta, Italy
Francesco Viti, Luxembourg
S. Travis Waller, Australia
Chien-Hung Wei, Taiwan
Yair Wiseman, Israel
Jacek Zak, Poland
G. H. de Almeida Correia, Netherlands

Contents

Intelligent Autonomous Transport Systems Design and Simulation

Cheng Siong Chin , Xionghu Zhong, Mohammad Hamdan, Rongxin Cui , Juan C. Cano ,
and Jose R. Martinez-de Dios

Volume 2018, Article ID 1468040, 2 pages

An Eco-Driving Advisory System for Continuous Signalized Intersections by Vehicular Ad Hoc Network

Wei-Hsun Lee  and Jiang-Yi Li

Volume 2018, Article ID 5060481, 12 pages

Design Method of ADAS for Urban Electric Vehicle Based on Virtual Prototyping

Katarzyna Jezierska-Krupa  and Wojciech Skarka 

Volume 2018, Article ID 5804536, 19 pages

Neural Network Control System of UAV Altitude Dynamics and Its Comparison with the PID Control System

Jemie Muliadi and Benyamin Kusumoputro 

Volume 2018, Article ID 3823201, 18 pages

State-of-Charge Estimation and Active Cell Pack Balancing Design of Lithium Battery Power System for Smart Electric Vehicle

Z. C. Gao, C. S. Chin, W. D. Toh, J. Chiew, and J. Jia

Volume 2017, Article ID 6510747, 14 pages

Autonomous Path Planning for Road Vehicles in Narrow Environments: An Efficient Continuous Curvature Approach

Domokos Kiss and Gábor Tevesz

Volume 2017, Article ID 2521638, 27 pages

Bifurcation of Lane Change and Control on Highway for Tractor-Semitrailer under Rainy Weather

Tao Peng, Zhiwei Guan, Ronghui Zhang, Jinsong Dong, Kening Li, and Hongguo Xu

Volume 2017, Article ID 3506053, 19 pages

Liveness-Based RRT Algorithm for Autonomous Underwater Vehicles Motion Planning

Yang Li, Fubin Zhang, Demin Xu, and Jiguo Dai

Volume 2017, Article ID 7816263, 10 pages

Reactive Path Planning Approach for Docking Robots in Unknown Environment

Peng Cui, Weisheng Yan, and Yintao Wang

Volume 2017, Article ID 6716820, 11 pages

Consensus Based Platoon Algorithm for Velocity-Measurement-Absent Vehicles with Actuator Saturation

Maode Yan, Ye Tang, Panpan Yang, and Lei Zuo

Volume 2017, Article ID 8023018, 8 pages



Experimental Autonomous Road Vehicle with Logical Artificial Intelligence

Sergey Sergeevich Shadrin, Oleg Olegovich Varlamov, and Andrey Mikhailovich Ivanov

Volume 2017, Article ID 2492765, 10 pages

A Variable Interval Rescheduling Strategy for Dynamic Flexible Job Shop Scheduling Problem by Improved Genetic Algorithm

Lei Wang, Chaomin Luo, and Jingcao Cai

Volume 2017, Article ID 1527858, 12 pages

Motion Planning for Omnidirectional Wheeled Mobile Robot by Potential Field Method

Weihaio Li, Chenguang Yang, Yiming Jiang, Xiaofeng Liu, and Chun-Yi Su

Volume 2017, Article ID 4961383, 11 pages

Editorial

Intelligent Autonomous Transport Systems Design and Simulation

Cheng Siong Chin ¹, **Xionghu Zhong**,² **Mohammad Hamdan**,³
Rongxin Cui ⁴, **Juan C. Cano** ⁵, and **Jose R. Martinez-de Dios**⁶

¹Newcastle University in Singapore, Ngee Ann Polytechnic, SIT Building, Singapore 599493

²Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

³Heriot-Watt University, Dubai Campus, Dubai, UAE

⁴Northwestern Polytechnical University, Xi'an 710072, China

⁵Universitat Politècnica de València, Camí de Vera, s/n, 46022 València, Spain

⁶University of Seville, Calle San Fernando 4, 41004 Sevilla, Spain

Correspondence should be addressed to Cheng Siong Chin; cheng.chin@ncl.ac.uk

Received 8 February 2018; Accepted 8 February 2018; Published 11 March 2018

Copyright © 2018 Cheng Siong Chin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The automation transport system is essential in increasing the communication, control, and level of autonomy of a broad range of applications in land, sea, and air. The increase in workforce age and cost has driven the technological development and adaptation of innovative system design, simulation, and application of intelligence in transport systems. The changing technological demand and operating uncertainties have resulted in new types of transportation challenges and increased demand for sensors and sensor platforms for capturing mobility analytics under uncertain operating environment. For example, various intelligent autonomous systems such as unmanned aerial and submersible vehicle for traffic survey, marine sediment transport, and vehicle accident survey are used. For land transport system, the autonomous vehicles such as Google autonomous car and Tesla are focusing on the vehicle's self-information and are developed based on the Light Detection and Ranging (LiDAR) systems, radar, and cameras. The Vehicular-to-Everything (V2X) communication technology enables cars to communicate with each other and with surrounding infrastructure access points like roadside units (RSUs). It can provide global information of the networked vehicles and has drawn extensive attention due to its broad application prospects including advanced driver assistance systems, enhanced collision avoidance, and

automatic toll fee collection. In all these developments toward intelligent autonomous transport systems, there are apparently a lot of significant problems and challenges in system design and simulation that must be faced. Efficient system design and simulation using artificial intelligence techniques could help to address such challenging issues in the intelligent autonomous transport systems. In this particular issue on intelligent autonomous transport systems design and simulation, we have invited few papers that address such issues.

One paper of this special issue addresses ecodriving advisory system (EDAS) to reduce CO₂ emissions and energy consumption by letting the continuous vehicle pass through multiple intersections with the minimum possibilities of stops. Two strategies including maximized throughput model (MTM) and smooth speed model (SSM) were designed and compared with maximized throughput model (MaxTM), open traffic light control model (OTLCM), and predictive cruise control (PCC) models. Another paper introduced a novel method for obtaining good quality paths for autonomous road vehicles in narrow environments. The approach consists of a global planner which generates a preliminary path consisting of straight and turning-in-place primitives and a local planner to make the preliminary path feasible to car-like vehicles. The approximation process with

the proposed local planner was proven to be convergent for any preliminary global paths.

One of the papers presented a method to analyze the nonlinear dynamics and stability in lane changes on highways for tractor-semitrailer under rainy weather. The five degrees of freedom mechanical model with nonlinear tire based on vehicle test was used in the lane change simulation on low adhesion coefficient road. A feedback linearization controller combined with Active Front Steering (AFS) and DYC (Direct Yaw Control) was used to eliminate bifurcations and to improve lateral stability of tractor-semitrailer, during lane changing on the highway under rainy weather. Another paper investigated the vehicle platoon problems where the actuator saturation and absent velocity measurement were taken into consideration. A novel algorithm, where a smooth function was introduced to deal with the sharp corner of the input signals, was proposed for a group of vehicles with actuator saturation by using the consensus theory. A control strategy for the vehicle platoon with actuator saturation and absent velocity measurement was successfully designed via the adaptive control approach.

Another paper presented a unique method for designing advanced driver assistance systems (ADAS) to minimize the costs of the design phase and system implementation and, at the same time, to maximize the positive effect the system has on driver and vehicle safety. The described method was based on using a virtual prototyping tool to simulate the system performance in real life situations. The approach enabled an iterative design process that resulted in a reduction of errors with almost no prototyping and testing costs. It leads to another paper on the use of the genetic programming which was proposed to give good results in the field of tasks scheduling in the design phase. In the improved genetic algorithm, a mix of random initialization population by combining initialization machine and initialization operation with random initialization was designed for generating high-quality initial population. One of the papers tried to solve the problem associated with the precise prediction of the dynamic trajectory of an autonomous vehicle. A Mivar expert system was integrated into the control system of the experimental autonomous vehicle. The system was made more flexible and efficient by the introduction of hybrid artificial intelligence with logical reasoning for an autonomous transport vehicle being involved in a collision.

One of the papers described a new rapidly exploring random tree (RRT), that is, liveness-based RRT (Li-RRT) to address autonomous vehicles motion. Different from typical RRT, an index of each node in the random searching tree named "liveness" to describe the potential effectiveness of the expanding process was used. The expected time of returning a valid path with Li-RRT was improved to verify the efficiency of the algorithm. Another paper used potential field method to navigate a three omnidirectional wheels' mobile robot to avoid obstacles. The potential field method was used to overcome the local minima problem and the goals unreachable with obstacles nearby (GNRON) problem. Also, model predictive control (MPC) was used to incorporate motion constraints to make the velocity more realistic and flexible in the mobile robot.

Another paper applied a comparative method to assess the performance of artificial neural network's (ANN) direct inverse control (DIC-ANN) with the proportional-integral-derivative (PID) control system. The comparison served as an analysis tool to assess the advantages of DIC-ANN over conventional control method for an unmanned aerial vehicle (UAV) attitude controller. The DIC-ANN performed learning mechanism to overcome the limitation of PID tuning. Another paper presented a reactive path planning approach named Dubins-APF (DAPF) to solve the path planning problem for docking in an unknown environment with obstacles. The Dubins curves were combined with the designed obstacle avoidance potential field for planning a possible pathway. The robot was navigated to the docking station with proper pose via the DAPF approach.

One of the papers presented an integrated state-of-charge (SOC) estimation model and active cell balancing of a 12-cell lithium iron phosphate (LiFePO₄) battery power system for an electric vehicle (EV). It is vital as energy storage system is present in all autonomous transport systems. The strong tracking cubature extended Kalman filter (STCEKF) gave an accurate SOC prediction compared to other Kalman-based filter algorithms. The proposed GroupWise balancing of the multiple SOC exhibited a higher balancing speed and lower balancing loss than other cell balancing designs for EV under varying the ambient temperature. In summary, it is imperative that we continue to progress in our search for better intelligent autonomous transport systems design and simulation. The progress reported in this special issue suggests that achieving these aims is an unattainable one.

*Cheng Siong Chin
Xionghu Zhong
Mohammad Hamdan
Rongxin Cui
Juan C. Cano
Jose R. Martinez-de Dios*

Research Article

An Eco-Driving Advisory System for Continuous Signalized Intersections by Vehicular Ad Hoc Network

Wei-Hsun Lee  and Jiang-Yi Li

Department of Transportation and Communication Management Science, National Cheng Kung University, No. 1, University Road, Tainan City 701, Taiwan

Correspondence should be addressed to Wei-Hsun Lee; leews@mail.ncku.edu.tw

Received 20 May 2017; Accepted 15 January 2018; Published 5 March 2018

Academic Editor: Mohammad Hamdan

Copyright © 2018 Wei-Hsun Lee and Jiang-Yi Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the vehicular ad hoc network (VANET) technology which support vehicle-to-vehicle (V2V) and vehicle to road side unit (V2R/R2V) communications, vehicles can preview the intersection signal plan such as signal countdown message. In this paper, an ecodriving advisory system (EDAS) is proposed to reduce CO₂ emissions and energy consumption by letting the vehicle continuously pass through multiple intersections with the minimum possibilities of stops. We extend the isolated intersection model to multiple continuous intersections scenario. A hybrid method combining three strategies including maximized throughput model (MTM), smooth speed model (SSM), and minimized acceleration and deceleration (MinADM) is designed, and it is compared with related works maximized throughput model (MaxTM), open traffic light control model (OTLCM), and predictive cruise control (PCC) models. Some issues for the practical application including safe car following, queue clearing, and gliding mode are discussed and conquered. Simulation results show that the proposed model outperforms OTLCM 25.1%~81.2% in the isolated intersection scenario for the CO₂ emissions and 20.5%~84.3% in averaged travel time. It also performs better than the compared PCC model in CO₂ emissions (19.9%~31.2%) as well as travel time (24.5%~35.9%) in the multiple intersections scenario.

1. Introduction

Greenhouse gases (GHG) are recognized as the main cause of the global warming, with CO₂ being the primary GHG emitted through human activities. Studies by the IEA [1, 2] show that transport was responsible for 23% of world CO₂ emissions in 2014, as shown in Figure 1(a). Within this, the fastest growth in emissions has been from the road transport sector, as shown in Figure 1(b), which increased by 64% since 1990 and accounted for about three-quarters of transport-related emissions in 2014 [1]. In other words, encouraging environmentally friendly driving practices (eco-driving, hereafter) by eliminating unnecessary vehicle acceleration and braking would reduce fuel consumption and CO₂ emissions and contribute to slowing down global warming. The development of eco-driving systems is thus attracting strong interest from academia, vehicle industry, and governments.

A typical driving trip consists of idling, accelerating, cruising, and decelerating, and the related CO₂ emissions depend on changes in driver behavior, road geometry, or

traffic congestion [3]. Barth and Boriboonsomsin show that when a vehicle is idling it consumes more fuel and emits more exhaust fumes than when it cruises at a steady speed [3]. Similarly, Frey et al. [4] show that accelerating and decelerating cause more emissions than idling. In other words, avoiding unnecessary engine idling periods and stops and optimizing driving speeds will reduce CO₂ emissions and fuel consumption. An eco-driving advisory system (EDAS), which provides smooth driving suggestion according to current traffic dynamics and traffic signal plan, can help drivers travel in an environmentally friendly manner.

With vehicular ad hoc network (VANET), also named as connected vehicle technology, vehicles equipped with an on-board unit (OBU) can communicate with road side units (RSU) by vehicle-to-roadside (V2R/R2V) to preview the traffic signal plans and obtain real-time information such as waiting queue length and signal countdown message. Forward collision warning (FCW), which can detect the distance relative to the vehicle ahead, facilitates vehicle safety driving assistance such as adaptive cruise control (ACC). Assuming

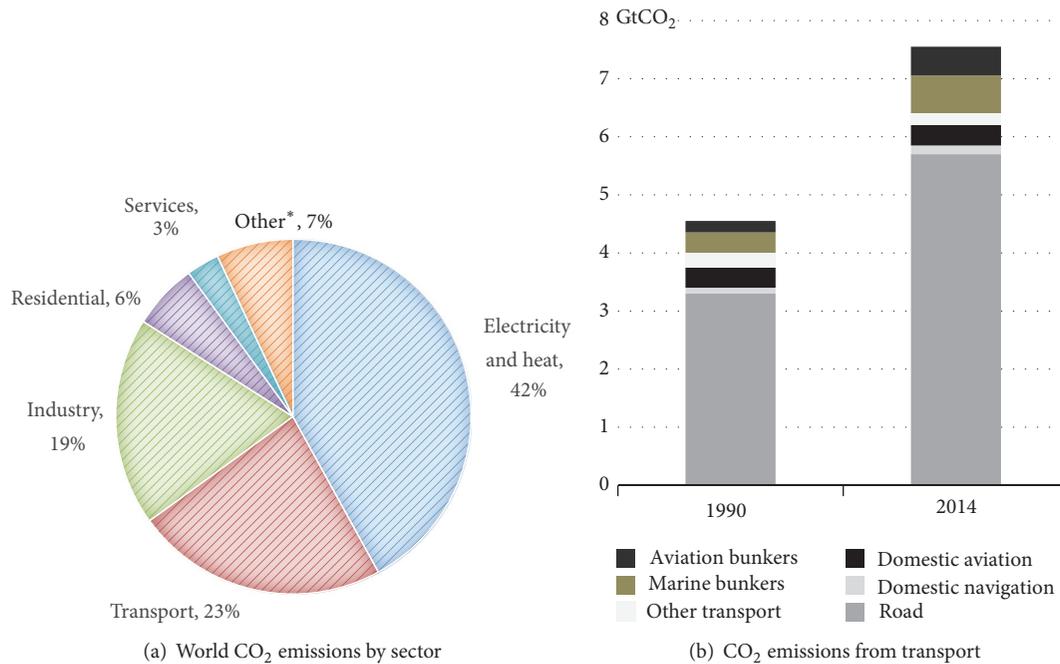


FIGURE 1: Statistics on CO₂ emissions produced by the IEA [1].

that vehicles are equipped with VANET and FCW, OBU can obtain real-time information and calculate the optimal eco-speed to cross an intersection. Our previous work [5] proposed two decision tree based eco-driving suggestion models for an isolated signalized intersection, using OBU to calculate the best eco-driving speed based on real-time information including the traffic signal countdown messages and waiting queue length broadcasted by RSU. These two models are called MaxTM (maximize throughput model) and MinADM (minimize acceleration and deceleration model); simulation results show that MaxTM outperforms MinADM and Open Traffic Light Control Model (OTLCM) [6], being 5% to 102% better than MinADM and 13% to 209% better than OTLCM with regard to CO₂ emissions in the simulation cases, and 8% to 14% better than MinADM and 15% to 231% better than OTLCM in the real traffic cases [5].

MaxTM, MinADM [5], and OTLCM [6] are applied for the isolated intersection scenario; however, in practice vehicles may pass a number of intersections when traveling from origin to destination. In such cases, the isolated intersection eco-driving model may not always be appropriate for use with multiple intersections, since OBU does not acquire the necessary information from the next RSUs, and thus these models cannot provide an optimized speed for the sequential intersections. For example, in Figure 2(a) the vehicle may drive at an unnecessarily high speed if it applies the isolated EDAS model, and it has to stop at the second intersection as it does not have the traffic signal countdown message for the next intersection, as illustrated in Figure 2(b). With the real-time information of the traffic signal timing plans of the sequential intersections, it is possible to design a better eco-driving model to avoid these energy wasting cases, thus lowering fuel consumption and CO₂ emissions.

In this paper we extend our previous works [5, 7] to develop an EDAS that help drivers to avoid unnecessary acceleration, braking, engine idling periods, and stops and to optimize the driving speed in a continuous multiple intersections scenario. Several studies in the literature discuss the multiple intersection scenario. Alsabaan et al. proposed an EEFG model [8] by using decision tree to measure the optimal speed to pass multiple intersections. They then improved this in 2013 [9] by dividing the road region into two parts, one passable and the other unpassable. Asadi and Vahidi [10] use a predictive cruise control (PCC) algorithm in a mathematical programming model to calculate the most economic cruise speed without changing the velocity to pass through sequential intersections. Similar to the underlying concept of the MaxTM, the speed suggested by PCC [10] can be as high as the free flow speed in order to maximize the throughput in conditions when this is possible. The results of their work show that the proposed model can reduce fuel consumption by up to 47% compared to the baseline model. Katsaros et al. [11] proposed GLOSA (Green Light Optimized Speed Advisory), which combines GPS information with common vehicle sensors to obtain more accurate road slope estimates, which are then used to optimize the fuel consumption of the vehicle, thus achieving a 7% reduction in fuel consumption comparing to the baseline. In summary, while a number of works have proposed eco-driving advice systems for multiple intersections [8–11] they are all insufficient or inefficient for practical applications due to the following issues not fully considered.

(1) *Safe Car Following*. Previous studies either do not consider the issue of safely following cars [6, 8, 11] or following a car with a fixed time gap [5]. This approach may result

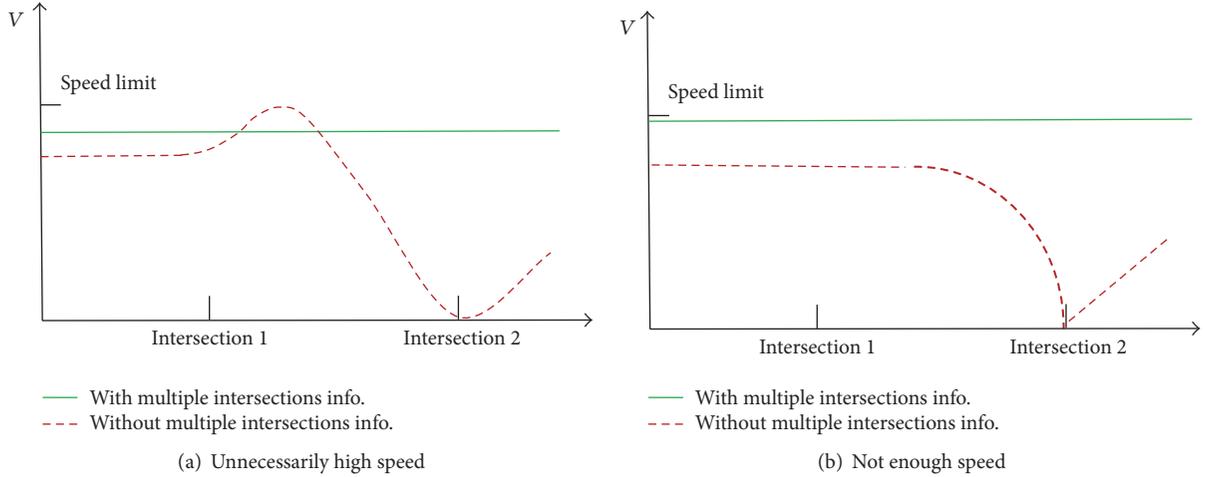


FIGURE 2: Comparison of the system with and without multiple intersection data.

in wasting road capacity at low speed conditions and may cause accidents at high speed ones. A dynamic time gap mechanism, based on the relative speed and distance of the preceding vehicle and the host vehicle, will increase the usage of road capacity.

(2) *Queue Clearing*. When vehicles are stopped by the traffic signals, the time needed for the waiting queue dissipating cannot be neglected, especially when we aim to enable a vehicle to pass smoothly through an intersection without stopping. In related works [6, 8–11], this issue is not considered so that the model would not be practical for realistic applications. Waiting queue dissipating time, which is related to the time needed from current time to the last vehicle in the waiting queue passing through the vehicle, must be considered. When the issue of queue clearing is considered, the resulting eco-driving model will be more accurate and of greater practical use.

(3) *Gliding Mode*. Only three vehicle moving control modes (acceleration, deceleration, and maintaining current speed) are applied in the traditional models [6, 8–11]. However, vehicle gliding mode (free from gas pedal) leverages the engine brake to slow the vehicle without additional fuel consumption, thus causing less CO_2 emissions than these control modes. When the traffic signal plans are known in advance, vehicles can apply gliding to slow down to the suggestion speed.

In contrast to previous works, we take the above issues into consideration and propose more realistic and accurate suggestions with regard to the optimal eco-driving speed. Safety is the major concern so that the proposed model must follow the safe car following rule, speed limit, and traffic signal regulations. Two eco-driving strategies are proposed and compared in the continuous intersections scenario, namely, the maximized throughput model (MTM) and smooth speed model (SSM). These are illustrated in Figure 3, in which the green time window upper bounded by $S_H(i)$ and lower bounded by $S_L(i)$ (both in red dotted line) are

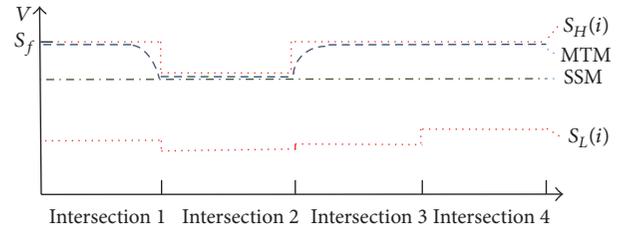


FIGURE 3: Comparison of two strategies: MTM and SSM.

first discovered in four continuous signalized intersections. The MTM strategy (blue dashed line) keeps the suggested speed as high as possible to avoid blocking the following vehicles, thus maximizing the throughput. On the other hand, the SSM strategy (green dashed line) adopts the smooth moving speed upper bound to avoid unnecessary acceleration and deceleration.

The rest of this paper is organized as follows. The assumptions, terminologies, and VANET protocol are presented in detail in Section 2. The algorithms used in MTM and SSM are proposed in Section 3. Two simulation experiments are then discussed in Section 4, including an isolated intersection scenario and multiple intersections scenario, and the proposed system models are compared with the PCC [10], OTLCM [6], and MaxTM [5] strategies. Finally, Section 5 concludes this paper and presents some directions for future works.

2. Eco-Driving System Protocol

Assume that each traffic signal controller is equipped with an RSU, and each vehicle is equipped with an OBU which integrates a location module (ex. GPS) and FCW module to detect the speed and distance of the front vehicle. With these capabilities, the main purpose of the OBU is to calculate the recommended speed (S_r) and provide driving suggestions (speed up/maintain speed/slow down/glides) to pass through as many intersections as possible by considering all the collected real-time information.

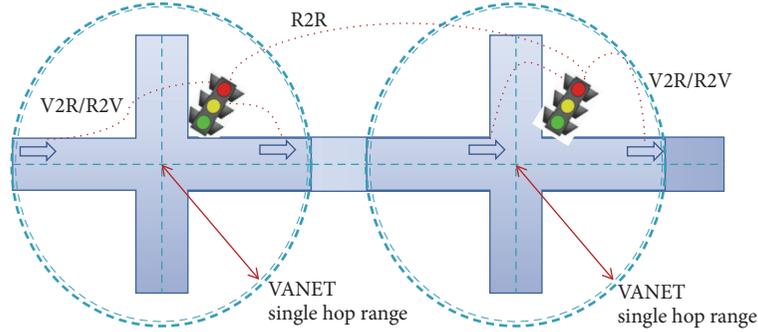


FIGURE 4: Eco-driving advisory system communication schematic.

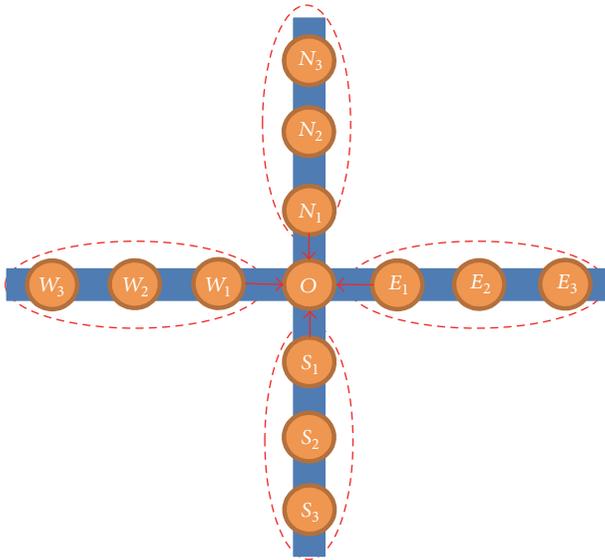


FIGURE 5: RSU-to-RSU information exchange schematic.

The EDAS model is designed on top of the VANET environment with V2R/R2V communication, as illustrated in Figure 4. OBU can collect RSU broadcasted information beyond the range of visual contact. RSU-to-RSU (R2R) communication among the neighborhood RSUs is also included in the system protocol to exchange the signal timing plan and waiting queue length information. As illustrated in Figure 5, every RSU (R_O) will exchange real-time information with three neighborhood RSUs (R_{Di}) in each direction, including data on the waiting queue, signal phase timing plan, road length, and traffic conditions.

The communication protocol of EDAS is designed as illustrated in Figure 6, which is composed of six steps of R2R, V2R, and R2V communication, as listed in Table 1 and explained as below.

Step 1 (R2R exchange basic configuration and signal plan). Each RSU, as the host RSU itself, will exchange its information with twelve neighborhood RSUs. The exchanged information, as shown in (1), is organized as two parts including basic configuration and signal plan. Basic configuration of an RSU (R_s), as defined in (1), includes RSU ID (R_i), location

TABLE 1

Parameters	Value
Road length (m)	200 m; 400 m; 600 m
Traffic volume (vph)	50; 100; 200; 400; 800
Traffic signal (sec) (green/red)	30/30; 45/45; 60/60
Spatial depth	Four intersections
Temporal depth	Four cycles
VANET range (m)	200
Cruise gliding acceleration	-0.15 m/s^2
Idle gliding speed (S_{\min})	2 m/s

(X, Y), road length to the next intersection in four directions ($L(n, s, e, w)$), and BSS (basic service set) identification in 802.11p (SSID). Traffic signal plan and current traffic information (R_p), as defined in (2), consisted of cycle time (T_p), main direction (north-south) green split (T_g), current phase (P_c , 0 indicates currently signal is green in north-south direction and red in east-west direction and 1 indicates the reverse case), and countdown remaining seconds in north-south and east-west direction ($R(n-s, e-w)$). With the above definitions, the R2R data exchange protocol is defined in (3), which includes RSU configuration (R_s), traffic signal plan (R_p), and timestamp (T_s).

$$R_s = \{R_i, X, Y, L(n, s, e, w), \text{SSID}\} \quad (1)$$

$$R_p = \{T_p, T_g, P_c, R(n-s, e-w)\} \quad (2)$$

$$\text{R2R} : \{R_s, R_p, T_s\}. \quad (3)$$

Step 2 (RSUs broadcast synchronized message to OBUs). Each RSU periodically broadcasts message to notify all the vehicles (OBUs) with the range of its transmissions. These messages are used to synchronize the OBUs and wake them up if they are not currently in working status. The message format is shown in (4), including broadcast RSU information (R_s) and timestamp for time synchronization. When an OBU receives one or more RSU broadcasted messages from different RSUs, it first decides which one is the host OBU based on the relative location and moving direction and computes the recommended speed using its current location,

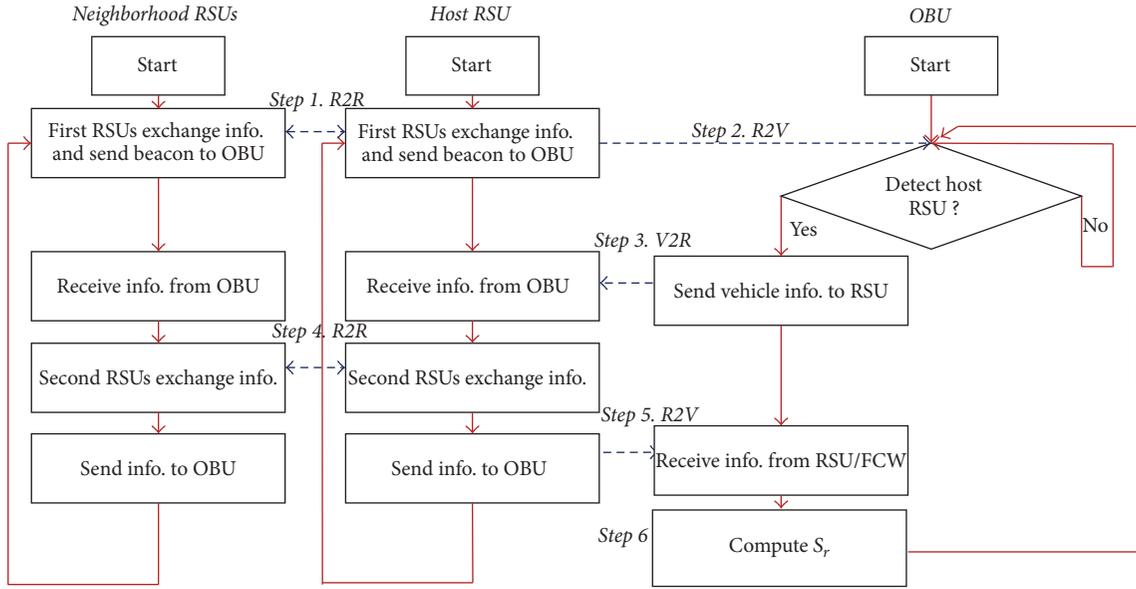


FIGURE 6: EDAS communication protocol.

speed, front vehicle speed and distance (from FCW), and the information collected from the host RSU, as explained in the next section.

$$R2V : \{R_s, T_s\}. \quad (4)$$

Step 3 (OBUs send information to RSUs). After receiving a synchronized message from the host RSU, an OBU will be in active mode and periodically broadcast information including OBU ID (O_i), current speed (S_c), moving direction (S_d), current acceleration (A_c), and position (X, Y), as shown in (5). The OBU also starts to detect the distance and speed of the front car via FCW at the same time.

$$V2R : \{O_i, S_c, S_d, A_c, X, Y\}. \quad (5)$$

Step 4 (RSUs exchange dynamic traffic information collected from OBUs). An RSU exchanges dynamic traffic information with the neighborhood RSUs in this step, calculating the waiting queue length in each directions by estimating the vehicles' positions using the collected OBU information. Each host RSU updates the neighborhood RSUs array each time an RSU exchange event occurs. The exchange message format is defined in (6), including RSU ID (R_i), current phase and signal plan (R_p), waiting queue length in each direction ($Q(n, s, e, w)$), and timestamp T_s .

$$R2R : \{R_i, R_p, Q(n, s, e, w), T_s\}. \quad (6)$$

Step 5 (RSUs broadcast dynamic signal message to OBUs). In this step, the summarized information of all the RSUs, including the RSU itself (R_o) and the twelve neighborhood RSUs ($R_{d,i}$), organized as an array as shown in (7), is periodically broadcasted to OBU. The information includes the RSU basic information (R_s), traffic signal timing plan (R_p)

and waiting queue (Q) of host as well as neighborhood RSUs, and timestamp T_s .

$$R2V : \{R_o [R_s, R_p, Q(n, s, e, w)], \quad (7)$$

$$R_{d,i} [R_s, R_p, Q(n, s, e, w), T_s\}.$$

Step 6 (OBU calculates recommended speed). To calculate the suggested eco-speed, the OBU integrates the collected RSUs array information broadcasted in Step 5, GPS status (location, speed, and moving direction), and front vehicle information, including S_p (speed of front vehicle), and D_p (distance to the front vehicle) from FCW. The OBU itself repeats this step several times based on dynamic GPS status and FCW information until the new R2V packet arrives.

3. Eco-Driving Advisory System

With the EDAS protocol, OBU can collect the up-to-date information about upcoming traffic signals including RSU locations, road segment length, and traffic signal timing plan, and real-time traffic information, such as the waiting queue length in each direction for each subsequent RSU, and vehicle flow information collected from the host RSU. The goal of the EDAS is to calculate the optimal speed and suggest driving actions to maximize the possibility of moving through the subsequent intersections without stopped or waiting behind the signal. By comparing the current moving speed, the suggested speed will be converted into suggested driving actions such as speed up/brake/maintain speed/glide and can be presented in colored LED bar-like graphical user interface.

Two critical decisions need to be made before calculating the recommended speed (S_r) based on the collected real-time information, and then the three eco-driving strategies are applied, as shown in Figure 7. The suggested driving

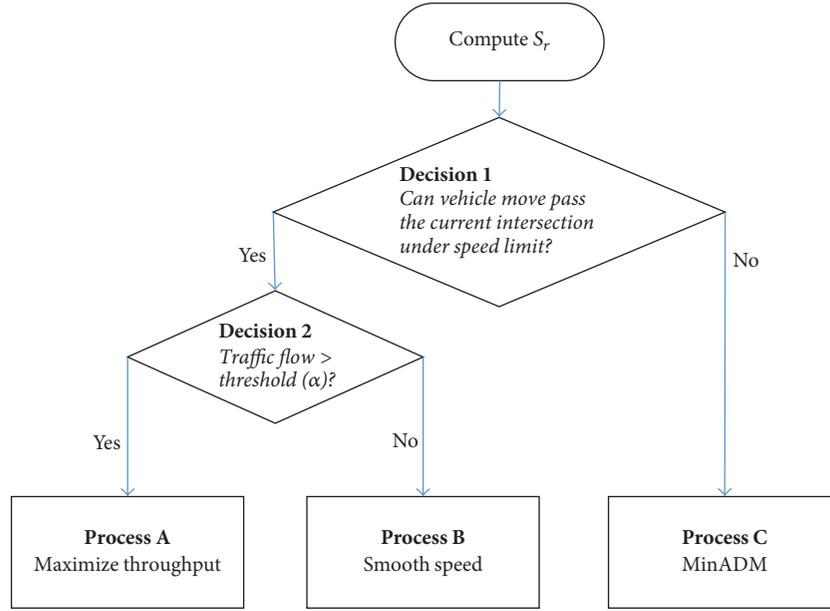
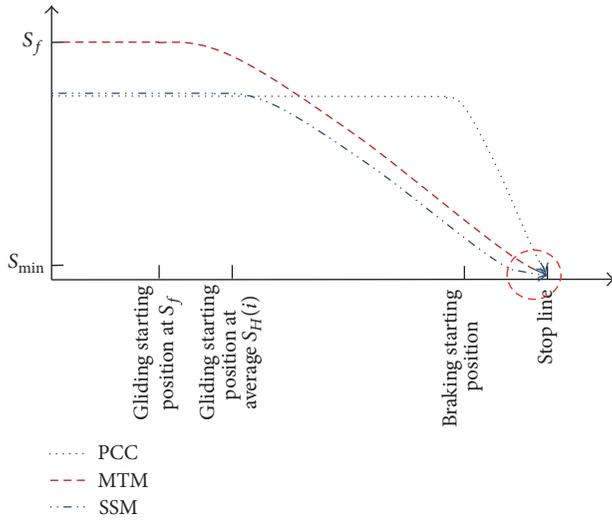
FIGURE 7: Three strategies are applied in computing S_r .

FIGURE 8: Eco-driving strategies comparison of the three models.

behaviors of the proposed eco-driving strategies MTM and SSM as well as the comparative eco-driving model PCC [10] are illustrated in Figure 8. The major difference between the MTM, SSM, and PCC [10] is that the proposed models apply vehicle gliding concept to leverage the engine brake. MTM adopts free flow speed (S_f) and thus applies gliding earlier than the SSM, which adopts smooth speed at upper bound of the speed range ($S_H(i)$). On the other hand, PCC [10] has to brake the vehicle in front of the stop line, where the averaged acceleration is much larger than gliding mode. The three modes, two decision criteria, and some issues about eco-driving are discussed in detail below.

Decision 1: Isolated or Multi-Intersections Mode. The top decision node determines if a vehicle should apply multi-intersection mode or isolated intersection mode by calculating

whether a vehicle can pass through the current intersection under its current conditions with the appropriate speed limit. The decision goes to the left subtree if the answer to Decision-1 is “Yes,” and otherwise it goes to the right subtree, where a modified isolated intersection model, the simplified MinADM, will be applied.

To find out whether a vehicle can pass the intersection under current traffic signal phase P_c (green (0), red (1)), phase remain time ($R(n-s, e-w)$), we can check if the speed needed for the vehicle to move the distance D_i by time period G_t falls into the speed limit range $[S_f, S_{\min}]$, as shown in (10). G_t is defined as the remaining green time if the current signal is green, and as the remaining red time plus the queue clearing time (T_Q) if the current signal is red, as illustrated in (9). The waiting queue clearance time (T_Q), as defined in (8), is obtained from the literature [12].

$$T_Q = \sum_{k=1}^K (|D_k - D_h|) \quad (8)$$

$$G_t = \begin{cases} R_i(ns), & \text{if } P_c = 0 \\ R_i(ew) + T_Q, & \text{if } P_c = 1 \end{cases} \quad (9)$$

$$\text{Decision 1} = \begin{cases} \text{Y,} & \text{if } \left[\frac{D_i}{G_t} \right] \in [S_f, S_{\min}] \\ \text{N,} & \text{if } \left[\frac{D_i}{G_t} \right] \notin [S_f, S_{\min}]. \end{cases} \quad (10)$$

Decision 2: Apply the MTM or SSM Strategy. Once Decision-1 is “Y,” which means the vehicle can pass through current intersection without stop, the multi-intersections mode should be applied. In Decision-2, the EDAS will decide whether MTM or SSM should be adopted. As shown in Figure 3, the MTM strategy tries to maximize the traffic

throughput so that the suggested speed will be as high as possible while remaining under the speed limit to reduce the possibility of blocking upstream vehicles. The SSM strategy, on the other hand, tries to minimize the acceleration and deceleration to keep the vehicle moving as smoothly as possible. The choice between these two strategies will be a compromise solution. On peak hours, adopting SSM may result in more serious traffic congestion, as it may block the upstream vehicles so that the number of vehicles stopped by the signal will rise, thus making fuel consumption and CO₂ emissions worse. On the other hand, adopting the MTM strategy on off-peak hour may increase the fuel consumption due to unnecessary acceleration.

For this reason we set a compromise factor α as the threshold to alternate between the throughput oriented and smooth speed models. The value of α is determined by the sensitivity test, and once α is determined OBU can decide to apply MTM or SSM based on the vehicle volume information collected from the host RSU.

$$EDAS = \begin{cases} \text{MTM,} & \text{if } \alpha \geq \text{threshold} \\ \text{SSM,} & \text{if } \alpha < \text{threshold.} \end{cases} \quad (11)$$

Process A: Maximized Throughput Model. In the MTM and SSM cases, the target of EDAS is to provide the optimal driving speed suggestion so that vehicles can move through multiple sequential intersections without being stopped by a signal. In order to maximize throughput and reduce the possibility of blocking the upstream vehicles, MTM suggests driving at as high a speed as possible while remaining safe and under constraints such as the speed limit, safely following the car in front, and so on. OBU suggests the highest speed under the calculated speed range to pass the next intersection when the vehicle passes the current intersection. By using highest speed $S_H(i, j)$ and lowest speed $S_L(i, j)$ from (12) which are the speeds needed to pass the i th intersection at the beginning and ending time of the green signal at the j th cycle is calculated, where TP_i and TG_i are the cycle time and green time of the i th intersection, respectively. For example, $S_H(2, 1)$ indicates the speed needed to drive through the second intersection at the first green second of the next signal cycle ($j = 1$). $S_r(3)$ indicates the suggested speed to pass the third intersection, which is computed by calculating all the possible speeds in all possible cycles ($j = 0$ to 4) and intersecting this with the speed limitation $[S_f, S_{\min}]$. Based on this, the highest legal speed in the lowest cycle will be the MTM eco-driving suggested speed (S_R).

$$S_H(i, j) = \begin{cases} \frac{D_i}{R_i(ns) + j * TG_i}, & \text{if } P_c = 0 \\ \frac{D_i}{R_i(ew) + T_Q + j * TP_i}, & \text{if } P_c = 1 \end{cases} \quad (12)$$

$$S_L(i, j) = \begin{cases} \frac{D_i}{R_i(ns) + j * TP_i}, & \text{if } P_c = 0 \\ \frac{D_i}{R_i(ew) + TG_i + j * TP_i}, & \text{if } P_c = 1. \end{cases}$$

As illustrated in Algorithm 1, MTM first calculates the speed needed to pass the i th intersection at the j th cycle in the first green period ($S_H(i, j)$) or the last green period ($S_L(i, j)$); that is, it finds the green band for the vehicle to sequentially drive through the intersections without any signal stops. By using the MTM strategy, OBU will choose the highest speed in the legal speed range which is available by intersecting the calculated speed range ($[S_H(i, j), S_L(i, j)]$) with the legal speed range ($[S_f, S_{\min}]$). The $MaxSR[i]$ array in Algorithm 1 is applied to keep the calculated highest speed in each cycle. It is initialized as zero and is assigned the calculated speed during the iterations and will remain at zero if there is no legal speed that can be assigned because the intersection of the two speed ranges is empty. The results of the MTM are presented in the suggested speed array $S_r[i]$. For example, the array $S_r = [60, 60, 50, 0]$ indicates that the vehicle can continuously drive through three intersections without stopping, and the suggested speeds for these intersections will be 60, 60, and 50 km/hr.

Process B: Smooth Speed Model. If there is a low volume traffic flow then the MTM strategy may not be energy efficient, because unnecessary acceleration would not help the vehicle to pass through more intersections than the SSM strategy. The SSM suggests a smooth driving speed that passes as many intersections as possible at a stable speed. This enables more economic driving as well as a more comfortable experience, since it avoids sudden acceleration or deceleration. Here we introduce a parameter β to let the advised speed range from $S_H(i, j)$ to $S_L(i, j)$, and the recommended speed S_r in SSM can be obtained from

$$S_r(i) = S_L(i, j) + \beta * (S_H(i, j) - S_L(i, j)). \quad (13)$$

Process C: Isolated Intersection Model (MinADM). In this process, the vehicle condition is that it has to stop by the signal at the intersection under the current signal phase (either green or red), and there is no need to consider the probability of blocking any vehicles that are behind it. We thus adopt the minimized acceleration and deceleration model (MinADM), which is simplified from a model proposed in our previous work [5]. In this model, OBU is only concerned with maintaining speed and deciding when to begin slowing down by gliding [5]. The recommended speed S_r in this model is obtained from (14) and (15).

(1) When the current phase is green

$$S_r = \begin{cases} S_c + A_g \cdot t, & \text{if } \frac{S_c^2}{-2 * A_g} \leq d_I \leq \frac{S_c^2}{-2 * A_g} \\ S_c, & \text{otherwise.} \end{cases} \quad (14)$$

(2) When the current phase is red

$$S_r = \begin{cases} S_c + A_g \cdot t, & \text{if } \frac{S_c^2}{-2 * A_g} \leq d_I - Q \leq \frac{S_c^2}{-2 * A_g} + D_h \\ S_c, & \text{otherwise.} \end{cases} \quad (15)$$

```

Input:  $N$ ,           // number of intersections
       $M$ ,           // number of cycles
       $RSU[i]$ ;      // RSU array ( $D_i, R_i(ns), R_i(ew), TG_i, TP_i$ )
Output:  $S_r[i]$ ,     // suggested speed for passing intersections:  $1 \cdots N$ 
(1) // finding out min, max speed for passing through  $i$ th intersection at  $j$ th cycle
(2)   for ( $i = 1; i \leq N; i++$ ) {           // iterate  $N$  intersections
(3)     for ( $j = 0; j < M; j++$ ) {         // iterate  $M$  cycles,  $j = 0$  means current cycle
(4)       if ( $P_c = 0$ ) {                 // current signal is green
(5)          $S_H(i, j) = D_i / (RSU[i] \cdot R_{ns} + j * RSU[i] \cdot TG)$ ; //
(6)          $S_L(i, j) = D_i / (RSU[i] \cdot R_{ns} + j * RSU[i] \cdot TP)$ ;
(7)       } else {                       // current signal is red
(8)          $S_H(i, j) = D_i / (RSU[i] \cdot R_{ew} + RSU[i] \cdot T_Q + j * RSU[i] \cdot TP)$ ;
(9)          $S_L(i, j) = D_i / (RSU[i] \cdot R_{ew} + RSU[i] \cdot TG + j * RSU[i] \cdot TP)$ ;
(10)      }
(11)    }
(12)  }
(13) // calculate the suggested speed for each intersection  $i$ 
(14)   for ( $i = 1; i \leq N; i++$ ) { // iterate  $N$  intersections
(15)      $MaxSR[i] = 0$ ;           // initialize suggested speed
(16)     for ( $j = 0; j < M; j++$ ) { // iterate  $M$  cycles to get max legal speed
(17)       if ( $(S_f < S_L(i, j))$ ) continue; // no solution
(18)       if ( $(S_H(i, j) \leq S_f)$ )  $S_r[i] = S_H(i, j)$  else  $S_r[i] = S_f$ ;
(19)       if ( $(MaxSR[i] < S_r[i])$ )  $MaxSR[i] = S_r[i]$ ; // choose the fastest
(20)     }
(21)      $S_r[i] = MaxSR[i]$ 
(22)   }
(23)   Output  $S_r[i]$ 

```

ALGORITHM 1: MTM eco-driving suggested speed (S_r).

The Issue of Safely Following Cars. In EDAS the top priority is the issue of safely car following, whatever the suggested speed is, OBU must obey the car following rule to prevent the vehicle from being closer than the safe distance. In such cases, OBU will dynamically calculate the suggested speed to maintain a safe distance from the front vehicle.

If the FCW module detects a front vehicle, OBU will switch to car following mode due to safety concerns. In this study, we apply ISO 15623 [13] to keep a safe distance between the focal vehicle and the one in front. As shown in (16), the safe distance (D_s) can be calculated by the vehicle's current speed (S_c), front car current speed (S_p), average maximum deceleration of both the vehicles (A_d), OBU refresh time t , and additional safe headway space (D_h , and in this study we set this as 1 meter). The suggested safe speed S_s for maintaining the safe distance D_s in the car following mode is defined in (17), which can be inferred by combining Newton's three law of motions with (16).

$$D_s = S_c \cdot t + \left(\frac{S_c^2 - S_p^2}{2A_d} \right) + D_h \quad (16)$$

$$S_s = 2S_p - S_c + A_p \cdot t - \frac{2}{t} (D_s - D_p) \quad (17)$$

Driving Action Suggestion. With the suggested moving speed S_r (without a front vehicle) or S_s (with a front vehicle), OBU can suggest appropriate eco-driving actions, including

speeding up, maintaining current speed, gliding (no pressure applied on the gas pedal), maintaining the safe speed, or braking by comparing the suggested speed (S_s or S_r) and the vehicle's current speed (S_c).

4. Simulation Study and Discussions

Three experiments are designed to evaluate the performance of the proposed MTM and SSM by comparing them with three related models. In the isolated intersection scenario, MTM is compared with MaxTM [5] and OTLCM [6]. In the multiple intersection scenario, MTM and SSM are compared with MaxTM [5] and PCC [10]. The third experiment tests 10 real cases collected from local city traffic bureau in isolated scenario. The performance of all the models is evaluated and simulated using Arena [14], a general simulation software which is effective in modeling and analyzing processes or flows.

To simplify the proposed model and without loss of generality, the traffic signal in the yellow (Y) period is combined with that in the green (G) period, and all the red (AR) periods are combined into the red (R) period. The assumptions used in all the simulation experiments are as follows: (1) all the vehicles are with traditional combustion engine, hybrid electric vehicle (HEV), and plugin hybrid electric vehicle (PHEV) are not considered in this work; (2) each traffic signal is equipped with an RSU, and the OBU penetration rate is 100%; (3) there are no lane changes or overtaking; (4) drivers always follow the recommend speed;

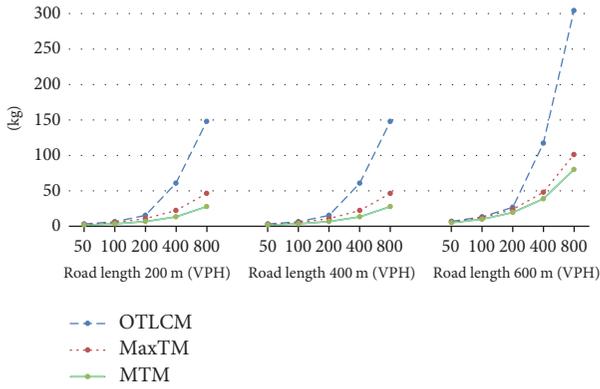


FIGURE 9: CO2 emission comparison in isolated intersection case.

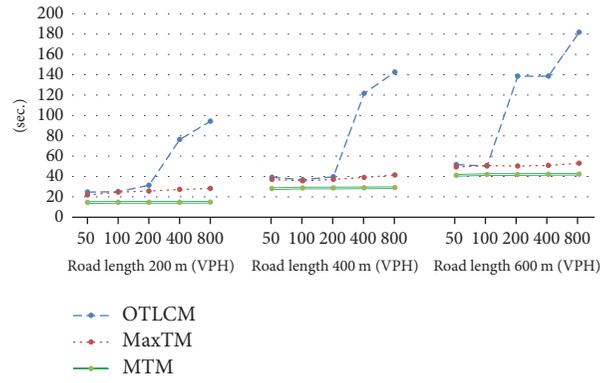


FIGURE 10: Averaged travel time comparison in isolated intersection case.

(5) there is only one type of vehicle; (6) all the signal cycle times and green splits (the fraction of a cycle when the signal is green) are the same in one simulation; and (7) both RSU and OBU adopt the one-hop message broadcasting mechanism. The simulation parameters are summarized in Table 1. The VANET one-hop communication range is 200 m, the road length parameters are 200~600 m, the traffic volume (in each direction) parameters range from 50 to 800 vph, and the traffic signal green-red period parameters are 30/30, 45/45, and 60/60. For each OBU, four sequential intersections are simultaneously considered in the spatial dimension, and four cycles are considered for each RSU in the temporal dimension.

To evaluate and compare the CO₂ emissions we adopt the Models for Projecting Energy Consumption and Air Pollutants Emissions (MPECAPE) [15], which considers microscopic CO₂ emissions and was developed by the Institute of Transportation, Ministry Of Transportation and Communications (MOTC) in Taiwan. It is established in 2007 by MOTC using emissions data collected from various types of vehicles considering multiple vehicle driving scenarios, including on highways and in townships and urban areas.

Experiment 1 (isolated intersection scenario). In an isolated intersection located in the center of four road segments in four directions, each road segment has two lanes connected in two directions, one for each direction. The proposed MTM is compared with MaxTM [5] and OTLCM [6] with regard to CO₂ emissions and average travel time, as shown in Figures 9 and 10, respectively. The results show that in the case of road lengths 200 m, 400 m, and 600 m, the CO₂ emission of MTM is less than MaxTM [5] and OTLCM [6] models on the cases from sparse traffic condition (50 vph) to congested condition (800 vph). The travel time simulation also shows the similar result; MTM has the best performance averaged travel time and the travel time variability is very stable in all the road length and traffic volume combination cases.

Experiment 2 (multiple intersections scenario). A cross road network topology consisting of thirteen intersections, as shown in Figure 5, is applied in the experiment. The assumptions and parameters are the same as those in Experiment 1. The two proposed strategies, MTM and SSM, are compared

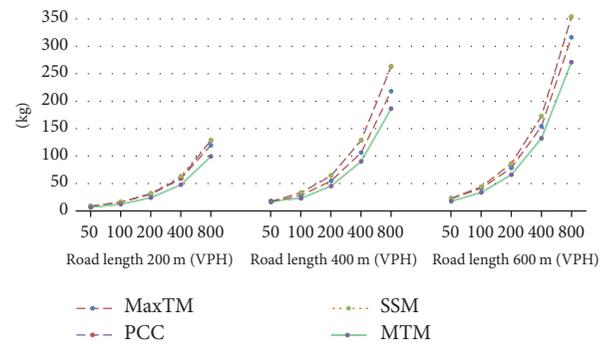


FIGURE 11: CO2 emission comparison in multiple intersections case.

with MaxTM [5] and PCC [10] with regard to CO₂ emissions and average travel time, as shown in Figures 11 and 12, respectively. In multiple intersections simulation, all the performances of these four models are quite well; MTM outperforms the other three models SSM, PCC [10], and MaxTM [5] in both CO₂ emissions, average travel time. Two interest points are revealed in this experiment. First, the performances of PCC [10] and SSM are almost the same due to the similar strategy and idea behind these models (except the gliding). Second, although MaxTM [5] is an isolated model which without the real-time knowledge of multiple intersections, it performs better than the PCC [10] and SSM. This may due to when SSM tries to smoothly pass through the multiple intersections, the green band (ranged from $S_H(i, j)$ to $S_L(i, j)$) will be narrower when the number of previewed intersections increased. This will result in the suggestion speed (S_r) advised by the SSM being much less than the MaxTM [5], which blocks the upstream vehicles and increases the possibilities of stops by signal for them.

Experiment 3 (real data collected from Tainan traffic bureau). In addition to these simulation cases, 10 real traffic cases including AM and PM peak hours as well as normal hours (as listed in Table 2) collected from the Transportation Bureau of Tainan City Government are also simulated. The collected parameters including road length, traffic flow volume, and traffic signal plan (green and red splits) are stated in Table 3. The proposed MTM is compared with MaxTM [5] and

TABLE 2: Case definition of the real traffic cases.

Case name	Duration	Classification
C1, C6	7:00 to 12:00 am	Hybrid
C2, C7	12:00 to 17:00 am	Normal
C3, C8	7:00 to 9:00 am	AM peak
C4, C8	17:00 to 19:00 pm	PM peak
C5, C10	19:00 to 24:00 pm	Night time

TABLE 3: Sample parameters of the real traffic cases simulation.

Case name	Time plan offset	VD ID	Road length	Traffic flow	Timing plan T_G	Timing plan T_R
C1	Initial	V009600	180 m	279.72	80 s	40 s
C2	Initial	V009600	180 m	383.80	80 s	40 s
C3	Initial	V009600	180 m	360.85	80 s	40 s
C4	Initial	V009600	180 m	526.06	90 s	40 s
C4	105 min	-	-	-	80 s	40 s
C5	Initial	V009600	180 m	419.02	80 s	40 s
C5	120 min	-	-	-	75 s	45 s
C5	195 min	-	-	-	55 s	35 s
C6	Initial	V052370	170 m	312.5	75 s	45 s
C7	Initial	V052370	170 m	443.35	75 s	45 s
C8	Initial	V052370	170 m	398.59	75 s	45 s
C9	Initial	V052370	170 m	507.54	95 s	35 s
C9	105 min	-	-	-	75 s	45 s
C10	Initial	V052370	170 m	401.51	75 s	45 s
C10	195 min	-	-	-	63 s	27 s

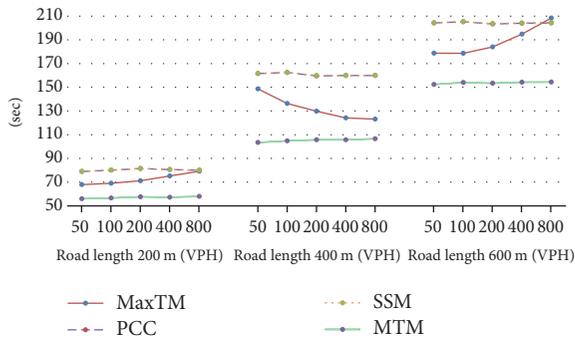


FIGURE 12: Averaged travel time comparison in multiple intersections case.

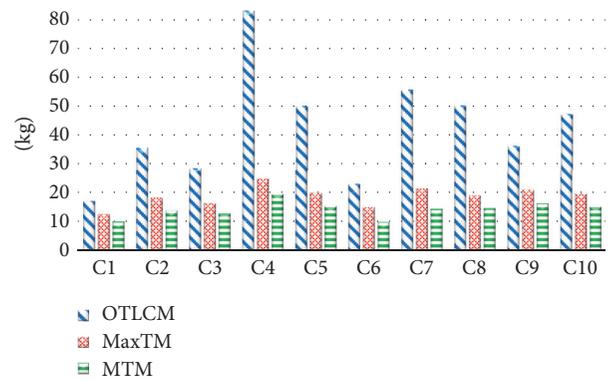


FIGURE 13: CO2 emission comparison in 10 real traffic cases.

OTLCM [6] with regard to CO₂ emissions and average travel time, as shown in Figures 13 and 14, respectively. The results show that the MTM outperforms MaxTM [5] and OTLCM [6] models in CO₂ emissions and average travel time on all the 10 cases.

5. Conclusions

With VANET, OBU on the vehicle can collect the information from RSUs and preview the intersection signal plans so that it can decide the optimal eco-driving speed to pass through multiple intersections with the minimum possibilities of stops. This work proposes a hybrid model that combines three

strategies including MTM, SSM, and MinADM to provide eco-driving suggestions. With the proposed EDAS, the suggested driving actions can help vehicles avoid unnecessary acceleration and braking and unnecessary stops. Moreover, several practical issues with regard to applying EDAS in the real world are taken into consideration including a safe and efficient car following mode, signal queue clearing time, cruise gliding, and idle gliding.

Traditionally, three indexes are applied to measure the performance of an intersection in traffic management science, including minimizing waiting queue length, minimizing vehicle waiting time, and maximizing the average cycle

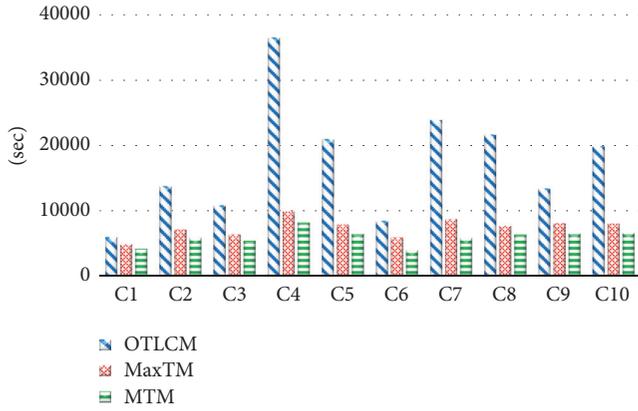


FIGURE 14: Travel time comparison in 10 real traffic cases.

time throughput. These indexes are generally considered to be positively correlated to each other, and thus if one index improves so will the two others. We adopt a strategy of maximizing the average cycle time throughput, and the simulation testing yielded promising results. Originally the MTM is designed for multiple intersections cases; however, the results show that it is 18.6%~40.1% better than MaxTM [5] and 25.1%~81.2% better than OTLCM [6] in CO₂ emissions, 16.6%~47.7% better than MaxTM [5], and 20.5%~84.3% better than OTLCM [6] in averaged travel time when MTM is applied to the isolated intersection scenario. For the multiple intersections scenario, the proposed MTM and SSM outperformed the PCC [10] model as well as the MaxTM [5]. Comparing to PCC [10], MTM is 19.9%~31.2% better in CO₂ emissions and 24.5%~35.9% better in travel time.

In real applications eco-driving recommendations depend heavily on traffic conditions [5]. While the maximize throughput strategy has good performance in congested as well as normal traffic conditions, it may not be suited for the case of very low traffic flow in off-peak hours, since the chance of blocking the rear vehicles is very low. In such conditions the SSM approach should be applied, as it may have better performance than in the MTM. In future work the traffic congestion threshold (α) will be studied in more detail and decided by sensitivity test experiments. Moreover, the smooth speed control parameter (β), which is applied to decide the suggested speed in SSM (as shown in (13)), can be further studied to acquire the optimal driving speed. Once these issues have been dealt with, the MTM and SSM can be combined into one eco-driving advisory system that is more practical for use with continuous signalized intersections.

Notations

R_i, R_s, R_p : RSU ID, configuration, and signal plan
 $R_o, R_{d,i}$: Host RSU, i th neighborhood RSU in direction d
 O_i : OBU ID
 n, s, e, w : North, South, East, West
 $n-s, e-w$: North and south direction, East and west direction

X, Y : Longitude and latitude
 T_p : Cycle time period
 $T_g(n-s)$: Green light time period in north-south direction
 T_s : Time stamp
 P_c : Current phase (0 = green, 1 = Red) in main direction (north-south)
 $R(n-s, e-w)$: Countdown remaining period
 L : Road length (L_n, L_s, L_e, L_w)
 Q_d : Queue length in direction d
 $T_Q(d)$: Queue dissipating time for direction d
 S_f : Free flow speed (Or speed limit)
 S_{min} : Min speed limit
 S_r : Recommend speed
 S_s : Safety speed for car following
 S_c : Current speed
 S_p : Front car speed
 S_d : Direction
 $S_H(i, j)$: Pass speed upper limit for i th signal at j th cycle
 $S_L(i, j)$: Pass speed bottom limit for i th signal at j th cycle
 A_c : Current acceleration
 A_p : Acceleration of the front car
 A_g : Cruise gliding deceleration
 D_p : Front car distance
 D_h : Minimum discharge headway
 D_k : Headway of the k th queued vehicle
 $SSID$: RSU Wi-Fi SSID
 α : Traffic congestion threshold
 β : SSM control parameter.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported in part by Ministry of Science and Technology of the Republic of China under Grant no. MOST 106-2221-E-006-044-.

References

- [1] IEA, *CO₂ Emissions from Fuel Combustion 2016*: IEA, 2016.
- [2] IEA. (2017). *World Energy Outlook 2017*, Nov. 2017.
- [3] M. Barth and K. Boriboonsomsin, "Traffic congestion and greenhouse gases," *Access Magazine*, vol. 2-9, 2009.
- [4] H. C. Frey, A. Unal, N. M. Roupail, and J. D. Colyar, "On-road measurement of vehicle tailpipe emissions using a portable instrument," *Journal of the Air & Waste Management Association*, vol. 53, no. 8, pp. 992-1002, 2003.
- [5] W.-H. Lee, Y.-C. Lai, and P.-Y. Chen, "A study on energy saving and CO₂ emission reduction on signal countdown extension by vehicular Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 890-900, 2015.
- [6] C. Li and S. Shimamoto, "An open traffic light control model for reducing vehicles' CO₂ emissions based on ETC vehicles," *IEEE*

- Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 97–110, 2012.
- [7] J.-Y. Lee, *A VANET Based ECO-Driving Advisory System for Reducing Fuel Consumption and CO2 Emissions on Signalized Multi-intersections [Master, thesis]*, National Cheng Kung University, 2014.
 - [8] M. Alsabaan, K. Naik, T. Khalifa, and A. Nayak, “Vehicular networks for reduction of fuel consumption and CO2 emission,” in *Proceedings of the 8th IEEE International Conference on Industrial Informatics, INDIN 2010*, pp. 671–676, jpn, July 2010.
 - [9] M. Alsabaan, K. Naik, and T. Khalifa, “Optimization of fuel cost and emissions using V2V communications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1449–1461, 2013.
 - [10] B. Asadi and A. Vahidi, “Predictive cruise control: utilizing upcoming traffic signal information for improving fuel economy and reducing trip time,” *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 707–714, 2011.
 - [11] K. Katsaros, R. Kernchen, M. Dianati, and D. Rieck, “Performance study of a Green Light Optimized Speed Advisory (GLOSA) application using an integrated cooperative ITS simulation platform,” in *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC '11)*, pp. 918–923, Istanbul, Turkey, July 2011.
 - [12] J. A. Bonneson, “Change intervals and lost time at single-point urban interchanges,” *Highway Capacity and Traffic Flow (Transportation Research Record)*, 32, 1993.
 - [13] ISO. (2013). *Intelligent transport systems – Forward vehicle collision warning systems -Performance requirements and test procedures ISO15623:2013*.
 - [14] W. Kelton, R. Sadowski, and N. Zupick, *Simulation with Arena, 6/e*, McGraw-Hill, 6/e, 2015.
 - [15] K. Lin, *Integrating the Applications of Sustainable Transportation Planning Model and Models for Projecting Energy Consumption and Air Pollutants Emissions*, Institute of Transportation, Ministry of Transportation and Communications, 2010.

Research Article

Design Method of ADAS for Urban Electric Vehicle Based on Virtual Prototyping

Katarzyna Jezierska-Krupa  and Wojciech Skarka 

Institute of Fundamentals of Machinery Design, Faculty of Mechanical Engineering, Silesian University of Technology, Gliwice, Poland

Correspondence should be addressed to Wojciech Skarka; Wojciech.Skarka@polsl.pl

Received 25 June 2017; Revised 17 October 2017; Accepted 12 November 2017; Published 5 March 2018

Academic Editor: Rongxin Cui

Copyright © 2018 Katarzyna Jezierska-Krupa and Wojciech Skarka. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Since 2012, the Smart Power Team has been actively participating in the Shell Eco-marathon, which is a worldwide competition. From the very beginning, the team has been working to increase driver's safety on the road by developing Advanced Driver Assistance Systems. This paper presents unique method for designing ADAS systems in order to minimize the costs of the design phase and system implementation and, at the same time, to maximize the positive effect the system has on driver and vehicle safety. The described method is based on using virtual prototyping tool to simulate the system performance in real-life situations. This approach enabled an iterative design process, which resulted in reduction of errors with almost no prototyping and testing costs.

1. Introduction

Conducting the design process of Advanced Driver Assistance Systems (ADAS) in an optimum way requires a specific approach to defining and solving problem [1]. Profound research and analysis have to be carried in order to prepare well for this task. Sometimes it is impossible though to identify and avoid certain mistakes in the process, without using a prototype [2–4]. Currently, there are more and more tools available for engineers that enable creating virtual prototypes and therefore optimizing the design process in terms of its effectiveness and fault minimization [5]. Nevertheless, in order to use these tools efficiently, there needs to be a proper approach to design process adopted. The aim of this article is to present the approach to ADAS design process, which includes the use of simulation methods and virtual prototyping. Preceding the actual method description, there are some considerations included, whether there are reasonable grounds for using virtual prototyping tool and what other tools should be considered before making the final decision. The subsequent sections of this article include the method description and the use case, that is, design process of Blind Spot Information System for urban vehicle. This article is an extended version of the article [6] presented at the International Conference on Transdisciplinary Engineering 2016 in Curitiba, Brazil.

2. Consideration of the Rationale for Using Virtual Prototyping Tool

2.1. Defining the Needs. The first step that needs to be taken into system of design process needs definition [4, 7, 8]. What must be identified are the problems that a driver is facing and possibilities of finding the right solutions for them. The most effective ways of finding them are interviews with actual drivers and analysis of traffic situations of great risk for driver, vehicle, or third parts safety. The most common conditions in which the driving task is performed [1, 5] need to be analysed in this step. For vehicles dedicated for specific goals, such as competitions, these conditions can be described very precisely, which enables finding most current problems and most effective solutions. The choice of prototyping tool, which would be used in further steps, needs to be based on tool possibilities in terms of reflecting real environmental and situational conditions.

2.2. Defining the External Constraints. Before deciding on using virtual prototyping tool for ADAS design, it needs to be considered whether there is such a need at all. To answer this question, the external constraints must be considered and it must be verified how the design process will be affected by using simulation methods.

There are multiple constraints that must be considered, when thinking about designing ADAS. One of the most basic ones are costs, time, and quality. According to a well-known business rule of thumb, it is accepted that only two of these three constraints can be fulfilled at the same time.

2.3. Discussion of Possible ADAS Design Methods. There are various methods of ADAS design available and it is crucial to choose the one that fits the needs of the project best. ADAS can be designed in the following ways:

- (1) No prototype: which does not enable us to verify its performance in real-life situations until the system is implemented
- (2) Virtual prototype: which, accordingly to a chosen tool, enables us to simulate the natural conditions in really accurate but not perfectly accurate way, generates low costs, and enables the iterative design process [3, 5]
- (3) Real prototype: which enables absolute verification but has also many drawbacks such as costs, production, and construction time or difficult interaction [3]

When considering the virtual prototyping tool, the above-mentioned aspects must be included.

3. Design Method

3.1. Project Analysis. In order to minimize the need for changes in further steps of the ADAS design process, it is recommended to define all internal constraints before working on an actual model. This approach enables us to precisely set the goal of the designed ADA system and define the required constraints early enough to be included in the model. Using a virtual prototyping tool for such complicated systems as ADAS requires splitting considerations about the whole system into subsystems.

3.2. Designing and Testing. Testing part of designed system is a step that must not be neglected. Thorough tests are the last step before real system implementation that allow error detection without generating additional costs in manufacturing process. Correctly conducted testing enables also very accurate error identification and makes its correction easier.

In order to properly define test cases for ADAS virtual prototype testing, three separate system aspects must be considered, in which an error could appear:

- (1) Data processing: software
 - (1.1) What are the system inputs?
 - (1.2) What are the system outputs?
 - (1.3) What is the path of converting inputs to outputs?
 - (1.4) What values are independent from the system design and may not be tested?
- (2) Functional requirements: UI and usability
 - (2.1) What is the system supposed to do?



FIGURE 1: Bytel vehicle during Shell Eco-marathon race in Rotterdam (2015).

- (2.2) What information is it supposed to communicate to the driver?
 - (2.3) What must the system not do?
- (3) Models design: the idea
 - (3.1) How should the system work?
 - (3.2) What inputs and outputs should it accept/generate?
 - (3.3) What functions are the system described by?

Such analysis enables us to create test cases sheet that should be used in test experiments outcome evaluation. Only when every test case will end successfully, that is, the values will be corresponding with template values and the visual verification will be fulfilled, the system can be implemented.

For correct testing process of ADAS virtual prototype, it is also needed to cover all possible situations in simulations, in which an error could appear. The test experiments should include both typical traffic situations of a higher risk and nonrisky situations in order to test the system against false alarms.

4. Use Case: BLIS for Urban Electric Vehicle

The above-described design method was used in design process of Blind Spot Information System (BLIS) [2, 6, 9, 10] for urban electric vehicle, Bytel [6, 11, 12] (Figure 1). Bytel is a vehicle created to participate in Shell Eco-marathon (SEM) [13] competitions, the event which aims to encourage design of highly efficient vehicles by students and scientific organizations. Bytel has participated in SEM in 2014 and 2015 in two power source categories: battery electric and hydrogen (hydrogen fuel cell stack).

The steps described above were taken to successfully finish the process at minimal time and maximum quality, keeping the costs reasonable at the same time.

4.1. Historical Determinants of BLIS System Development. The concept of this BLIS system is based on the team's experience in this field, taking into account specific conditions and upon existing solutions. The basic assumption of the system is to inform the driver about the appearance of another vehicle in

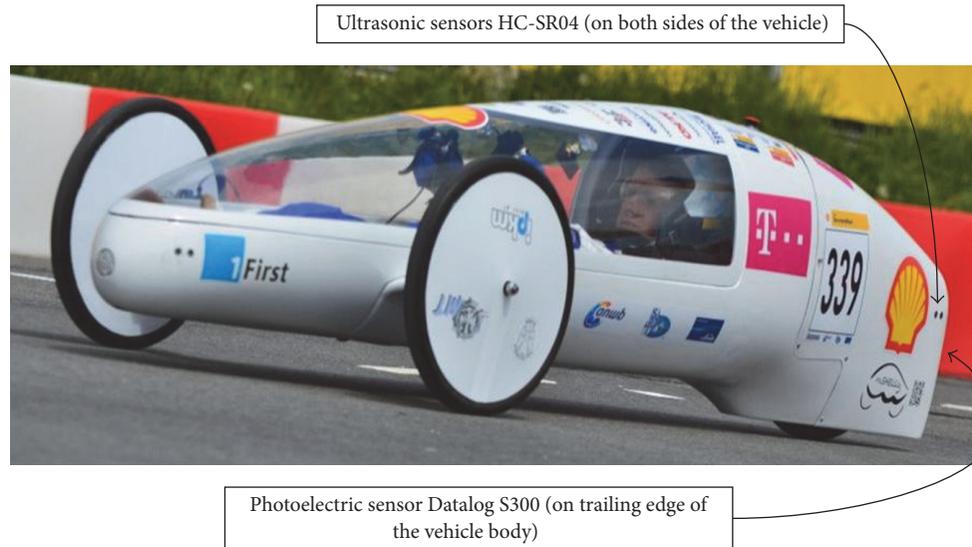


FIGURE 2: Vehicle MuSHELLka at Shell Eco-marathon 2013 in Rotterdam. Placement of BLIS system sensors in MuSHELLka.

the blind spot. The final form of the system was influenced by experience with previous ADAS systems, including the BLIS system, which was fitted with another vehicle [11] (MuSHELLka) developed by the team (Figure 2).

Previous technical solution of BLIS system [6, 11] is composed of a controller which is based on Microcontroller ATmega8 and 3 sensors. The main task of the controller is to handle sensors and to transmit information about the detected threat to ACS controller using RS232. Two ultrasonic sensors HCSR-04 and diffusive photoelectric sensor Datalog S300 are used for scanning area behind the vehicle. The HC-SR04 ultrasonic sensor uses sonar to determine the distance from an object in the range of 2 cm–400 cm, with 3 mm resolution. Sound wave that is used for measurement has a frequency of 40 kHz [11]. Datalogic S300 is an advanced photoelectric sensor with a detection range of 0–500 cm for white objects and 0–350 cm for black objects. Infrared radiation and triangulation are used for measurement. The sensor response is binary, and the detection range can be adjusted in the full range of measurement. Due to the use of the sensor, a critical parameter is the response time. Maximum response time for this sensor is shorter than 2 ms [11]. All system components are placed in the rear fairing, which was possible thanks to the compact design and minimum weight.

This solution proved to be a much smaller vehicle because of its simple construction, low cost, and high operational reliability. It is also suitable for small size vehicles and small mass. It is important to use the BLIS virtual prototyping method in the development of this system. As part of the development of the previous system, a virtual race track system was used, a vehicle model in the TASS PreScan [14] environment, and a numerical simulation model of the BLIS system in this environment using the MATLAB/Simulink environment. As part of the development of the previous system, a full simulation test was conducted in the simulation environment and subsequent verification was performed on

the actual site. These tests confirmed the compatibility of the system parameters with the assumptions and the experience of the design process, virtual testing, and racing operation are the basis for the development of the new BLIS system.

As the next designed vehicle [15, 16] (Bytel) is only equipped with side mirrors and does not have a rear window or a central rear-view mirror, it is not only the monitoring of the area on the side of the vehicle, but also the area immediately behind the vehicle. A number of conceptual solutions have been analysed, of which three are presented below (Figure 3).

The first concept (Figure 3(a)) was based on the use of photoelectric sensors. The experience of using this type of sensor in previous solutions indicated the potential for such a system, but ultimately, due to the need to use a large number of such sensors, the vehicle aerodynamic effects were adversely affected, and its relatively low efficiency and range, especially in variable weather conditions, and insufficient information obtained from such a system has been rejected.

The second concept (Figure 3(b)) was based on the use of a vision system based on the Kinect. The design assumes the use of the depth camera included in the kit to determine the distance of the approaching vehicles. If larger areas than a single sensor have to be scanned, more Kinect devices can be used. This, however, complicates the connection and requires data analysis from two or more different sensors. Creating a system, based on two or a few of this type of sensors, which would recognize an object with satisfying accuracy and calculates position, is a difficult task and thus it requires great processing power. This affects not only the recognition of the image, but also the angle at which the vehicles will be detected. Taking into account the parameters of a commonly available Kinect device, a sensor placement scheme was used when using four devices. The advantages of this solution are the innovation and the use of ready-made modules and the large potential of the system. The disadvantages include the large dimensions of the device, relatively small scanning angle

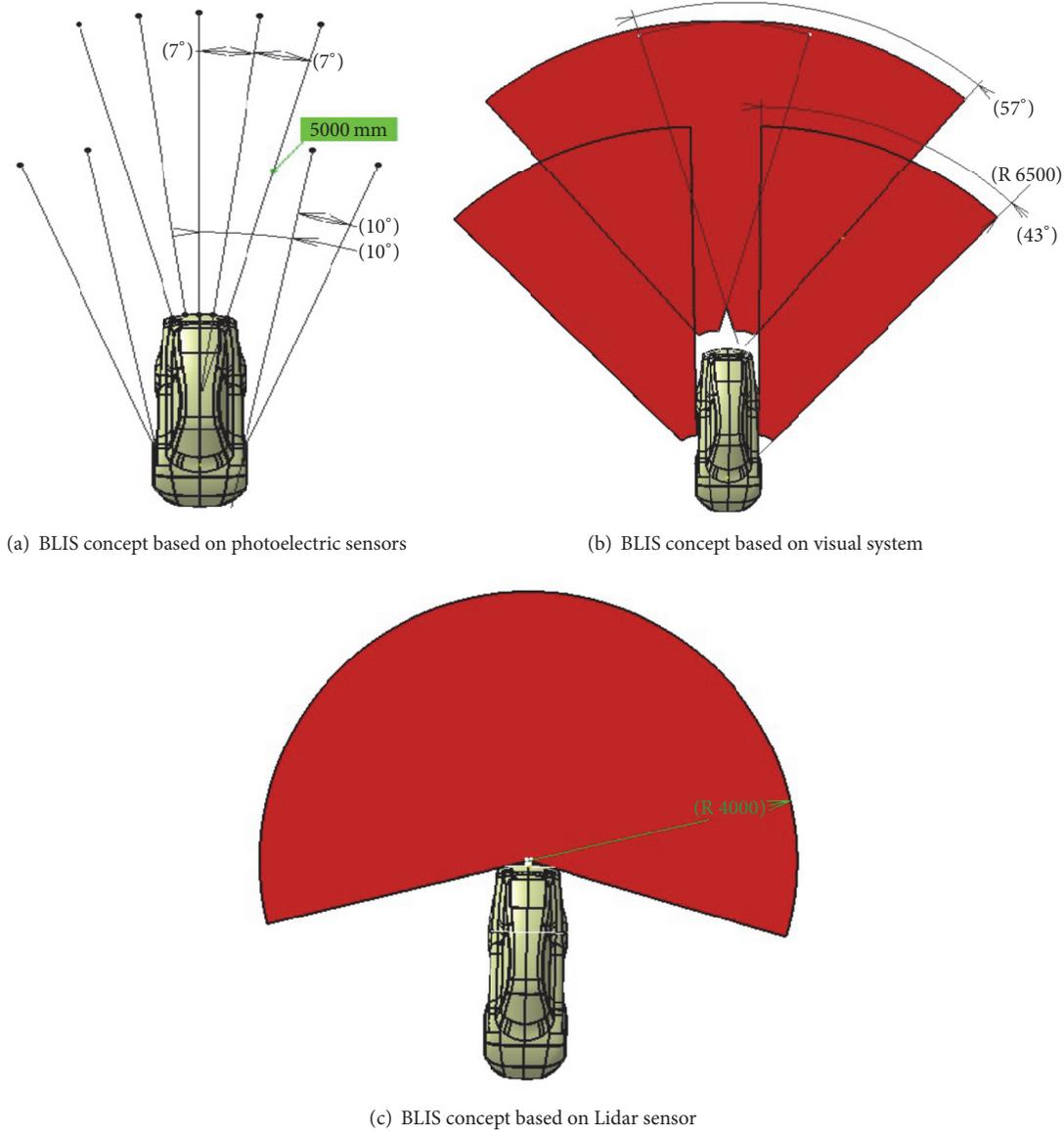


FIGURE 3: Alternative concepts of BLIS system.

in case of a single sensor, the possibility of disturbance of measurements in real conditions (sun rays), and the need to use a PC.

The third concept (Figure 3(c)) assumes the use of a Hokuyo laser sensor. This sensor is characterized by high resolution (1080 samples per 270° , corresponding to 4 samples per 1°), high sample rate (40 Hz), and wide scan area (270°) [17]. The laser used in this sensor fulfils the 1st safety class, which allows it to be used during SEM competition and is characterized by low sensitivity to atmospheric agents (sun, rain, snow, and fog). The advantages of this solution are simple assembly, single device, and high reliability, whereas the main disadvantage is the high cost of the sensor and the possibility of problems arising from the analysis of so much data. This, however, provides a field for the use of complex algorithms to analyse the data to eliminate errors and increase operational certainty, which was also used and presented later

in this paper. Ultimately, the laser sensor solution has been used in further development work.

The reflections and experience gained from previous development work led to the new BLIS solution to modify both the system itself and the design methodology. The new design method was shortened by abandoning the cost-intensive and time-consuming verification of the real object. Each of the newly constructed vehicles is treated as a prototype vehicle and a shortened test of the actual object is justified. These tests are designed to be carried out on the track and during races. This does not mean that the system will not be verified. The emphasis was placed on virtual prototyping, a formal verification phase was established during these studies, and new methods were introduced to verify the results of the simulation tests by introducing new independent computational methods to duplicate the results of standard simulations.

As far as the concept was concerned, it was decided to change the concept of the system completely by introducing a new interface in the driver's field of vision, providing much more information but not distracting the driver. The use of a single Lidar sensor with a range of measurable resistance to ambient conditions and at the same time to meet the rigorous requirements of the vehicle aerodynamic performance is critical to ensure high reliability of the proposed system of the development method using virtual prototyping methods and it provides the possibility of using this sensor and integration of the BLIS system into other ADAS systems [18, 19].

4.2. Consideration of the Rationale for Using Virtual Prototyping Tool. The decision to develop BLIS for urban vehicle was made based on low viewing range, which is characteristic for common conditions for these kinds of vehicles [18], that is, vehicles designed specifically for Shell Eco-marathon competitions. BLIS is the system that informs the driver about vehicles coming from the back of the vehicle, which are invisible to the driver due to its low viewing range. The need of creating such a system has been recognized based on analysis of many accidents and dangerous situations happening during the race and also after discussing the problem with drivers participating in SEM.

While deciding to use virtual prototyping tool, the crucial reason for using it was the necessity for parallel production process. As the vehicle was still in development phase during the design of BLIS, there was no possibility of using real prototype in order to conduct tests and work on data processing improvement. This is equivalent to accepting time as the main external constraint that applies to this project.

The second constraint was quality of the system, as the vehicle purpose was to take part in SEM competition; that is, in the race, it was considered extremely important to provide the driver with ADAS of the best possible quality. The point was to help the driver keep safe and avoid active or passive participation in accidents.

Last but not least, the adequate tool needed to be chosen for designing and conducting BLIS simulations. There were a few virtual prototyping tools especially dedicated to ADAS design which were considered. The final choice was TASS PreScan [14] due to its wide range of predefined sensors, very friendly user interface, and great visualization capabilities.

4.3. Project Analysis

4.3.1. The Scope of Development Works Performed Using Virtual Prototyping Methods. As part of the development work on the new BLIS system for the Bytel vehicle, the following scope of work is planned:

- (i) Analytical analysis of previous versions of BLIS and competing solutions
- (ii) Development of detailed functional requirements and requirements for integration with vehicle and other vehicle systems, as well as internal and external constraints
- (iii) Development of a detailed BLIS concept

- (iv) Construction of a simulation environment involving simulation models of an example race track, designed vehicle, and other vehicles
- (v) Construction of the mathematical simulation model of the BLIS system
- (vi) Identification and development of simulation scenarios
- (vii) Development of alternative algorithms for methods of verification of virtual prototyping simulations
- (viii) Carrying out a series of simulations with verification tests and improving the parameters and structure of the developed BLIS solution in iterative mode until the results are achieved
- (ix) Performance analysis and production and integration plan

According to the designers of the vehicle, a set of detailed planning simulation tests and independent allowing algorithms to verify the results of simulation studies are particularly important for the new method. Correct planning and carrying out of this part of the research guarantee the possibility of doing without the physical prototype. Therefore, further attention was paid to the detailed description of these fragments of the research.

4.3.2. Defining the Internal Constraints. Following the described method, the first step after recognizing the need of creating BLIS and defining the external constraints that lead to decision of using virtual prototyping tool was defining the internal constraints of the designed system.

- (1) The system cannot affect actively the breaking and/or steering system.
- (2) The system functions must be restricted to information/warning functions.

The self-imposed constraints of the project were also directly connected to Shell Eco-marathon assumptions, as the purpose of Bytel vehicle was to achieve the best result possible in this competition (SEM is about developing highly energetic vehicles). In order to satisfy this goal, there were the following constraint sets:

- (1) The weight of the system must be minimized.
- (2) The system should not consume relevant amount of power.

These constraints enabled us to define the general approach to designed system.

At this point, the decision regarding system parts has been made, which was vital to create the system model in TASS PreScan. There were three possible kinds of sensors considered: photoelectric sensors, Lidar, and depth camera. These kinds of sensors are available in predefined form in TASS PreScan, with a possibility of their configuration in very wide range. The decision was made to use Hokuyo Lidar (Figure 3(c)).

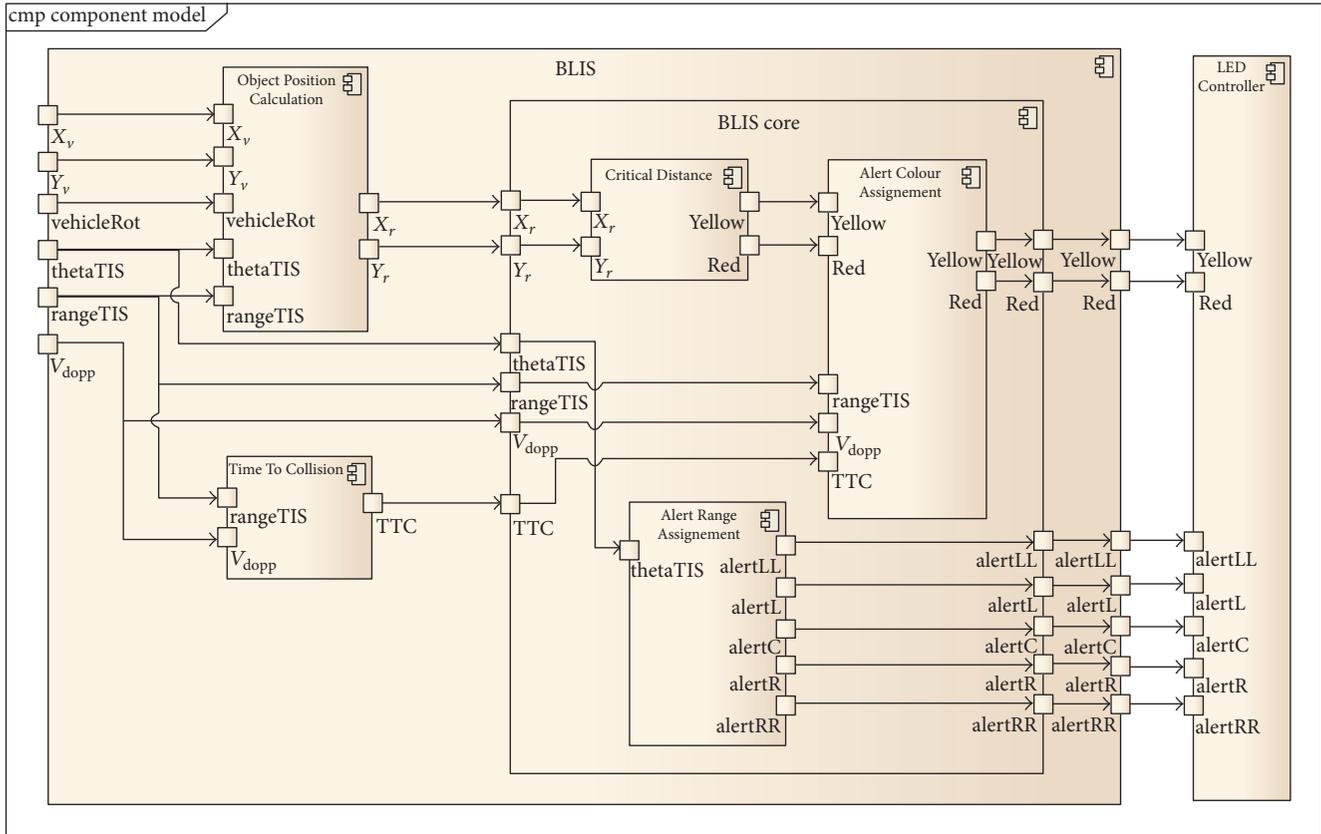


FIGURE 4: Flowchart of BLIS for Bytel vehicle.

What was crucial when deciding about rejecting the photoelectric sensors option was their low resistance to environmental conditions; a high risk of inaccurate results was identified. Also, in case of photoelectric sensors, there would be a need for using at least 9 sensors in order to cover the area invisible to the driver sufficiently, whereas there was only one Lidar needed to enable scanning of this range.

However, the difficulties in configuring and designing a system based on depth camera and the size of the system that interfere with the aerodynamic requirements have resulted in rejection of this solution.

4.3.3. Splitting System into Subsystems. Defining the subsystems and splitting the system into them when developing a virtual prototype not only made it easier to follow the design process step by step but also created good ground for tests analysis.

The main determinants of dividing line which would be used for splitting the system were assumed real-life subsystems. This means that it was needed to consider the separate parts of BLIS system, as they were meant to be implemented in real life. This is why the decision was made to consider BLIS in two subsystems: data processing subsystem and control subsystem.

The data processing system included also a verification subsystem, which uses the data generated in PreScan in order to compare the values between ideal ones and the ones that were received via data processing based on assumed

input values. An example of such comparison can be seen in Figure 9.

4.4. Operational Model of the System. Operational model of the system can be seen in Figure 4. The base of the system is the use of one laser sensor with a large angular range and coverage.

Input variables of the system are as follows:

- (i) X_v : X coordinate of the Bytel vehicle in the absolute coordinate system
- (ii) Y_v : Y coordinate of the Bytel vehicle in the absolute coordinate system
- (iii) vehicleRot: rotation angle of the vehicle in the z-axis
- (iv) thetaTIS: the angle at which the vehicle was detected
- (v) rangeTIS: the distance of the detected vehicle to the sensor
- (vi) V_{dopp} : vehicles relative velocity

Outputs of the system are as follows:

- (i) yellowAlert: low (yellow) risk information
- (ii) redAlert: high (red) risk information
- (iii) alertLL: information about the appearance of an object in the extreme left area

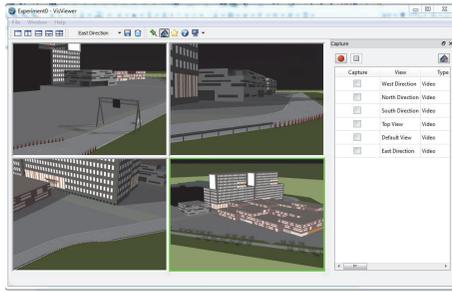


FIGURE 5: Model of the environment made in TASS PreScan.

- (iv) alertL: information about the appearance of an object in the left area
- (v) alertC: information about the appearance of an object in the center area
- (vi) alertR: information about the appearance of an object in the right area
- (vii) alertRR: information about the appearance of an object in the extreme right

4.5. Building Scenario: Building Environment

4.5.1. Simulation Environment Modelling. In order to carry out the relevant simulations to verify the correctness of operation of the designed system, it was necessary to design a simulation environment, vehicles with the dynamics, and trajectory, as well as modeling the sensor which is the basis of the system. Arrangement of roads and buildings is based on the imported Street Map model of the Ahoy Arena area in Rotterdam, where the Shell Eco-marathon 2014 competition took place. The designed model of the environment (Figure 5) was supposed to reflect the actual conditions as faithfully as possible.

4.5.2. Vehicle Movement Parameters Modelling. After building the environment, we have to upload the vehicle model for which Advanced Driver Assistance System is designed. It is necessary to assign a trajectory and to define the parameters of vehicle movement.

TASS PreScan software allows implementation of any velocity profile and trajectory. First trajectories have to be defined and then velocity profile for the two vehicles has to be determined. As a result, speeds and acceleration and paths of the vehicles during the simulation are described in detail. Figure 6 shows the speed profile for a vehicle overtaking a leading Bytel vehicle.

4.5.3. Sensor Parameters Modelling. An important point of the simulation design process is accurate mapping of parameters of used sensor. The PreScan allows the use of the Lidar model; however, due to the large angular range and high frequency sampling, a better solution was to use a sensor of TIS type (Technology Independent Sensor) (Figure 7). TIS, thanks to the wide possibilities of modification of parameters, has enabled a very accurate representation of the Hokuyo sensor to be used in the BLIS assistance system.

PreScan software enables visual verification of the range of the proposed sensor (Figures 13, 14, and 15), which helps in controlling the distance and angular range. With this visualization, the right area of operation for the designed sensor can be easily chosen. By controlling the moment of appearance of another vehicle in sensor range, it is possible to verify operation during simulation of the system as early as at design stage (Figures 13, 14, and 15).

4.6. System Model: Data Processing System. After developing all the details of the simulation, it is allowed to start modelling of the system. TASS PreScan software cooperates with the MATLAB/Simulink software. When installing TASS PreScan the reference to MATLAB/Simulink is established. After starting PreScan Sim module user has ready-made models of sensors, vehicles, and other control elements visible as Simulink models. These models are generated by the TASS PreScan individually for each of the simulations taking into account the modifications described earlier.

4.6.1. System Main Window. The main window of system visible after opening the PreScan Sim module contains two models of vehicles and components fulfilling control warning LEDs functions in the simulation.

The main block model represented in the main window of PreScan Sim module contains modifications resulting in BLIS system design. The output from the model of the first Bytel vehicle that is a vehicle equipped with a sensor provides brief information on occurrence of alarms yellow and red and the area of detection of the object. At the input of each diode is given the information about activity in its assigned area and information about yellow or red alarm occurrence.

4.6.2. Vehicle Model. GUI window of vehicle model includes four groups of blocks:

(i) After entering the vehicle model (Figure 8), in the left side of the window, you can see models generated by the PreScan software describing it. In the case of a Bytel vehicle, it is a trajectory model, parameters motion model, and model of sensors assigned to the vehicle, TIS. Based on previously developed block diagram showing the planned method of the system operation, you can immediately pull out of these variables which are necessary when developing a BLIS model: x and y coordinates of the absolute position of the Bytel vehicle, rotation of the vehicle in the absolute coordinate system (vehicleRot), speed of the vehicle (V_v), the value of distance of detected obstacle (rangeTIS), and the relative speed of vehicles were determined using the TIS sensor (V_{dopp}) and the angle in which the obstacle appears (thetaTIS).

(ii) Blocks containing programs designed to perform the calculations of parameters used in the BLIS system: Object Position Calculation, Time To Collision, and Object Velocity Calculation. The blocks are linked to necessary input variables and output variables are derived.

(iii) Block of BLIS system together with its inputs and outputs

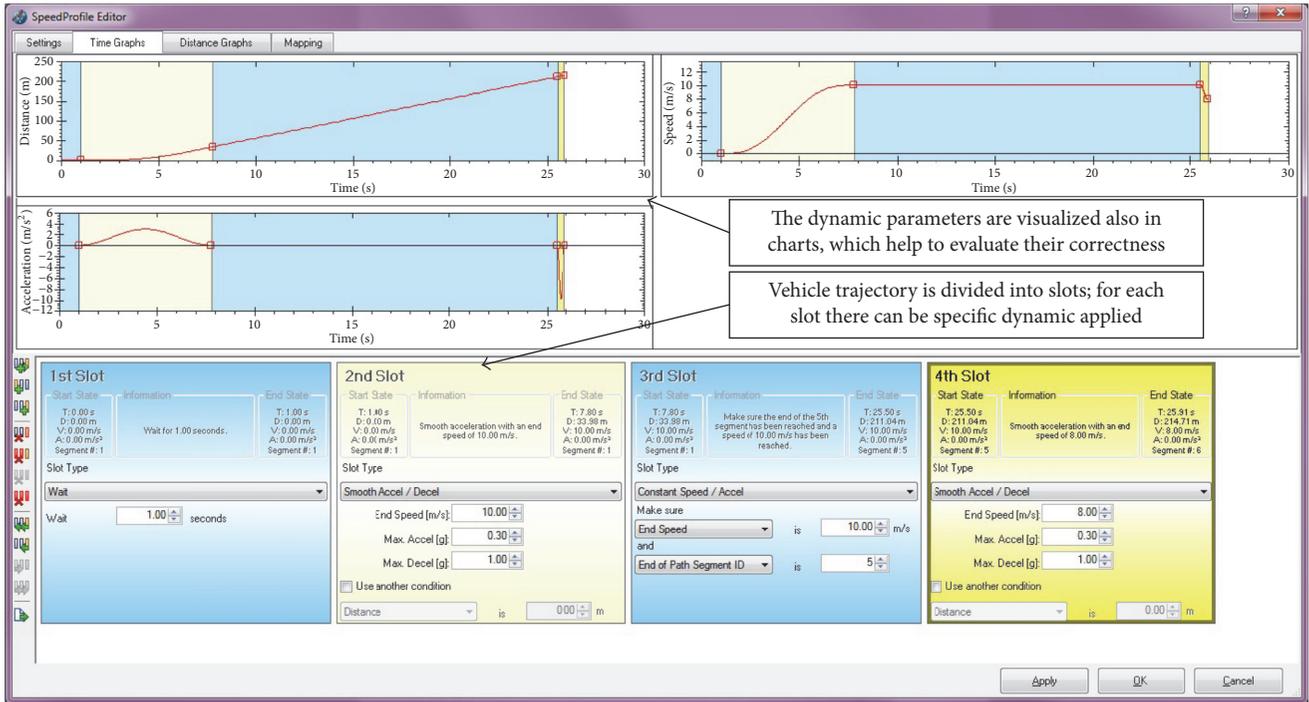


FIGURE 6: Speed profile for overtaking vehicle.

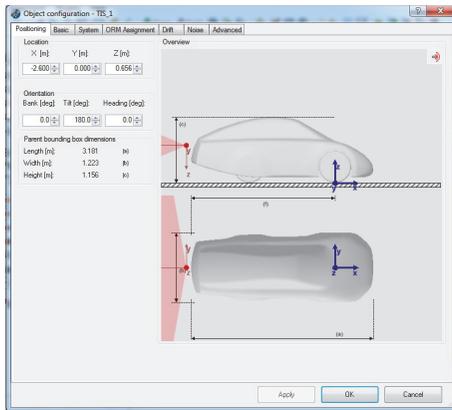


FIGURE 7: Modelling Hokuyo Lidar in BLIS.

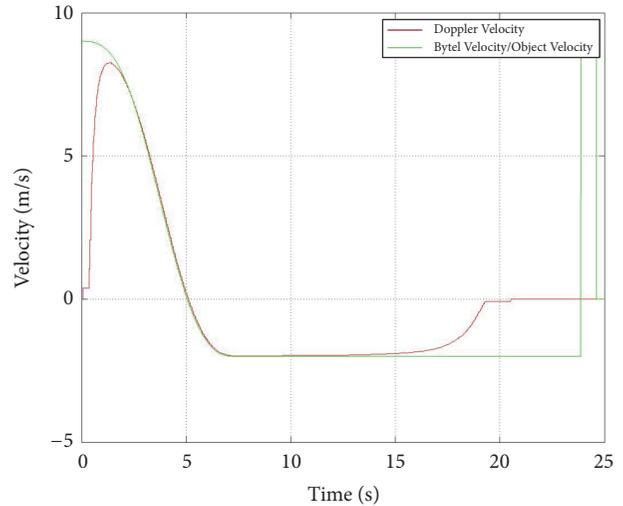


FIGURE 9: Comparison of two values: ideal one (Doppler Velocity) and the one achieved in data processing (Byte Velocity, Object Velocity).

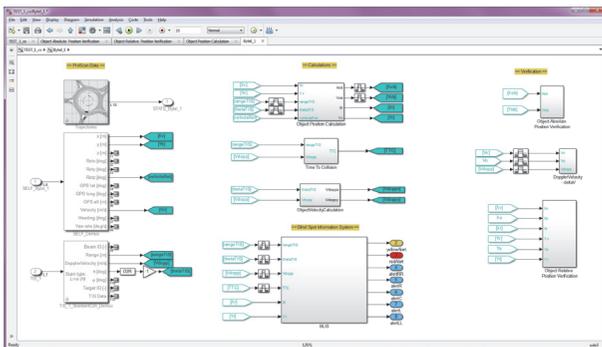


FIGURE 8: General models and blocks of the BLIS system.

(iv) Blocks including programs used to verification calculations

4.6.3. Part I: Calculations. Based on the block diagram of the BLIS system, it had to be determined which calculations should be performed to obtain the necessary to define the logic of the system variables. Directly from the PreScan software, the following variables were extracted:

- (i) X and Y coordinates of Bytel vehicle in the absolute coordinate system
- (ii) Rotation angle of the vehicle in absolute coordinate system (vehicleRot)
- (iii) Vehicle velocity (V_v),
- (iv) The distance of the detected obstacle (rangeTIS)
- (v) Relative velocity determined by TIS sensor (V_{dopp})
- (vi) The angle at which obstacle was detected (thetaTIS)

These variables do not provide direct way to determine the logic of the system, which was based on three basic pieces of information:

- (i) Comparison of the relative position of the object with the value of YCD (Yellow Critical Distance) or RCD (Red Critical Distance), which are functions defining the threshold value of position of the object in the x axis, for which there is raised up risk of collision
- (ii) Value of relative velocity of vehicles V_{dopp}
- (iii) TTC value (Time To Collision)

On the basis of these variables, the following result variables were obtained in the way of calculations:

- (i) X_r : x coordinate of the object relative position relative to Bytel vehicle determined based on the value rangeTIS and thetaTIS
- (ii) Y_r : Y coordinate of the object relative position relative to the Bytel vehicle determined based on the value rangeTIS and thetaTIS
- (iii) Yellow Critical Distance: function determined based on the value Y_r
- (iv) Red Critical Distance: function determined based on the value Y_r
- (v) Time To Collision: time to collision, determined based on the value rangeTIS and V_{dopp}

4.6.4. Object Position Calculation. The activities carried out in Object Position Calculation block are divided into two sub-systems: the calculation of the value of X and Y coordinates relative position of the detected object and calculation of the coordinate values of X and Y absolute position of the object.

Program used to calculate the values of the X and Y coordinates relative position of the detected object accepts as input values thetaTIS and rangeTIS. Values X_r and Y_r determining the relative position of the vehicle in the x and y axis are obtained by transforming the coordinate system of polar to Cartesian coordinate system.

The second part of the Object Position Calculation block represents calculations regarding the position of the object in the absolute coordinate system. The program uses the input values: thetaTIS, vehicleRot, rangeTIS, coordinates X_v and Y_v , point rotation matrix in the two-dimensional space, and value of coordinates and the value o displacement vector of the center of the local coordinate system of the Bytel vehicle from the edge detected in the x and y axis.

An important assumption was rangeTIS $= 0$, because the value 0 is assigned to variable rangeTIS at a time when there are any object detected in sensor range. In this case, variables X_{ob} and Y_{ob} are assigned the value 0.

4.6.5. Time To Collision. Time To Collision block task is to calculate the value of time, from the temporary situation of the temporary traffic parameters to the potential collision. This is an important element of calculations, because the value of TTC is directly influenced by the decision to activate the yellow or red alarm.

Also, here switch block was used making passed signal dependent on the value of the variable V_{dopp} . The variable takes the value 0 when there are no objects detected in the sensor range sensor. In this case, the variable TTC is assigned the value 20 as the maximum time for which the threat of a collision is practically nonexistent.

4.6.6. Object Velocity Calculation. Then, the third block of the calculation is an Object Velocity Calculation. It contains code to calculate the coordinates relative speed values V_{dopp} . Therefore, a distribution of the vector velocity V_{dopp} on the components of X and Y depends on the angle thetaTIS. Output values are V_{dopp_x} and V_{dopp_y} .

4.6.7. Part II: Verification. The second part of the prepared program was verification part. Before modelling BLIS system, it had to be checked whether calculations delivering the required input values to the system are correct. For this purpose, the values were compared with values generated by the simulation environment (PreScan).

Verification of the relative speed values V_{dopp} : relative speed V_{dopp} is called the Doppler Velocity, which results in the fact that its value is not always unequivocal from the expected one. In order to check the difference between the variable V_{dopp} and the actual vehicle speeds difference verification program was elaborated. The example results can be seen in Figure 9.

Verification of X and Y coordinates relative position of the object is as follows.

To verify the calculations of the value of X and Y coordinates relative position of the object (relative to the Bytel vehicle), the computed variables were compared with variables X_r and Y_r with the difference of the values of the coordinates of the vehicle Bytel (X_v, Y_v) and the object (X_o, Y_o) obtained artificially by PreScan software, taking into account the offset point to be the center of the coordinate system of the vehicle, to whom they are given the coordinates of absolute position.

Verification of coordinates X and Y absolute position of the object is as follows. In order to verify the calculation of the X and Y coordinates of the relative position of the object (relative to the Bytel vehicle), the computed variables X_{ob} and Y_{ob} were compared with the object position coordinates (X_o, Y_o) obtained artificially from the PreScan software.

4.6.8. Part III: BLIS System. The last step was to build the logics of the BLIS system. This part of the system consists of

three parts: Data, Alert Range Assignment, and Alert Colour Assignment.

Data. This section of the BLIS window contains ordered model input values, blocks of recording of signals to the workspace.

Alert Range Assignment. The part called Alert Range Assignment is responsible for assigning to the appropriate areas of the alarm (LL, L, C, R, RR) angular ranges in which the object appears.

- (i) Range LL is assigned for $\theta_{TIS} \langle 48; 87.5 \rangle$ deg.
- (ii) Range L is assigned for $\theta_{TIS} \langle 7; 49 \rangle$ deg.
- (iii) Range C is assigned for $\theta_{TIS} \langle -8; 8 \rangle$ deg.
- (iv) Range R is assigned for $\theta_{TIS} \langle -49; -7 \rangle$ deg.
- (v) Range RR is assigned for $\theta_{TIS} \langle -87.5; -48 \rangle$ deg.

Alert Assignment Colour. Code fragment entitled Alert Colour Assignment is responsible for assigning the alarm corresponding colour in accordance with the degree of threat, yellow or red.

Decisions on the colour of the alarm are taken on the basis of three pieces of information:

- (i) Comparison of the position of the object relative to the value of YCD (Yellow Critical Distance) or RCD (Red Critical Distance), which are functions defining the threshold value of position of the object in the x axis, for which there is an increased risk of collision
- (ii) Values V_{dopp}
- (iii) The value of TTC (Time To Collision)

The YCD function (Yellow Critical Distance) is given by

$$YCD = -4 * |Y_r| + 12. \quad (1)$$

The RCD (Red Critical Distance) is given by

$$RCD = -\frac{4}{3} * |Y_r| + 4. \quad (2)$$

Yellow alarm occurs when the following conditions are met:

$$(X_r \leq YCD \text{ (yellowFun)} \cap X_r \neq 0 \cap V_{dopp} \leq 0) \cup 4 < TTC \leq 10. \quad (3)$$

Red alarm occurs when the following conditions are met:

$$(((X_r \leq RCD \text{ (redFun)} \cap X_r \neq 0) \cup 0 < X_r \leq 1) \cap V_{dopp} \leq 0) \cup 0, 1 < TTC \leq 4. \quad (4)$$

Figure 10 visualizes the conditions for activating the alarms.

- X_r : X coordinate of relative position of the object
- YCD: Yellow Critical Distance
- RCD: Red Critical Distance
- V_{dopp} : relative velocity of the object
- TTC: Time To Collision

4.7. System Model: Control System. Having data processing system, we can begin to design the control system of warning LEDs. In the previous section, signals have been mentioned that we receive at the output of the processing unit. The system of transmission between the BLIS system and the LED is shown in Figure 11 and is the same for each of the diodes.

On the output from the Bytel vehicle block, we have seven binary signals:

- (i) yellowAlert: the state of yellow alarm
- (ii) redAlert: the state of red alarm

- (iii) alertRR: the state of alarm for the RR range
- (iv) alertR: the state of alarm for the R range
- (v) alertC: the state of alarm for the C range
- (vi) alertL: the state of alarm for the L range
- (vii) alertLL: the state of alarm for the LL range

At the input of each of the LEDs blocks, the following signals are given:

- (i) yellowAlert: state of yellow alarm
- (ii) redAlert: state of red alarm
- (iii) alertX: state of alarm for the corresponding diode range (RR, R, C, L, and LL)

For each of the five LEDs used to display warning information, the same control was developed.

4.7.1. Part I: System Activation. The logics of the system is based on a switch block which, depending on the condition of the simultaneous occurrence of any of the alarms and



FIGURE 10: Logic of the red and yellow alarm.

the activation of the range corresponding to LED (in this case the LL range), transmits a signal specifying the colour of the alarm (equivalent to the activation of the alarm) or transmits the signal from the default, LEDs gray colour. The same condition checked in switch block acts as a trigger of alertColour block.

4.7.2. Part II: Determination of Alarm Colour. The condition mentioned above checked in switch block acts simultaneously as a trigger of alertColour block.

In the case where you get any of the alarms (yellowAlarm or redAlarm) and at the same time the object was detected in the corresponding range, the block alertColour is executed. This block is to assign a diode suitable for the degree of danger colour: the colour red for a higher degree of threat and the yellow colour for a lower degree of threat. In the case where there is information about the appearance of red alert and activating the appropriate range, signal carrying information about the red alarm is passed, if not, yellow. This logic makes the red alarm the priority.

5. Testing

The test phase of the design process is often treated in a disrespectful way, but properly planned and carried out tests allow you to determine the imperfections and errors of the project very accurately and at the same time to facilitate their improvement greatly. During test planning, the first step is to verify the correctness of the calculations made in the model. In order to determine what we should investigate properly, it is necessary to answer the following questions:

- (i) What are the inputs to the model?
- (ii) What are the outputs from the model?
- (iii) How outputs were from the model obtained?
- (iv) Which of these activities relate to the independent variables of the operation of the system, which are the physical representation of unknown values?

Answering these questions will allow the identification of transformation, which should be explored in terms of their compliance with the physical values that do not belong to a set of values of the input model. PreScan software provides many of these values, which makes the comparison of the model with the actual data possible without an actual field testing.

The next step in the test planning is to return to the design assumptions and answer the questions:

- (i) What is the designed system supposed to do? What are the requirements?

- (ii) What information does the system provide the user with?
- (iii) What should not the system do?

The above questions relate to the direct effects of the system, its functionality, and purpose but they do not specify requirements regarding the project completely. The tests are considered for remaining elements, which are invisible to the user and dependent largely on the designer of the system. These elements are the basis of the system. In order to determine them, it is necessary to answer the following questions:

- (i) How should the designed system operate?
- (ii) What inputs and outputs should it generate/set?
- (iii) What functions are its main characteristics?

The answers to these questions should be clarified on the basis of the design phase, the software built, and the expected behaviour of a system. Having already collected information about the system sorted out according to the above scheme, it is possible to begin to plan test case studies. What is most important in the operation of the safety system is its correct operation and efficiency. If the system is to comply with the quite simple functions (e.g., warn the driver about a negative temperature), it is not necessary to develop complex test scenarios.

The case is complicated, however, when the system has a more extensive set of functionalities, when it has to react differently in different situations, when the output value depends on the value of larger quantities of inputs, and where there is the possibility of false alarms. In this case, it is good to previously develop scenarios to test and test cases for each scenario. After tests, a summary of the verification of the correctness of the system should be made. In the case where the system does not meet one (or several) of the designated test case studies, it is necessary to carry out repair procedures. This phase resembles the design phase, but instead of building the system from the beginning, repairs are carried out and changes to the existing system on the basis of data obtained in the test phase are introduced. Well-conducted tests allow us to identify errors in the system quickly and often arise the way for solutions. In this chapter, the test phase for BLIS system design for the Bytel vehicle will be described.

5.1. Tests Plan

5.1.1. Calculations Verification. Based on the previously elaborated flowchart, the first step of the procedure of Blind Spot Information System test planning for Bytel vehicle is the answer to the question about the correctness of the calculations implemented in the model.

- (i) What are inputs to the model?

In the model, the following inputs are used:

- (a) X_v , [m]: X coordinate of Bytel vehicle in absolute coordinate system

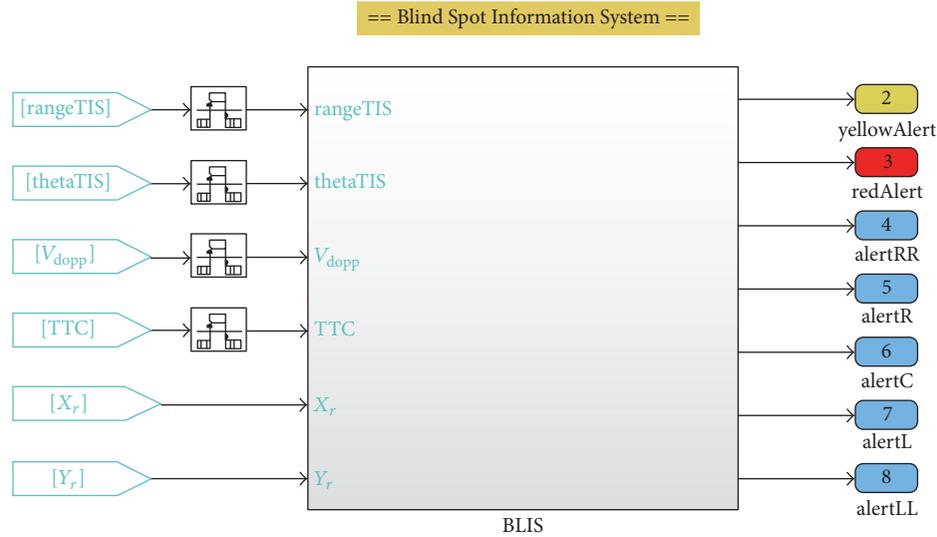


FIGURE 11: BLIS block.

- (b) Y_v [m]: Y coordinate of Bytel vehicle in absolute coordinate system
- (c) vehicleRot [$^\circ$]: rotation angle related to z axis of Bytel vehicle
- (d) V_v [m/s]: Bytel vehicle velocity
- (e) rangeTIS [m]: distance from the sensor to detected object
- (f) V_{dopp} [m/s]: relative vehicles velocity
- (g) thetaTIS [$^\circ$]: angle at which obstacle was detected

(ii) What are the outputs of model?

In the model, the following outputs are generated:

- (a) yellowAlert (0/1): yellow alarm status
- (b) redAlert (0/1): red alarm status
- (c) alertLL (0/1): LL alarm status
- (d) alertL (0/1): L alarm status
- (e) alertC (0/1): C alarm status
- (f) alertR (0/1): R alarm status
- (g) alertRR (0/1): RR alarm status

(iii) How are the above outputs generated?

Decisions on the classification of the alarm due to the level of risk (yellow or red) system are taken on the basis of three pieces of information:

- (a) Comparison of relative position of object with value of YCD (yellow Critical Distance) or RCD (Red Critical Distance), which are functions determining threshold values of object along x axis for which increased risk of collision exists
- (b) V_{dopp} value
- (c) TTC value (Time To Collision)

Decision on selection of the area, which is assigned to an alarm, depends only on the value of the angle thetaTIS.

- (iv) Which of these activities relate to the independent variables from the operation of the system and which are the physical representation of unknown values?

Variable determined by calculation are as follows:

- (a) X_r : X coordinate of object position related to Bytel vehicle determined on the basis of values of rangeTIS and thetaTIS
- (b) Y_r : Y coordinate of object position related to Bytel vehicle determined on the basis of values of rangeTIS and thetaTIS
- (c) Yellow Critical Distance: function determined on the basis of Y_r value
- (d) Red Critical Distance: function determined on the basis of Y_r value
- (e) Time To Collision: time to collision determined on the basis of rangeTIS and V_{dopp} values

Based on this information, it was decided to examine the above-mentioned dependence by comparing them with the values of the corresponding variables artificially generated by the PreScan software. An example of such a comparison can be seen in Figure 12.

5.1.2. Verification from the User Perspective: Visual Verification. The next step is to answer the questions about system importance from the perspective of the user (Figures 13 and 14).

- (i) What should the designed system do?

The purpose of the system is to inform the user of the location of another vehicle behind the driven one, which is not visible to the driver, or its notification

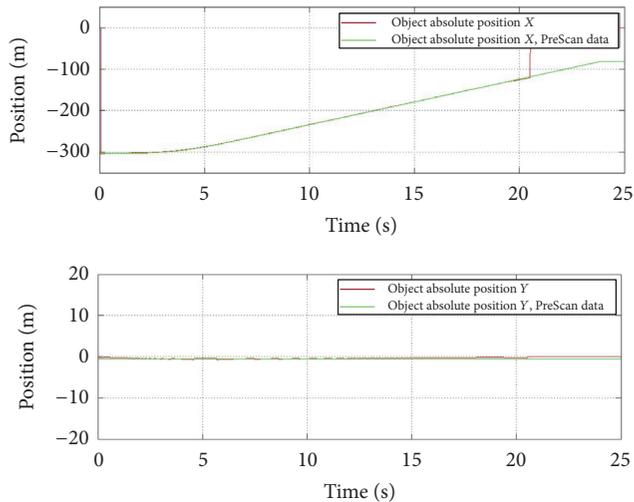


FIGURE 12: Comparison of corresponding variables determined by virtual BLIS system and directly by PreScan.

requires too much effort of the driver. The information should be communicated in a visual way, using LEDs.

- (ii) What information should the system provide the user with?

System is to inform the user of the observation of the vehicle in one (or more) of the five ranges designated for the driven vehicle and the degree of risk associated with the event. For low-level threats, the alarm should accept yellow colour and for high threat red.

- (iii) What should not the system do?

The system should not react to vehicles in the area of the sensor, but in a place visible to the driver of the vehicle. The system also should not generate false positives: that is, the alarm is not caused by the appearance of the object in the field of sight of the driver.

5.1.3. Numerical Verification. The next step is to analyse important questions from the perspective of the creators of the system (Figure 16):

- (i) How the designed system should operate?

The system is intended to operate on the basis of information from the vehicle (vehicle location, vehicle speed), and the information from the sensor (the distance of the detected object from the sensor, the angle at which the object appears, Doppler speed). The system should set, on the basis of the data, relative location information of vehicles to each other and the speed difference between vehicles. The system should assign one of the five predesignated areas to a given position. Based on the information about the speed difference between vehicles and their relative position system, it should set TTC (Time To Collision). Depending on the value of TTC and the distance between the vehicles, the system should assign areas

of the degree of risk: no risk, the LED is not lit; low threat, LED yellow; high risks, the LED lights red.

- (ii) What input and output values should it take?

The system should set the following inputs and outputs:

- Distance of detected object from driven vehicle: 0–40 [m]
- Velocity of driven vehicle: 0–15 [m/s]
- Relative velocity of vehicles: 0–15 [m/s]
- Position of the vehicle on x , y axis [m]
- Angle of appearance of object: $-87,5^\circ$ – $87,5^\circ$

The system should generate the following output values:

- Value 0/1 on output corresponding to LL, L, C, R, and RR LEDs
- Value 0/1 on output corresponding to yellow alarm
- Value 0/1 on output corresponding to red alarm

- (iii) What functions characterize it?

- TTC (Time To Collision) for every moment when object is within the sensor range
- RangeTIS for every moment when object is in the sensor range
- Relative object and Bytel vehicle position: X_r, Y_r
- ThetaTIS angle, in which the object detected remains for each moment of time in which the object is located within the sensor range
- Relative object and Bytel vehicle velocity

5.1.4. The Development of a Set of Test Cases. Based on the information collected, Table 1 was prepared containing test cases, which were used to verify the operation of the system BLIS.

5.2. Test Case Scenario: Vehicle Overtaken from the Left Side

5.2.1. Description of the Scenario. The object comes from the left side of the vehicle Bytel. The ratio of speed of an object to Bytel vehicle speed is in the range of 0 to 1.5 for each moment of virtual experiment. The speed difference between the object and the Bytel vehicle is not greater than 10 m/s. The speed of the object and the vehicle does not exceed 12 m/s (Figures 13, 14, and 15).

5.2.2. Results Verification. In order to investigate the above-mentioned cases, we have to use different methods. For testing cases (1) (Table 1) and (3) (Table 1), it was necessary to analyse the relevant variables for cases (2) (Table 1) and visual inspection simulation is sufficient.

Calculation Verification. In order to check the correctness of the algorithms implemented in the BLIS system model comparative charts were used comparing variables with variables generated by the PreScan software.

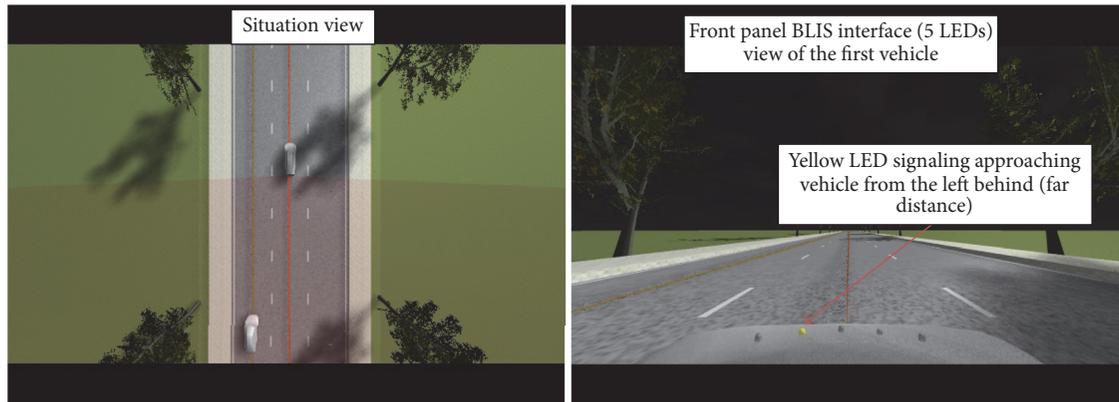


FIGURE 13: Example of visual verification, scenario 1, overtaking phase 1.

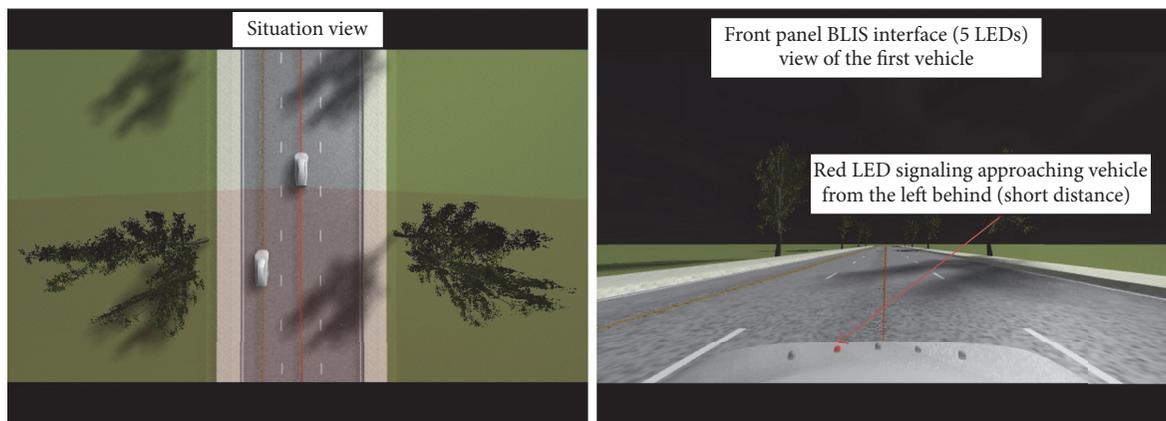


FIGURE 14: Example of visual verification, scenario 1, overtaking phase 2.

- (i) Comparison of V_{dopp} speed with the difference in velocity of the Bytel vehicle and the object overtaking Bytel; both speeds are available in the PreScan software.

Based on a comparison graph of the Doppler Velocity and difference of vehicles velocities, it can be concluded that the mapping of Doppler Velocity does not fully coincide with the actual vehicles speeds difference. The biggest differences can be seen for the time intervals: 0–2 s and 16–19 s. This is connected with the moment of occurrence of the object in a short distance from the sensor. Despite these cases, it is concluded that the proposed function of the velocity can be used in the design of system.

- (ii) Comparison of the relative position in the x and y axis of overtaking object with respect to the Bytel vehicle, calculated using information about the distances of overtaking object and the angle in which it is located, with a difference of location coordinates of the vehicle and the object with a leading Bytel; coordinates are available in the PreScan software.

Based on the comparison charts X and Y calculated using data on the distance and angle of the object

overtaking with the actual difference in the position of the Bytel vehicle and the overtaking object, it was found that the adopted model calculation of the relative position of the object is correct. In addition, the graphs present compliance of variables only until reaching 20th second. It is associated with the period when the object is within the range of the sensor, which enabled the calculation of its relative position.

- (iii) Comparison of the absolute position of the x and y axis overtaking object calculated using information about the distances of overtaking object, the angle at which it is located and the coordinates of the Bytel vehicle, the absolute position of the object, which is made available in the PreScan software.

Based on the comparison charts X and Y calculated using data on the distance, angle and coordinates of the object overtaking the Bytel vehicle with the actual position of the absolute object, it was found that the adopted computational model of the absolute position of the object is correct. The graphs show compliance only until reaching 20th second. It is associated with the period when the object is within the range of the sensor, which enabled the calculation of its relative position.

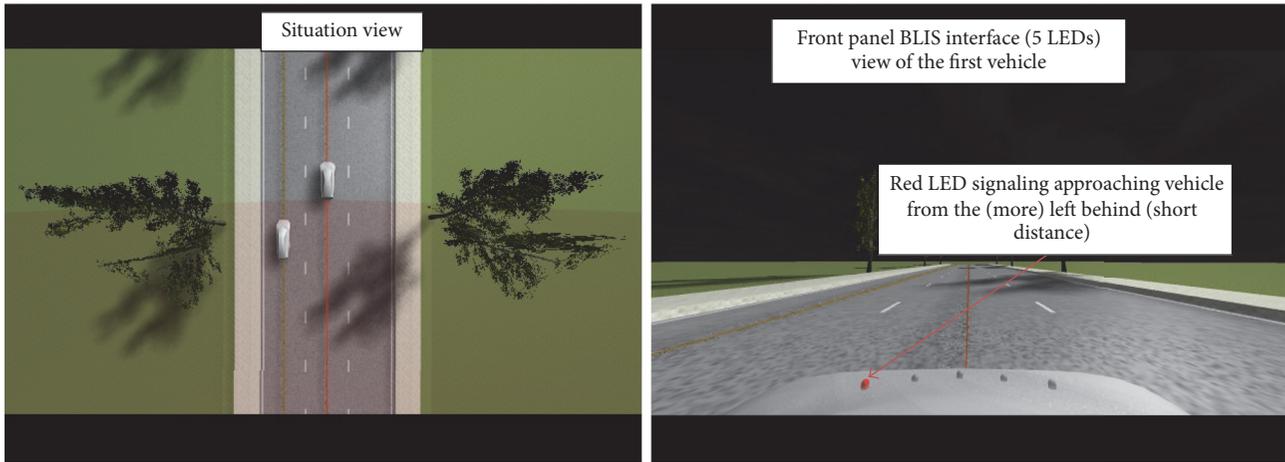


FIGURE 15: Example of visual verification, scenario 1, overtaking phase 3.

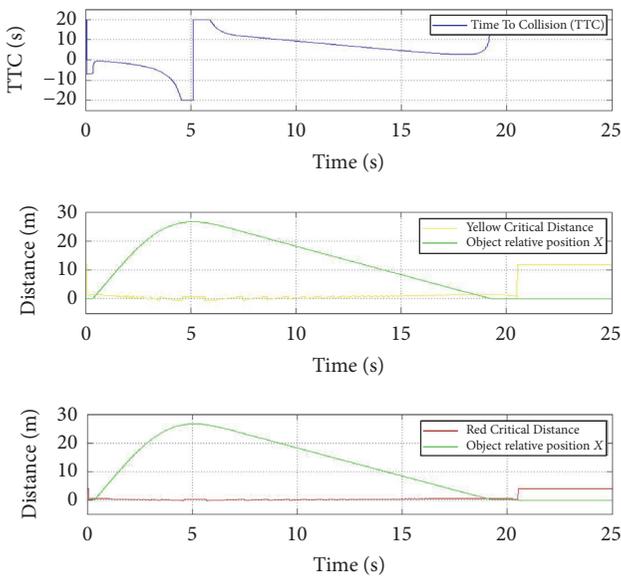


FIGURE 16: TTC (Time To Collision), comparison of YCD (Yellow Critical Distance), and RCD (Red Critical Distance) with object relative position X.

Visual Verification (Figures 13, 14, and 15)

- (i) The object is coming from the direction established in the description of the scenario: object is coming from the left.

Based on the simulation results, the system has been found to meet this condition.

- (ii) The object is located within the sensor range: it was possible to carry out tests of the BLIS system and the object must be in range of the sensor.

Based on the simulation results, the system has been found meeting the above condition.

- (iii) Operation of warning LL LEDs, since the object is overtaking on the left side of the Bytel vehicle after fulfilment of additional conditions LED should light

up depending on the object distances in red and/or yellow.

Based on the simulation results, the system has been found meeting the above condition.

- (iv) Operation of warning L LEDs, since the object is overtaking on the left side of the Bytel vehicle after fulfilment of additional conditions LED should light up depending on the object distances in red and/or yellow.

Based on the simulation results, the system has been found to meet the above condition.

- (v) Operation of warning C LED, since the overtaking object at any moment in time is located centrally behind the Bytel vehicle the LED should not light up.

Based on the simulation results, the system has been found to meet the above condition.

- (vi) Operation of warning R LED, since the overtaking object at any moment in time is located from the right side of Bytel vehicle the LED should not light up.

Based on the simulation results, the system has been found to meet the above condition.

- (vii) Operation of warning RR LED, since the overtaking object at any moment in time is located from the right side of Bytel vehicle, the LED should not light up.

Based on the simulation results, the system has been found to meet the above condition.

Numerical Verification

- (i) Inspection of Bytel vehicle velocity:

According to the assumptions of this scenario, Bytel vehicle velocity at any moment in time should not be greater than 12 m/s. Based on the velocity chart of the Bytel vehicle the system has been found to meet this assumption.

- (ii) Inspection of overtaking object velocity:

TABLE 1: Test cases used to verify the operation of the BLIS system.

Number	Test description	Initial conditions	Expected value	Outcome
(1)	Calculation verification			
(1.1)	X_r position	$\text{rangeTIS} \neq 0$	$X_r \cong X_v - X_o$	
(1.2)	Y_r position	$\text{rangeTIS} \neq 0$	$Y_r \cong Y_v - Y_o$	
(1.3)	Yellow Critical Distance	$\text{rangeTIS} \neq 0$	$\text{YCD} = -4 * \text{abs}(Y_r) + 12$	
(1.4)	Red Critical Distance	$\text{rangeTIS} \neq 0$	$\text{RCD} = -4/3 * \text{abs}(Y_r) + 4$	
(1.5)	Time To Collision	$V_{\text{dopp}} \neq 0$	$\text{TTC} = -\text{rangeTIS}/V_{\text{dopp}}$	
(2)	Visual tests			
(2.1)	Object approaches from direction assumed in the scenario description	-	Left/right side, behind the object	
(2.2)	Object is in the sensor range	-	The object is in the sensor range	
(2.3)	Correct functioning of the LL warning LED	Object in the LL area, danger	LED lights up Y/R	
(2.4)	Correct functioning of the L warning LED	Object in the L area, danger	LED lights up Y/R	
(2.5)	Correct functioning of the C warning LED	Object in the C area, danger	LED lights up Y/R	
(2.6)	Correct functioning of the R warning LED	Object in the R area, danger	LED lights up Y/R	
(2.7)	Correct functioning of the RR warning LED	Object in the RR area, danger	LED lights up Y/R	
(3)	Computational tests			
(3.1)	Object velocity/Bytel vehicle velocity	$V_v \neq 0$	≥ 1	
(3.2)	Object velocity/Bytel vehicle velocity	$V_v \neq 1$	$\leq 10 \text{ m/s}$	
(3.3)	Object velocity	-	$\leq 12 \text{ m/s}$	
(3.4)	Bytel vehicle velocity	-	$\leq 10 \text{ m/s}$	
(3.5)	Turning on yellow alert	$4 \leq \text{TTC} \leq 10 \cup (X_r \leq \text{YCD} \cap V_{\text{dopp}} < 0 \cap X_r \neq 0)$	$\text{yellowAlert} = 1$	
(3.6)	Turning on red alert	$0.1 \leq \text{TTC} \leq 4 \cup (X_r \leq \text{RCD} \cap V_{\text{dopp}} < 0 \cap X_r \neq 0)$	$\text{redAlert} = 1$	
(3.7)	LL area control	$48 \leq \text{thetaTIS} \leq 87.5$	$\text{alertLL} = 1$	
(3.8)	L area control	$7 \leq \text{thetaTIS} \leq 49$	$\text{alertL} = 1$	
(3.9)	C area control	$8 \leq \text{thetaTIS} \leq -8$	$\text{alertC} = 1$	
(3.10)	R area control	$-49 \leq \text{thetaTIS} \leq -7$	$\text{alertR} = 1$	
(3.11)	RR area control	$-87.5 \leq \text{thetaTIS} \leq -48.5$	$\text{alertRR} = 1$	

According to the assumptions of this scenario overtaking vehicle velocity at any moment in time should not be greater than 12 m/s. Based on the velocity chart of the overtaking vehicle the system has been found to meet this assumption.

(iii) Inspection of object and Bytel vehicle velocities ratio:

According to the assumptions of this scenario, ratio of velocities of object to Bytel vehicle velocity at any moment in time should not be less than 1. Based on the velocities ratio chart, the system has been found to meet this assumption.

(iv) Inspection of object and Bytel vehicle velocities difference:

According to the assumptions of this scenario, the difference of overtaking vehicle velocity and Bytel vehicle velocity at any moment in time should not be greater than 10 m/s. Based on the vehicles velocities difference chart system has been found to meet this assumption.

(v) Inspection of alarm activation:

In order to inspect correctness of activation of yellow and red alarm, the following charts were used: TTC (Time To Collision), YCD (Yellow Critical Distance), and RCD (Red Critical Distance) compared to X_r (relative position of object in x axis). Charts are shown in Figure 16.

(vi) Inspection of yellow alarm activation:

According to the assumptions of the design, the yellow alarm should be activated under the following conditions:

- (a) TTC (Time To Collision) ranges from 4 to 10 s or
- (b) the value of the relative position of the object in the x -axis representing the longitudinal axis of the Bytel vehicle is less than the value of the function YCD (Yellow Critical Distance).

Based on the charts of relative position of the object in the x -axis and functions of Yellow Critical Distance, charts of TTC and yellow alarm activity system have been found to meet this assumption.

- (vii) Inspection of red alarm activation:

According to the assumptions of the design, the red alarm should be activated under the following conditions:

- (a) TTC (Time To Collision) ranges from 0,1 to 4 s or
- (b) the value of the relative position of the object in the x -axis representing the longitudinal axis of the Bytel vehicle is less than the value of the function RCD (Red Critical Distance)

Based on the charts of relative position of the object in the x -axis and functions of Red Critical Distance, the charts of TTC and red alarm activity system have been found to meet this assumption.

- (viii) Inspection of operation of LL alarm corresponding to extreme left scanning area:

As designed LL alarm should be activated when the vehicle is detected in an angular range from 48° to 87.5° . Based on the charts, TIS sensor range, the angle of appearance of the object, and graph LL alarm activity system have been found to meet this assumption.

- (ix) Inspection of operation of L alarm corresponding to left scanning area:

As designed L alarm should be activated when the vehicle is detected in an angular range from 7° to 49° . Based on the charts, TIS sensor range, the angle of appearance of the object, and graph L alarm activity system have been found to meet this assumption.

- (x) Inspection of operation of C alarm corresponding to central scanning area:

As designed, C alarm should be activated when the vehicle is detected in an angular range from -8° to 8° . Based on the charts, TIS sensor range, the angle of appearance of the object, and graph C alarm activity system have been found to meet this assumption.

- (xi) Inspection of operation of R alarm corresponding to right scanning area:

As designed R alarm should be activated when the vehicle is detected in an angular range from -49° to -7° . Based on the charts, TIS sensor range, the angle of appearance of the object, and graph R alarm activity system have been found to meet this assumption.

- (xii) Inspection of operation of RR alarm corresponding to extreme right scanning area:

As designed RR alarm should be activated when the vehicle is detected in an angular range from -87.5° to -48° . Based on the charts, TIS sensor range, the angle of appearance of the object, and graph RR alarm activity system have been found to meet this assumption.

5.2.3. Final Results of Virtual Verification. Based on the above results, Table 1 of test cases was completed, and it was found that tests for scenario I were completed successfully as can be seen in Table 2.

6. Consideration of Possible Improvements

Although the final result of applying the described method was satisfying, there were still areas for the improvement. First of all there should be stronger emphasis put on the iterative approach to design, combining design phase with testing phase.

In this context, it is important to choose the scenarios of simulation tests carefully and, on the basis of subsequent experience, expand the list of these scenarios to the new situations observed in the race conditions.

What also would be of a great value is real-life implementation system and comparison of real-life values with outputs received from simulation. This could help to identify the areas where the described method does not prove correct or needs improvement.

Similarly, as TASS, PreScan enables also creating experiments for various weather conditions and it would be valuable to test the designed system for different atmospheric conditions. It is especially true for systems that assume using sensors which output values are weather-dependent (e.g., sonars, Lidars).

There is considerable potential for the transfer of work results related to the BLIS system as well as its methodology for commercial development. The particular advantage of the developed system is its modularity and the ability to adapt to other types of vehicle and other operating conditions.

7. Conclusions

The described method of Advanced Driver Assistance System design, with the use of virtual prototyping tool, has been applied in real-life project. The outcome of taking this approach can be evaluated as successful, since the reasons for using this kind of method have been proven and the final results have been considered satisfying.

Using simulation method in ADAS design shortens work time, makes the iterative approach to design easier and

TABLE 2: Approved results of test cases.

Number	Test description	Initial conditions	Expected value	Outcome
(1)	Calculation verification			Approved
(1.1)	X_r position	rangeTIS $\neq 0$	$X_r \cong X_v - X_o$	Approved
(1.2)	Y_r position	rangeTIS $\neq 0$	$Y_r \cong Y_v - Y_o$	Approved
(1.3)	Yellow Critical Distance	rangeTIS $\neq 0$	$YCD = -4 * \text{abs}(Y_r) + 12$	Approved
(1.4)	Red Critical Distance	rangeTIS $\neq 0$	$RCD = -4/3 * \text{abs}(Y_r) + 4$	Approved
(1.5)	Time To Collision	$V_{dopp} \neq 0$	$TTC = -\text{rangeTIS}/V_{dopp}$	Approved
(2)	Visual tests			Approved
(2.1)	Object approaches from direction assumed in the scenario description	-	Left/right side, behind the object	Approved
(2.2)	Object is in the sensor range	-	The object is in the sensor range	Approved
(2.3)	Correct functioning of the LL warning LED	Object in the LL area, danger	LED lights up Y/R	Approved
(2.4)	Correct functioning of the L warning LED	Object in the L area, danger	LED lights up Y/R	Approved
(2.5)	Correct functioning of the C warning LED	Object in the C area, danger	LED lights up Y/R	Approved
(2.6)	Correct functioning of the R warning LED	Object in the R area, danger	LED lights up Y/R	Approved
(2.7)	Correct functioning of the RR warning LED	Object in the RR area, danger	LED lights up Y/R	Approved
(3)	Computational tests			Approved
(3.1)	Object velocity/Bytel vehicle velocity	$V_v \neq 0$	≥ 1	Approved
(3.2)	Object velocity/Bytel vehicle velocity	$V_v \neq 1$	≤ 10 m/s	Approved
(3.3)	Object velocity	-	≤ 12 m/s	Approved
(3.4)	Bytel vehicle velocity	-	≤ 10 m/s	Approved
(3.5)	Turning on yellow alert	$4 \leq TTC \leq 10 \cup (X_r \leq YCD \cap V_{dopp} < 0 \cap X_r \neq 0)$	yellowAlert = 1	Approved
(3.6)	Turning on red alert	$0.1 \leq TTC \leq 4 \cup (X_r \leq RCD \cap V_{dopp} < 0 \cap X_r \neq 0)$	redAlert = 1	Approved
(3.7)	LL area control	$48 \leq \text{thetaTIS} \leq 87.5$	alertLL = 1	Approved
(3.8)	L area control	$7 \leq \text{thetaTIS} \leq 49$	alertL = 1	Approved
(3.9)	C area control	$8 \leq \text{thetaTIS} \leq -8$	alertC = 1	Approved
(3.10)	R area control	$-49 \leq \text{thetaTIS} \leq -7$	alertR = 1	Approved
(3.11)	RR area control	$-87.5 \leq \text{thetaTIS} \leq -48.5$	alertRR = 1	Approved

faster, enables early error detection and identification, and encourages parallel work. Therefore, it can be stated that using this method results also in costs reduction and system quality improvement.

The elimination of the need to verify the system in practice is due to the use of algorithms that verify the simulation results without allowing uncritical reliance on simulation results solely.

The high reliability of the BLIS system itself was achieved by taking into account not only the position of the adjacent vehicles, but also their direction of travel as well as the relative speed of the vehicles. This made it possible to calculate the potential collisions and predict the time it could take place (TTC). This approach eliminates false alarms of

most nonroad traffic situations, despite the relatively close proximity of the vehicle to other objects on the track or in the immediate vicinity.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] D. Ammon and W. Schiehlen, "Advanced road vehicles: Control technologies, driver assistance," in *Dynamical Analysis of Vehicle Systems. Theoretical Foundations and Advanced Applications*, W. Schiehlen, Ed., pp. 283–304, Springer, New York, NY, USA, 2009.

- [2] F. Coşkun, Ö. Tuncer, E. Karşlilgil, and L. Güvenç, "Vision system based lane keeping assistance evaluated in a hardware-in-the-loop simulator," in *Proceedings of the ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis, ESDA2010*, pp. 103–114, Istanbul, Turkey, July 2010.
- [3] O. J. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "Development of a driver information and warning system with vehicle hardware-in-the-loop simulations," *Mechatronics*, vol. 19, no. 7, pp. 1091–1104, 2009.
- [4] A. D. Hall, *A methodology for Systems Engineering*, Van Nostrand, 1962.
- [5] A. Belbachir, J. Smal, J. Blossville, and D. Gruyer, "Simulation-driven validation of advanced driving-assistance systems," *Transport Research Arena – Europe*, vol. 48, pp. 1205–1214, 2012.
- [6] K. Jezierska-Krupa and W. Skarka, "Using Simulation Method for Designing ADAS Systems for Electric Vehicle. Proceedings of the 23rd ISPE Inc. International Conference on Transdisciplinary Engineering Location: Fed Univ Technol, Curitiba, Brazil, October, 2016," in *Transdisciplinary Engineering: Crossing Boundaries*, M. Borsato, N. Wognum, and M. Peruzzini, Eds., vol. 4 of *Advances in Transdisciplinary Engineering*, pp. 595–604, 2016.
- [7] E. Hull, K. Jackson, and J. Dick, *Requirements Engineering*, Springer Science & Business Media, 2005.
- [8] G. Pahl and W. Beitz, *Engineering Design: A Systematic Approach*, Design Council/Springer, 1988.
- [9] W. Schiehlen and D. Ammon, *Advanced Road Vehicles: Control Technologies, Driver Assistance*, Institute of Engineering and Computational Mechanics, University of Stuttgart, Stuttgart, Germany.
- [10] B.-F. Wu, H.-Y. Huang, C.-J. Chen, Y.-H. Chen, C.-W. Chang, and Y.-L. Chen, "A vision-based blind spot warning system for daytime and nighttime driver assistance," *Computers and Electrical Engineering*, vol. 39, no. 3, pp. 846–862, 2013.
- [11] A. Cholewa, W. Skarka, K. Sternal, and M. Targosz, "Electric vehicle for the students' shell eco-marathon competition. design of the car and telemetry system," in *Telematics in the Transport Environment, Proceedings of the 12th International Conference on Transport Systems Telematics, TST 2012, Katowice-Ustroń, Poland, October, 2012*, J. Mikulski, Ed., Communications in Computer and Information Science, pp. 26–33, Springer, Berlin, Germany.
- [12] W. Skarka, "Reducing the Energy Consumption of Electric Vehicles," in *Transdisciplinary Lifecycle Analysis of Systems, Proceedings of the 22nd ISPE-Inc International Conference on Concurrent Engineering Location: Delft Univ Technol, Delft, Netherlands, July, 2015*, R. Curran, N. Wognum, M. Borsato et al., Eds., vol. 2 of *Advances in Transdisciplinary Engineering*, pp. 500–509, IOS Press, Amsterdam, Netherlands, 2015.
- [13] Shell, Shell Eco-marathon, referenced at: <https://www.shell.com/energy-and-innovation/shell-ecomarathon.html>.
- [14] TASS International, *PreScan: Simulation of ADAS and Active Safety*, 2015, <https://www.tassinternational.com/prescan>.
- [15] A. Jałowicki and W. Skarka, "Modeling in Ultra-Efficient Vehicle Design. Proceedings of the 23rd ISPE Inc. International Conference on Transdisciplinary Engineering Location: Fed Univ Technol, Curitiba, Brazil October, 2016," in *Transdisciplinary Engineering: Crossing Boundaries*, M. Borsato, N. Wognum, M. Peruzzini et al., Eds., vol. 4 of *Advances in Transdisciplinary Engineering*, pp. 999–1008, 2016.
- [16] W. Skarka and M. Wąsik, "Aerodynamic Features Optimization of Front Wheels Surroundings for Energy Efficient Car. Proceedings of the 23rd ISPE Inc. International Conference on Transdisciplinary Engineering, Fed Univ Technol, Curitiba, Brazil, October, 2016," in *Transdisciplinary Engineering: Crossing Boundaries*, M. Borsato, N. Wognum, M. Peruzzini et al., Eds., vol. 4 of *Advances in Transdisciplinary Engineering*, pp. 483–492, 2016.
- [17] LTD Hokuyo Automatic Co. Scanning laser range finder utm-30lx-ew specification. 2012.
- [18] K. Cichoński and W. Skarka, "Innovative Control System for High Efficiency Electric Urban Vehicle. Proceedings of the 15th International Conference on Transport Systems Telematics (TST): Wrocław University of Science and Technology, Wrocław, Poland, April, 2015," in *Mikulski, J.*, OF. TOOLS, Ed., vol. 531 of *Communications in Computer and Information Science*, pp. 121–130, 2015.
- [19] S. A. Rodríguez Flórez, V. Frémont, P. Bonnifait, and V. Cherfaoui, "Multi-modal object detection and localization for high integrity driving assistance," *Machine Vision and Applications*, vol. 25, no. 3, pp. 583–598, 2014.

Research Article

Neural Network Control System of UAV Altitude Dynamics and Its Comparison with the PID Control System

Jemie Muliadi and Benyamin Kusumoputro 

Department of Electrical Engineering, Universitas Indonesia, Kampus Baru UI, Depok 16424, Indonesia

Correspondence should be addressed to Benyamin Kusumoputro; kusumo@ee.ui.ac.id

Received 5 May 2017; Revised 25 July 2017; Accepted 16 August 2017; Published 22 January 2018

Academic Editor: Cheng S. Chin

Copyright © 2018 Jemie Muliadi and Benyamin Kusumoputro. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article proposes a comparative method to assess the performance of artificial neural network's direct inverse control (DIC-ANN) with the PID control system. The comparison served as an analysis tool to assess the advantages of DIC-ANN over conventional control method for a UAV attitude controller. The development of ANN method for UAV control purposes arises due to the limitations of the conventional control method, which is the mathematical based model, involving complex expression, and most of them are difficult to be solved directly into analytic solution. Although the linearization simplified the solving process for such mathematical based model, omitting the nonlinear and the coupling terms is unsuitable for the dynamics of the multirotor vehicle. Thus, the DIC-ANN perform learning mechanism to overcome the limitation of PID tuning. Therefore, the proposed comparative method is developed to obtain conclusive results of DIC-ANN advantages over the linear method in UAV attitude control. Better achievement in the altitude dynamics was attained by the DIC-ANN compared to PID control method.

1. Introduction

The advancement of the Unmanned Aerial Vehicle (UAV) has triggered the development of the autonomous technology for various assignments, from the applied strategic and tactical mission to the simple carrier of the video shooting camera and, recently, the SAR purposes [1, 2]. Meanwhile, in the field of autonomous technology, UAVs already become the preferred platform for research and development of the flight control systems. Numerous types of UAV have been constructed and developed for research platform, and between those types, the quadrotor becomes the popular type. The popularity emerged because of the construction's simplicity, the easiness of maintenance, their ability to hover, and the capability of performing vertical take-off and landing (VTOL) [3]. The quadrotor also provides two advantages from the traditional rotorcraft vehicle like the helicopter [3]. First, the quadrotor does not require complicated linkages to actuate its rotor; instead, the flight was controlled by varying the angular speed of each rotor. Second, by using four propellers instead of one, the quadrotor required smaller diameter for propellers.

The quadrotor is a basically unstable aerial vehicle [4]. Such instability—although quadrotor has been successfully flying in 1920—has inhibited the development of the practical quadrotor until recent years. The instability occurred because of the difficulties of controlling all of the four rotors simultaneously with a proper amount of bandwidth [3]; until these days, due to the availability of sophisticated flight controller board, one can fly the unmanned quadrotor using the aeromodelling remote controller. These conventional flight controllers, including the commercially produced, were based on the PID control method and designed by mathematical based model.

The PID control method is a common method of control which processes the error signal into input signal by the aid of proportional factor (P), integrative action (I), and differential action (D). The states were compared with their respective references (the set points) to obtain the error signal. The error signal amplified with the proportional gain (K_P), interpreted as the present error term, to improve the speed of the system response to its error. In parallel, the error signal also integrated together with the integral gain (K_I) as the accumulation of the past error term and being

differentiated together with the derivative gain (K_D) as the prediction of future error term [5]. These gains were adjusted with the tuning methods based on the linear mathematical model that represents the dynamics of the system. The PID gains can be tuned easily for the systems with a low level of nonlinearity and perfectly uncoupled between the modes of its dynamics. On the contrary, the quadrotor flight dynamics is a highly nonlinear, strongly coupled, multivariable, and also underactuated system [4]. Hence, these contraries stimulated various research about quadrotor flight control methods besides PID.

Alongside the linear control method like PID, the nonlinear methods are also developed widely from the underwater robot like ROV [6, 7] and AUV [8, 9], to the aerial robot like quadrotor. Several applied methods for the quadrotor and unmanned robots are the backstepping control [10–13], sliding mode control [9, 14–17], and the learning-based control method like fuzzy logic and artificial neural-network-based controllers [18]. This article compared one of the artificial neural-network-based controllers, that is, the artificial neural network's direct inverse control (DIC-ANN), with the PID controller in the quadrotor attitude dynamics.

Artificial neural network (ANN) is a parallel distributed processor consisting of simple neurons that enabled memorizing the knowledge of a system by mimicking its nonlinear model after training process [19]. In the control areas, the ANN-based control system arises due to the drawbacks of the nonlinear control systems like backstepping, sliding mode controller, and so forth. The backstepping controller has a fast converge rate and can handle uncertainties, but the robustness is bad; while the sliding mode controller has good robustness and simple physical implementation, it suffers the chattering phenomena [20].

The ANN worked as regressor structure which minimizes an error function in an iterative process by applying the backpropagation of error algorithm. Since ANN is a non-parametric model consisting of connected neurons in several layers, the training process will adjust the weight in every neuron's connection to obtain the optimum fitting of system dynamics representation. For control purposes, the ANN has been used to invert the system dynamics to obtain the input signal correlated with the desired output results. The inversion technique initiates the direct inverse control (DIC) method with the use of ANN. The ANN method has been applied for nonlinear systems from chemical reactor [21] to nuclear purposes [22], whereas, in the field of unmanned system's controller, the DIC-ANN becomes research topics in UAV, for example, hexacopter [19], helicopter [23], and quadrotor [24] to the AUV [8].

This article proposed the method of comparing the DIC-ANN with the PID control method. The research of DIC-ANN in UAV controller [19, 23, 24] has published the advantages of the method for UAV application. However, to validate their improvement, the results still not be comparable with the conventional control method. Hence this article assesses the requirement of such comparison for the validation purpose. The method was derived to perform a "head to head" comparison to measure the improvement occurred.

To give insight into the proposed comparative method between ANN and PID control in quadrotor attitude, the paper will be organized as follows. After introductory narration in the first section, Section 2 will address the PID control system representing the dominant conventional control method for UAV. To introduce the ANN control system, Section 3 will present the DIC-ANN control system for quadrotor attitude dynamics. Since the ANN-based control appears to have advantages compared to the conventional control method, Section 4 will discuss the comparison method required for validation purposes. Continued from the previous sections, the comparative analysis will be performed in Section 5 to validate the advantages of DIC-ANN over PID control method. Finally, Section 6 will conclude the work in this article.

2. PID Control of Quadrotor Dynamics

The PID is a popular method of control for various system dynamics. Besides the wide application in the industry, the PID was also applied in UAV control systems. Including the quadrotor, PID also become a common method to build the flight controller. Although many methods and control algorithms have been explored for a quadrotor, like backstepping, sliding mode, feedback linearization, and adaptive control, the PID method was popularly applied in current control board developer [25]. The main reason of PID implementation is the simple structure, the simple hardware implementation, being easily tunable, and having a good range of performance [26]. Yet the PID controller was simple to be implemented, but the manual tuning process for its gains is a complex and time-consuming effort with required experience and caution [25]. Moreover, from the practitioner point of view, the mathematical based PID tuning is considered to be difficult and impractical for the quadrotor purpose, so they tuned their PID by *trial and error* method instead [27–30].

2.1. Quadrotor Attitude Dynamics. The quadrotor is a UAV built from four rotors installed at the corner of "plus" shaped frame (the + configuration). Each rotor consists of a fixed-pitch propeller mounted to the brushless DC motor for lifting the vehicle while controlling its attitude simultaneously. As an illustration, Figure 1 showed the quadrotor built by the Computational Intelligence and Intelligent Systems (CIIS) research group of Universitas Indonesia for their research purposes.

Quadrotor gains its recent popularity from recreational, commercial, and military purpose [31]. The commercial aspect varies from filming landscapes to movies, even delivering purchased items to buyers with the aid of GPS [32]. The advanced purposes also served by quadrotor are such as real-time information gathering for disaster area [1] and becoming part of rescuing technology in SAR activities [2]. Quadrotor is also popular as a research platform in flight control areas due to the simple design and maintenance; the ability to take-off, hover, and land vertically; being inexpensive and of lower risk compared to traditional helicopter [3, 32]. The quadrotor controlling can be illustrated as shown in Figure 2.



FIGURE 1: Quadrotor UAV.

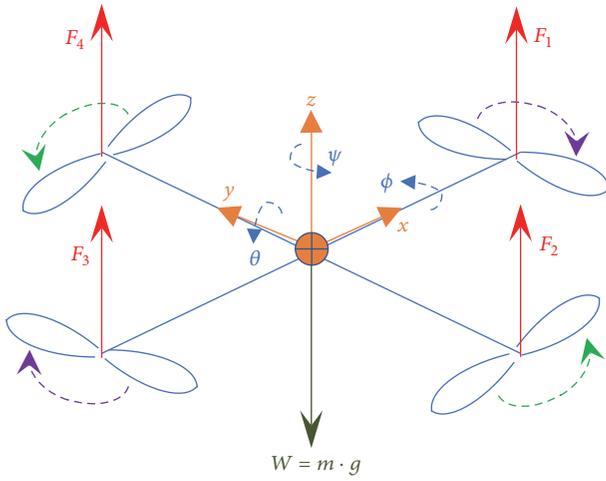


FIGURE 2: Quadrotor flight convention.

The typical quadrotor sketched in Figure 2 was having the known rotorcraft characteristics of underactuated and strong coupling in pitch-yaw-roll [33]. The latitude (z) of hovering, rolling (ϕ), pitching (θ), and yawing (ψ) are directly actuated by changing the rotor's RPM, while the forward and side translation can only be performed indirectly. This underactuated system of quadrotor dynamics will become a suitable case for control method comparison and evaluation. The addition of strong nonlinearity and coupled dynamics brought another challenge to be handled by the compared methods.

Conventionally, the quadrotor dynamic model was derived from the first principle technique. Starting with the Newton-Euler equation of motion, the rotational dynamics of UAV can be expressed by the following vector equation [34]:

$$\mathbf{J}\dot{\vec{\omega}} = \vec{u} - \vec{\omega} \times (\mathbf{J}\vec{\omega}). \quad (1)$$

Equation (1) is related to the quadrotor's angular acceleration vector, $\dot{\vec{\omega}}$, due to the control torque vector, \vec{u} , over the cross product of its current angular rate, $\vec{\omega}$, and its inertia tensor, \mathbf{J} , while the quadrotor's angular acceleration vector, $\dot{\vec{\omega}}$, is inversely proportional to the inertia tensor, \mathbf{J} . The inertia

tensor consists of three inertia terms with respect to x , y , and z rotation axis, that is, I_{xx} , I_{yy} , and I_{zz} , plus another six terms of product of inertia between their respective crossed axis, that is, I_{xy} , I_{yx} , I_{yz} , I_{zy} , I_{zx} , and I_{xz} , arranged as follows:

$$\mathbf{J} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}. \quad (2)$$

The quadrotor angular rate $\vec{\omega}$ consists of the rolling rate, p , pitching rate, q , and yawing rate, r , in the vector bracket $\vec{\omega} = [p \ q \ r]^T$. The quadrotor control torque vector, \vec{u} , consists of rolling torque, u_ϕ , pitching torque, u_θ , and yawing torque, u_ψ , as expressed in the vector bracket $\vec{u} = [u_\phi \ u_\theta \ u_\psi]^T$.

In practice, by expanding (1) together with (2), one can rearrange it to obtain another form of Newton-Euler formalism for the quadrotor attitude dynamics equation as follows [35, 36]:

$$\begin{aligned} \dot{p} &= \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{J_{TP}}{I_{xx}} + \frac{u_\phi}{I_{xx}} \\ \dot{q} &= \frac{I_{zz} - I_{xx}}{I_{yy}} pr - \frac{J_{TP}}{I_{yy}} + \frac{u_\theta}{I_{yy}} \\ \dot{r} &= \frac{I_{xx} - I_{yy}}{I_{zz}} p + \frac{u_\psi}{I_{zz}} \\ \ddot{z} &= -g + (\cos \theta \cos \phi) \frac{u_z}{m}. \end{aligned} \quad (3)$$

Equation (3) shows the quadrotor's angular acceleration in rolling (\dot{p}), pitching (\dot{q}), and yawing (\dot{r}) axis, as functions of moment of inertias in x -, y -, and z -axis (I_{xx} , I_{yy} , I_{zz}); rotor inertia (J_{TP}); control torque for rolling-pitching-yawing (u_ϕ , u_θ , u_ψ); and roll-pitch-yaw angular rates (p , q , r). The hovering dynamics expressed the quadrotor's vertical acceleration (\ddot{z}) as a function of mass (m), gravity constant (g), vertical control thrust (u_z), and pitching and rolling angle (θ and ϕ). In kinematics, the roll-pitch-yaw angular rates (p , q , r) were measured in body frame and rotating the quadrotor to achieve its orientation in the inertial frame. The orientation of the quadrotor can be observed as the roll-pitch-yaw flight angles (ϕ , θ , ψ) and their differential with respect to time related with p , q , r , as follows [37]:

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix}. \quad (4)$$

Equations (3) and (4) clearly show the nonlinearity of the quadrotor angular dynamics, for example, the multiplication between states p , q , r in (1) or between states ϕ , θ , ψ in (2). The strong coupling can also be seen in the θ and ϕ interaction when moving the (\ddot{z}); in other words, to induce a vertical climb the quadrotor also experienced the pitch-roll coupling

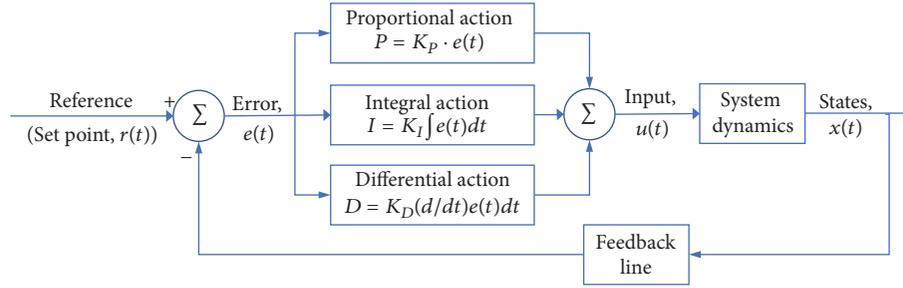


FIGURE 3: PID control scheme of universal plant system.

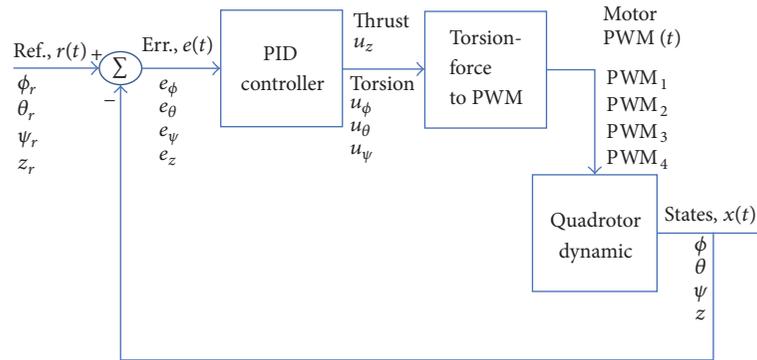


FIGURE 4: PID control scheme for quadrotor dynamic system.

and vice versa. Consequently, a proper model of altitude dynamics involved all roll, pitch, and yaw coupling each other. Thus, controlling the altitude dynamics is interrelated with the control of rolling, pitching, and yawing movement. These conditions enhanced the requirement to perform a modeling method which accommodates such strong nonlinearities. After the completion of Section 2, such requirement of nonlinear modeling is accommodated by applying the ANN-based system identification to model the quadrotor dynamics in Section 3.

2.2. The Main Ideas in PID Control Systems. The PID control method works in a feedback scheme to stabilize a system dynamics. The feedback line passes the output to the summation junction to obtain the errors and proportionally multiplies it (P-action), integrates it (I-action), and differentiates it (D-action) into the system's input to have it reduced. For optimal result, these P, I, and D actions are amplified with their respective gains, K_p , K_I , and K_D as illustrated in Figure 3.

The proportional action is speeding the system's response to counter the error according to the K_p gain. Thus, better response due to error measured should be obtained by high K_p gain. However, the increment of K_p gain above a certain limit can destabilize the system. The integral action accumulated the error and continually reduce the further error values. The optimum value of K_I eliminates the system's steady state error. The differential action processed the rate of change of the system's error. The sensitivity of this action was affected by K_D . The optimum value of K_D predicts the future error of the system, to anticipate such error.

To produce the best result, these gains must be tuned to attain the optimal combination. Such tuning was dependable on the accuracy of the dynamic modeling. For practical purposes, the developed PID technique is mostly tuned in the linearized model of the system. Thus, applying PID in quadrotor will potentially uncover its nonlinearity aspects.

2.3. Constructing PID Controller for Quadrotor Attitude Dynamics. Recently, most of the various flight controller boards like Ardupilot, Pixhawk, and OpenPilot applied the PID control method with double loop structure for roll, pitch, and yaw channel [25]. Thus, PID becomes the conventional method in UAV's autonomous feature. In the field of research, the quadrotor's PID control was processing the error values to generate the control torsion [38–40]. This is due to the application of Newton-Euler model to analyze the quadrotor flight behavior.

The controller scheme in Figure 4 compares the roll, pitch, yaw, and altitude with their references as their respective errors. Each error values were fed into the PID to produce the hovering thrust, the rolling, pitching, and yawing control torsion. Such configuration was common in quadrotor studies involving PID, according to the following structure [40–42]:

$$u_z = K_{P_z} (z_r - z) + K_{I_z} \int (z_r - z) + K_{D_z} \frac{d}{dt} (z_r - z)$$

$$u_\phi = K_{P_\phi} (\phi_r - \phi) + K_{I_\phi} \int (\phi_r - \phi) + K_{D_\phi} \frac{d}{dt} (\phi_r - \phi)$$

$$u_\theta = K_{P_\theta} (\theta_r - \theta) + K_{I_\theta} \int (\theta_r - \theta) + K_{D_\theta} \frac{d}{dt} (\theta_r - \theta)$$

TABLE 1: The PID gains.

Configuration	Mode	KP	KI	KD	N	Remark
Config. #1	Roll (ϕ)	0.4	0.3	0.1	5	Assuming all rotors have similar characteristics
	Pitch (θ)	0.4	0.3	0.1	5	
	Yaw (ψ)	0.2	0.1	0.1	5	
	Altitude (z)	0.3	0.3	0.1	5	
Config. #2	Roll (ϕ)	0.4	0.3	0.1	5	Treating each rotor with its own respective coefficients
	Pitch (θ)	0.4	0.3	0.1	5	
	Yaw (ψ)	0.2	0.1	0.1	5	
	Altitude (z)	0.3	0.3	0.1	5	
Config. #3	Roll (ϕ)	2	5	2	5	Treating each rotor with its own respective coefficients
	Pitch (θ)	2	5	2	5	
	Yaw (ψ)	1	5	2	5	
	Altitude (z)	2	2.5	0.5	5	

$$\begin{aligned}
u_\psi = & K_{P_\psi} (\psi_r - \psi) + K_{I_\psi} \int (\psi_r - \psi) \\
& + K_{D_\psi} \frac{d}{dt} (\psi_r - \psi).
\end{aligned} \tag{5}$$

The symbols ϕ , θ , and ψ refer to the flight angles of roll, pitch, and yaw. Thus, u_z denotes the hovering thrust, while u_ϕ , u_θ , and u_ψ denote the rolling, pitching, and yawing torsion.

To perform the comparative simulation, the quadrotor was modeled using the ANN-based system identification which directly related the altitude and flight angles as the function of the PWM values, that is, the motor command signals. The choice of such PID structure, although common in quadrotor flight analysis was unable to be directly cascaded to the twin ANN model for comparison purpose in this article. Thus, as illustrated in previous Figure 4, a special inverse mechanism must be constructed to convert this controlling force-torsion into PWM values or the motors command signals.

2.4. Tuning the Quadrotor PID Gains . There are many types of tuning methods, varying from the simple Ziegler-Nichols to the more advanced fuzzy logic based tuning to obtain a good combination of PID gains [43]. However, an extensive study by O'Dwyer revealed that there were many industrial PID controllers “out of tune” [44]. Thus, regardless of the numerous methods available, the PID tuning was not really a simple task. Similarly, in the practical world, the quadrotor practitioners found that the tuning of quadrotor’s PID was difficult [27]. Hence, they tuned their PID by using *trial and error* method [27–30].

From their *trial and error* approach, these quadrotor practitioners developed systematics steps that differ from one to another. Beginning from the rough method, the quadrotor flight dynamics were observed, and then the PID gains are adjusted until the oscillation disappears [29] and continued with the common procedure which tuned the pitch and roll parameters with the same values and then tuned the yaw [27]. Another separated technique is tuning one axis at a time, from roll, pitch, and then yaw axis, while the quadrotor

hovers. In that axis, they adjust each gain of K_P , K_I , and K_D . Due to the coupling dynamics of quadrotor, they return to readjust the gains of previous axis [28]. The different sequence is also used by tuning of K_P first, then K_D , and finally the K_I gains [30]. Initially, the gains were set to zero and then slowly increased until the quadrotor’s oscillation critically damped; after that, the tuning process repeated until quadrotor flies in a stable manner.

A practitioner’s method in tuning quadrotor’s PID gains using computational software was reported in [45]. Although the computational software seemed to result in optimized gains, the first flight crashed due to the incorrect sampling rate, and extensive tuning had to be performed again to fly the quadrotor at the following attempt. Furthermore, the increment of the controller performance conducted by tuning the PID gains manually. These conditions have enhanced the requirement of an accurate model of quadrotor’s dynamic to correctly tune its PID gains.

The manual tuning of PID still becomes a common preferable method in quadrotor control studies [40, 41, 46]. Thus, in this work, the PID gains were manually tuned to work on the discrete form of controller [47]:

$$K_P + K_I \cdot T_s \frac{1}{z-1} + K_D \frac{N}{1 + N \cdot T_s (1/(z-1))}. \tag{6}$$

After extensive iteration, the optimal gains are obtained in Table 1.

Figure 5 showed the result of several tuning processes with the gains described in Table 1 before obtaining the preferable response of PID control on quadrotor altitude. The reference altitude profile is plotted by the green dashed line, showing two ramps and two segments of constant altitude. This choice of the pattern is explained in detail in Section 5. Thus, the objective of the tuning is to obtain such altitude response which follows the reference profile with small error to ensure the quadrotor is able to hover as commanded.

The first gain combination in the upper part of Table 1 results in the “Config 1” response drawn in black dash-dotted line in Figure 5. The gain was the same as the “Config 2” whose responses are also displayed in the blue dotted line. The significant difference between them is the way of

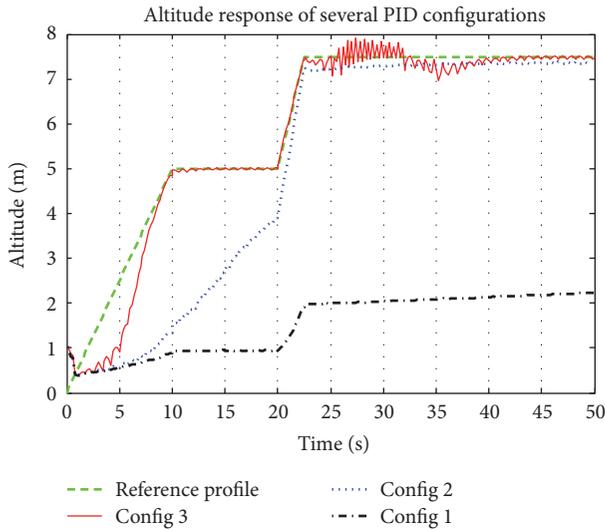


FIGURE 5: Altitude responses with respect to 3 configurations of PID tunings.

converting PID results into PWM values. In the “Config 1,” the PWMs are calculated using a single parameter of the motor, assuming that they were similar to each other. Unfortunately, each motor has their unique characteristic which cannot be neglected, and the ANN system identification captured this characteristic inside the quadrotor model. Hence, each motor then separately measured again and each of their parameters inputted the force-torsion converter to PWM. After recognizing the characteristics of each motor individually, the PID with the gain combinations in the middle part of Table 1 yield better response as plotted in “Config 2” curve. The further explanation of the PWM converter will be presented in Section 4.3.

The response of “Config 2” PID is still inadequate to bring the quadrotor into the desired altitude. Thus, the gains were increased again as in the combination in the lower part of Table 1 to speed up the response, while the limiters added and bound the controlling torsion and thrust inside the upper and lower threshold to avoid divergence. In the final trial, the “Config 3” gains obtained and enabled the quadrotor achieved the desired altitude shown with the red solid curve in Figure 5. However, a superposition of oscillation occurred as transient response. Nevertheless, since these gains are the best result of extensive trial and error process, then the PID with “Config 3” gains are chosen to be compared with the DIC-ANN in the following section. The performance of this PID controller is discussed in the comparative analysis at the Section 4.

3. Direct Inverse Control: Artificial Neural Network for Controlling Quadrotor Attitude Dynamics

This article will compare the DIC-ANN with the conventional control, which is represented by the PID method. The artificial neural network’s direct inverse control (DIC-ANN)

method itself had shown terrific achievement in controlling nonlinear dynamics aspect of UAV [19, 23, 24]. Considering the quadrotor as a highly nonlinear, strongly coupled, interacting in multivariable dynamics, and underactuated system [4], the DIC-ANN will be suitable for controlling such system. Thus, this section will present the quadrotor flight data and then discuss the DIC-ANN method together with the artificial neural-network-based modeling and control.

The direct inverse control (DIC) is a method to obtain the required input of a system to produce the desired output by constructing its dynamics inversion. In a linear system, such inversion can be performed analytically, for example, the dynamic inversion control method [48]. However, since quadrotor flight is a dynamic system with strong nonlinearity [4], the dynamic inversion will not be suitable to be implemented. In addition, the dynamic inversion cannot be performed alone for quadrotor because the system underactuated condition cannot guarantee that the vehicle will achieve the desired outputs [33]. Thus, the inversion of quadrotor’s dynamic needs to be constructed in such manner to accommodate its strong nonlinearity and underactuated condition. In this work, the DIC-ANN were implemented to control the quadrotor for a better flight performance.

3.1. The Direct Inverse Control of the ANN-Based Controller Design. The DIC-ANN applies the artificial neural network controller for a nonlinear process. Generally, the artificial neural network has been utilized for system identification and controller in wide areas of applications from the industrial nonlinear process to the vehicle control system. These included the thermal dynamic of pulsating heat pipe [49] and greenhouse temperature [50] to the autonomous vehicle control [51] and UAVs [19, 23, 24].

The DIC-ANN was constructed by directly cascading the system’s inverse, which consists of the artificial neural network controller, with the plant or the controlled system. The arrangement provides an identity that mapped the reference signal to the output of the controlled system. Consequently, the cascaded arrangement allows the artificial neural network to directly control the system’s dynamic and achieve the desired response. The cascaded arrangement illustrated in Figure 6 was developed to control the quadrotor in this article. The DIC produce required control command u , to bring the quadrotor states y , in achieving the reference states, r .

As an advantage, the DIC-ANN was able to use the most powerful feature of the artificial neural networks learning mechanism to synthesize its controller parameters. However, the initial output depended on the semirandomly initial weight matrix of the artificial neural network, which may reduce the controller robustness [52]. The block diagram illustrated in Figure 6 also showed that the direct inverse controller will depend solely on the reliability of the ANN controller in the inverse block.

In general, problems might be showed up and arisen severely because of no feedback signal involved, and furthermore, the absence of the feedback line will cause the lack of the ANN controller robustness. To optimize the open loop arrangement of the DIC, a feedback line was then equipped

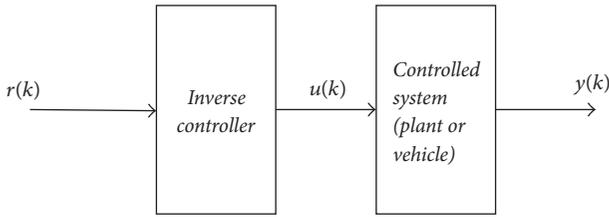


FIGURE 6: The DIC-ANN scheme.

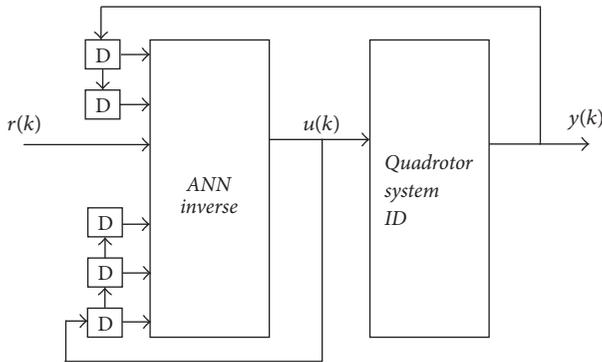


FIGURE 7: The concept of ANN inverse control cascaded with the quadrotor system identification.

into the input channel of the ANN controller so the closed loop system is achieved as illustrated in Figure 7. The required control command u , quadrotor states y , and the reference states r become the input for DIC-ANN with their delayed values symbolized by the D blocks.

The DIC applied the multilayer perceptron in its ANN controller which consists of an input layer, one hidden layer, and an output layer while the learning mechanism employed the backpropagation of error method. The learning process was conducted iteratively by adjusting the ANN weights that connect the neurons in the hidden layer with the output layer, according to the vector of the quadrotor input pattern and its desired output vector, that is, the corresponding set of the quadrotor output patterns. In parallel with the iteration, an input vector propagated forward in the network to temporarily calculate the output vector of the output layer. The difference of this output vector with its reference value will be defined as the error to be minimized by adjusting the connections weights. The backpropagation of error determines these adjustments by using the function of the neuron's mathematical model. As the learning process converges to minimize the mean square root error (MSE), the iteration stopped once the MSE fall below a chosen threshold.

3.2. The Quadrotor Telemetry Data. The artificial neural network (ANN) is a suitable method for modeling nonlinear systems, for example, quadrotor's flight dynamic. To obtain the model of the system's dynamics, the ANN runs a learning algorithm which iteratively relates the output of the system with its input until the acceptable error is achieved. Thus,

to obtain the quadrotor ANN model, the flight test must be conducted in gathering the flight data.

The UAV flight test is different from the common aircraft flight test. Due to the small dimension of the quadrotor UAV, its flight test becomes less in risk and less in the effort. The popularity of quadrotor had brought many practitioners which become a skilled pilot to fly the quadrotor for required maneuvers. The commercial controller device also equipped with data logging feature to record the gyro and accelerator output. Since conducting flight test has become feasible, the system identification for quadrotor can be performed to model its dynamics directly from the logged flight data.

The scope of this article is the altitude dynamics which involved the inner loop of the quadrotor flight control. Hence, the flight data to be processed consist of the quadrotor's altitude, its Euler flight angles, and its command signal for the motors. The Euler angles were roll, pitch, and yaw angle, and the commanded motor was PWM values for all the motor. The 4 minutes of the test result was presented in the green dash-dotted thick lines in Figure 9.

While the quadrotor was hovering, the test pilot excited the control stick to oscillate the quadrotor in rolling and pitching. In the green dash-dotted thick lines in roll and pitch graph of Figure 9, these variations logged between -20 and $+20$ degrees of flight angle. Before ending the flight test, the test pilot performs circular movement to vary the yaw angle from the direction of 100 to direction 120 (both in degrees). The altitude response showed the take-off, hover, and landing in the flight test. These flight data, especially the variations recorded, identified the quadrotor dynamics system by training the ANN model. After obtaining the sophisticated model, the ANN-DIC and PID controller can be compared head to head using this twin quadrotor dynamic model.

3.3. Modeling Quadrotor Using Artificial Neural Network System Identification. The quadrotor dynamics can be modeled by many ways, such as the physical based modeling which derived from the first principle or using the system identification methods which relied on the flight data. In a conventional way, the model from the first principle depended on the accuracy of its physical properties such as dimension, weight, and moments of inertia. However, moments of inertia can only be estimated by geometrical approximation [37, 53], CAD [54, 55], or pendulum method [56]. While the estimation relied on the measurement precision, dimension, and the construction of the pendulum [57], it also suffers from unmodeled phenomena when oscillating the UAV [58]. Thus, a system identification performed to build a reliable model for quadrotor dynamics which accommodated the nonlinearity, coupled and underactuated dynamics, that is, the ANN system identification.

Artificial neural network can be envisaged as a nonlinear transformation between multidimensional input and output [59]. In this work, the implemented ANN related the output of a system as the function of its input to be processed inter-relatedly into a set of activation functions via an optimum combination of weights. The weights then were iteratively tuned by the backpropagation of error algorithm using the

difference between the models' output from the system real output to adjust the weights.

A nonlinear dynamic system, which processes input x into output y , can be approximated with the following model:

$$y(k) = f(\Phi(k), \Theta). \quad (7)$$

$y(k)$ denotes the model's output, $\Phi(k)$ denotes the regression vector, and Θ denotes the parameter vector. Choose the Nonlinear AutoRegressive networks with eXogeneous inputs (NARX) structure using the finite amount of past inputs and past outputs as expressed by the regression function as follows:

$$\Phi(k) = (x(k-1), \dots, x(k-N_x), y(k-1), \dots, y(k-N_y)), \quad (8)$$

where N_x is the maximum lag of input and N_y is the maximum lag of output.

As the learning mechanism, the back propagation of error will approximate the nonlinear mapping for the NARX identification model as illustrated in Figure 8. The artificial neural network to identify the quadrotor dynamics will be defined by the input neuron including the input and output signal with $N_x = 2$ and $N_y = 2$, followed by a hidden layer consisting of 20 neurons and 4 output neurons. Each of the input elements ($x(k-1), \dots, x(k-N_x)$ and $y(k-1), \dots, y(k-N_y)$) is multiplied with the weight v_{xnm} or v_{ynm} and summed into each neuron in the hidden layer. The index $n = 1, 2, \dots, N$ denotes the number of input neurons and $m = 1, 2, \dots, M$ denotes the number of the hidden neurons.

These summed values were then inputted into the sigmoid activation function to obtain the output of the hidden neuron $z_m(k)$. The next step is multiplying the hidden neuron $z_m(k)$ with the weights w_m , then summing them all into the sigmoid function, and yielding the output neuron $y(k)$ according to the following expression:

$$y(k) = \sum_{m=1}^M w_m \left(\frac{1 - \exp \left\{ - \left(\sum_1^{N_y} y(k-n_y) v_{ynm} + \sum_1^{N_x} x(k-n_x) v_{xnm} \right) \right\}}{1 + \exp \left\{ - \left(\sum_1^{N_y} y(k-n_y) v_{ynm} + \sum_1^{N_x} x(k-n_x) v_{xnm} \right) \right\}} \right). \quad (9)$$

The system identification from the ANN has modeled the quadrotor dynamics in the fine result as shown in Figure 9. The measured values were plotted in the green dash-dotted line, while the modeling results were displayed in the solid red line. By using the PWM logged as input, the results showed that the ANN system identification has mimicked the quadrotor dynamics. Thus, the ANN model obtained became the Quadrotor Twin ANN ID to be controlled for both PID and DIC-ANN for the comparative simulation.

3.4. Performance of Quadrotor DIC-ANN. The DIC-ANN is composed of artificial neural network structures which represented the inverse of the system dynamics after finishing a training process. Mathematically, the DIC-ANN can be expressed in the following regression function:

$$x(k) = f(x(k-1), \dots, x(k-N_x), y(k+1), \dots, y(k-N_y)). \quad (10)$$

The schematic diagram in Figure 10 illustrates the iteration to obtain the DIC-ANN with $N_x = 2$ and $N_y = 2$ as the maximum lag of the input and output. The ANN architecture consists of a three-layer network with 28 neurons at the input layer, 20 neurons in the hidden layer, and 4 neurons in the output layer. The 28-20-4 network then trained to follow the response of the quadrotor by using the backpropagation of error algorithm.

After training the inverse block, the simulation of DIC-ANN showing satisfactory results in directing the quadrotor to achieve its desired reference. By cascading the DIC-ANN block with the system identification block and using a flight profile as the references for altitude, roll, pitch, and yaw,

the DIC showed coinciding curve between simulation results with the desired values in Figure 11. The altitude, roll, pitch, and yaw responses in the red dotted line following the variation of the reference are given in the green dashed line.

The quadrotor UAV flight responses in Figure 11 have validated the DIC-ANN performance. The ability to execute curvy and oscillatory profile as their references indicated that the DIC-ANN has accommodated the strong nonlinearity and the cross-coupling of the quadrotor dynamics. The constructed inverse block has mapped the targeted quadrotor output into a proper PWM command [19]. To assess the advantages of DIC-ANN over the conventional control method, the following sections will design the comparative method and conclude the results.

4. Deriving Comparative Methods of DIC-ANN and PID

The DIC-ANN has been reported as a promising method for the UAV flight dynamics control [19]. Beginning from the application in pattern recognition, the ANN developed into intelligent control systems and furthermore to autonomous technology in the intelligent transportation system [60]. This autonomous concept aimed to be the backbone of transportation technology to overcome the human error problem and increase the safety level of vehicle operation [60]. In current transportation applications, the autonomous technology was also implemented in UAV application for traffic monitoring [61].

This condition initiates a curiosity to analyze the increment of the performance of intelligent control method from its predecessor, the conventional control method. Moreover,

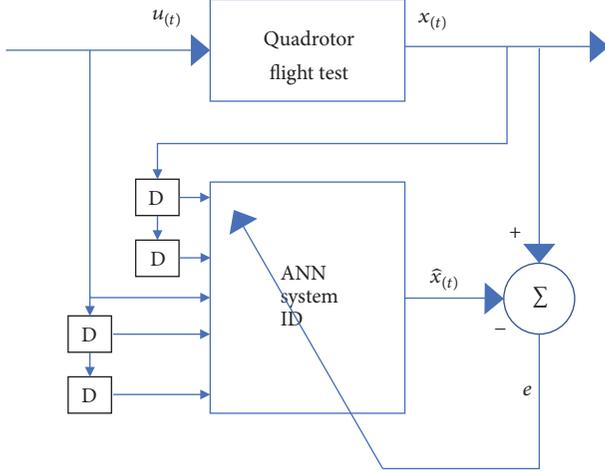


FIGURE 8: System Identification using ANN for the quadrotor dynamics.

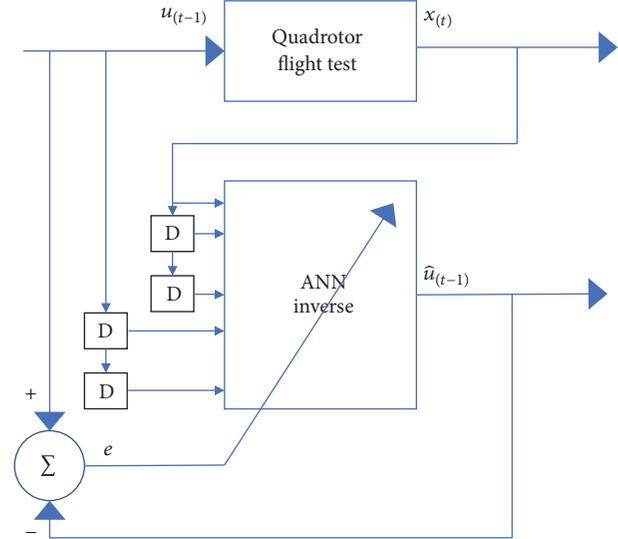


FIGURE 10: The inverse training scheme.

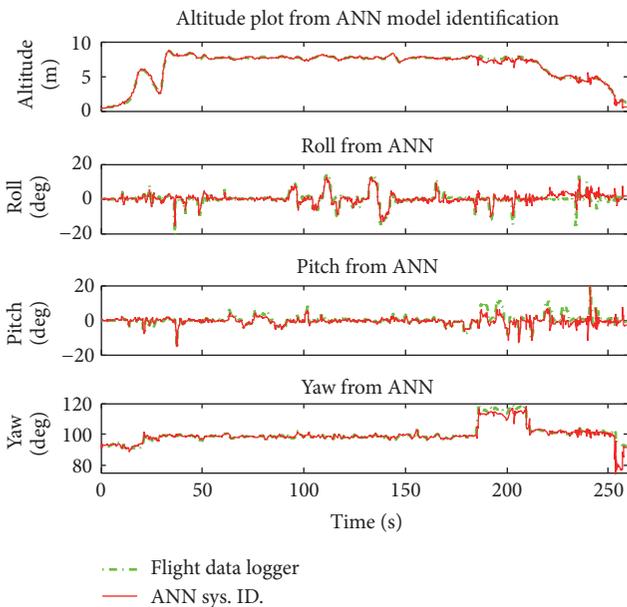


FIGURE 9: Quadrotor ANN system identification plot.

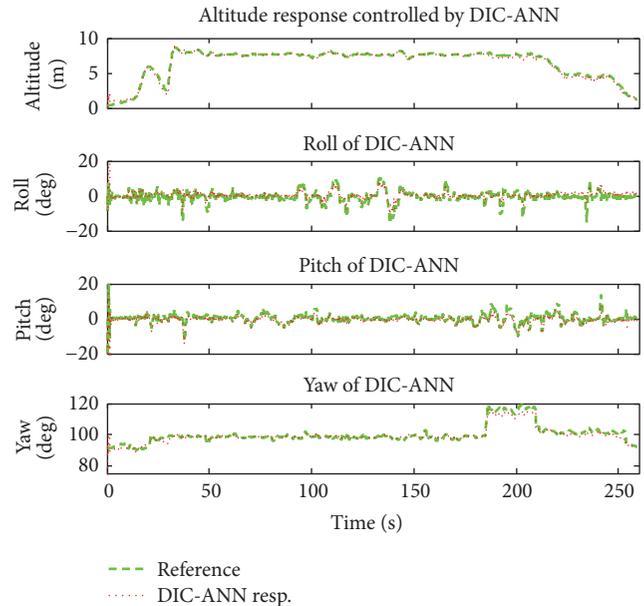


FIGURE 11: The DIC-ANN response of altitude, roll, pitch, and yaw.

the investigation of quantifying the advantage of intelligent control method over conventional method becomes a significant research. This article initiates addressing those issues by proposing a comparative methodology and, to be specific, comparatively simulating the DIC-ANN with PID control in quadrotor UAV.

4.1. The Proposed Method of Comparison. In control engineering research, there are many comparative studies between ANN-based control and conventional method that were reported, from the simple comparison of ANN control with classical control (including the comparison between ANN modeling with state space modeling) [62] to the comparison with PD control [63]. Since the PID control is widely applied, there are also several comparative studies between ANN-based control and PID at industrial purposes,

that is, in kiln [64], water bath temperature [65], compressor [66], gas production [67], and air conditioning [68].

These studies show satisfactory performance of the ANN-based method in controlling the nonlinear plants. Since quadrotor UAVs were governed in a nonlinear fashion, the implementation of ANN-based control should also be performed better than the PID. To prove such hypothesis, a proper comparative study was required in analyzing the advantages and performance increment of DIC-ANN controller from the PID.

There are many comparative studies performed between ANN-based and PID controller and hence, various methodologies are also applied. Thus, in this article, a methodology of comparison will be developed to obtain a proper conclusion.

The method to perform the comparison was designed to achieve the “apple to apple,” the “head to head,” and the “ceteris paribus” condition.

A good comparison must handle the “apple to apple” condition; that is, both control methods should be applied to the exact same quadrotor so their response will make sense to be comparable to each other. Thus, both controllers must have similar functionality in controlling the quadrotor dynamics. They must process the same type of input and produce the same type of output.

The second achievement is the “head to head” condition; that is, both control methods responses must be compared directly with each other. Consequently, both controllers will process exactly the same reference to be executed by the quadrotor. Their response will be plotted in the same graph to be directly compared and analyzed.

The last achievement is the “ceteris paribus” condition; that is, each comparison in a certain flight parameter will be held by exciting that parameter and keeping the other unexcited. The technique is applied to isolate the responses of a certain flight parameter from the rest of them.

To achieve the “apple to apple,” the “head to head,” and the “ceteris paribus” condition, the following methods are generated:

- (1) Using twin model (the ANN system identification of quadrotor dynamics) for both controllers
- (2) Converting the control force and torsion in PID output into PWM values
- (3) Simultaneously performing numerical simulation of both control methods
- (4) Using equivalent reference profile for both DIC-ANN and PID controller and then plotting their responses in a joint graph.

These steps guided the numerical simulation in producing the comparable results for analysis purposes.

4.2. Using the Quadrotor ANN System Identification as the Twin Model. The twin model was applied to obtain the “apple to apple” condition in the comparative simulation. The practice of using twin model in the joint simulation will ensure that both controllers experienced the exact same environment while interacting with one quadrotor dynamics. This type of equality will be unachievable if the comparison was held in the flight test. The controlled aerial environment will almost be impossible to be attained in the open space. Although the weather can be predicted with certain accuracy, the instantaneous wind condition is unpredicted in its direction and speed. For small dimension aircraft like UAVs, any wind occurring is not negligible for the vehicle’s dynamics. Such unexpected disturbances made the comparison be better performed in a simulation than in flight test.

Most of the studies of the PID controller applied the linearized state space model. On the other hand, most of the studies of DIC-ANN applied the ANN system identification for modeling the quadrotor dynamics. However, the use

of state space model will eliminate the nonlinear aspect which is the challenge to overcome. Thus, the ANN system identification becomes a suitable model of quadrotor dynamics for both control methods to be compared. The ANN model captured the strong nonlinearity, highly coupled, and underactuated system of quadrotor dynamics. In previous studies, besides the quadrotor, the comparison of ANN control and PID was using the ANN mutual model due to the strong nonlinearity in their plant [64, 68].

In the previous section, the quadrotor dynamics for the comparative simulation has been modeled with the ANN system identification scheme. The previous studies which modeled the UAV by the ANN system identification had proved the model’s high accuracy due to the very small value of mean of squared error [19, 23, 24]. The quadrotor ANN model in this comparative simulation receives PWM values as input and produces the roll, pitch, yaw, and altitude as the output. Thus, the ANN model becomes a more realistic representation of the quadrotor controller since the model directly processed command values to its final states which accommodate all aspects of nonlinearities and coupled dynamics to the internal mechanical interactions.

4.3. Converting Force-Torsion into PWM Signal. The quadrotor in this work was flown by four propellers and each of them was driven by BLDC motor. These motors receive command in the form of PWM signals from the main controller. These PWM signals were recorded in the flight data log and then together with the roll, pitch, yaw, and altitude were used to build the common ANN dynamic model of the quadrotor.

Consequently, a more realistic simulation can be achieved by using the PWM quantities as the controller’s output to be fed into the quadrotor dynamics model. Such arrangement was applied in directly converting the PID’s output to be PWM quantities without calculating the control forces and torsion [46]. However, such arrangement cannot be implemented here, since the output of PID was the control force and torsion. Thus, an inverse mechanism was required to convert these control force and torsion to be individual motor thrust and then invert the thrust to the motor’s PWM quantities.

In this article, the inverter mechanism was named as the “Torsion-Force-PWM converter” to denotes its input, output, and function. Similar PWM generator block was known with different terms since they have a different construction from each other, such as *rotational speed inverter* [11], *inverted movement matrix* [69], *PPM converter* [70], or *RPM-PWM mapping function* [42]. The steps of generating PWM values were summarized as follows:

- (1) Convert the control force-torsion to individual rotor thrust.
- (2) Convert the rotor thrust into its PWM values.

The first step is converting the control force-torsion to individual rotor thrust. The controlling torsion can be derived

and showed from each motor thrust by using Figure 2 in these relations:

$$\begin{aligned} u_z &= F_1 + F_2 + F_3 + F_4 \\ u_\phi &= \ell(-F_2 + F_4) \\ u_\theta &= \ell(-F_1 + F_3) \\ u_\psi &= c(F_1 - F_2 + F_3 - F_4). \end{aligned} \quad (11)$$

The notations F_1, F_2, F_3, F_4 are the thrust of motor numbers 1 to 4, respectively, ℓ are the distance between propeller's center hub and the quadrotor center point, and c is the constant of drag moment. Hence, each motor thrust can be computed as follows:

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -\ell & 0 & \ell \\ -\ell & 0 & \ell & 0 \\ c & c & c & c \end{bmatrix}^{-1} \begin{Bmatrix} u_z \\ u_\phi \\ u_\theta \\ u_\psi \end{Bmatrix}. \quad (12)$$

The second step is converting the rotor thrust into the motor's PWM values. This conversion was obtained by the aid of polynomial regression to invert the PWM command values as a function of its thrust. The regression curve in Figure 12 was obtained after measuring the rotor's thrust in a bench test. The similar description about the thrust measurement can be viewed in [71, 72].

To obtain the c constant, the coefficient for rotor's drag moment as a function of its thrust, the measurement result was mapped in a linear curve. By using the tachometer, the rotor thrust can be fitted with a linear curve when plotted with the squared values rotor's speed, Ω^2 , in Figure 13 and continued with fitting the drag moment with the squared values rotor's speed, Ω^2 , in Figure 14. Thus, by combining their gradient, the c constant can be obtained.

Thus, the Torsion-Force-PWM converter generates the PWM in the linear approximation. After finishing the Torsion-Force-PWM converter, the PID gains can be tuned in the conventional technique. However, the plots of measurement results were showing the existence of nonlinear effect in the small amount. While the "Torsion-Force-PWM converter" excludes the existence of this nonlinearity, the DIC-ANN embedded them.

4.4. Simultaneous Numerical Simulation. Parallel with the aim of using the twin model, the simultaneous simulation also performed to ensure an "apple to apple" comparison. Thus, the simulation should run in the discrete time domain with the equivalent reference profile, each into the twin model. In simultaneous simulation it was meant that each control method runs on its own model, which is an equal model of quadrotor dynamics. The simultaneous simulation treats both control methods with an equal condition; that is, both are isolated from disturbance or environmental effect. Figure 15 showed the illustration of this simultaneous simulation.

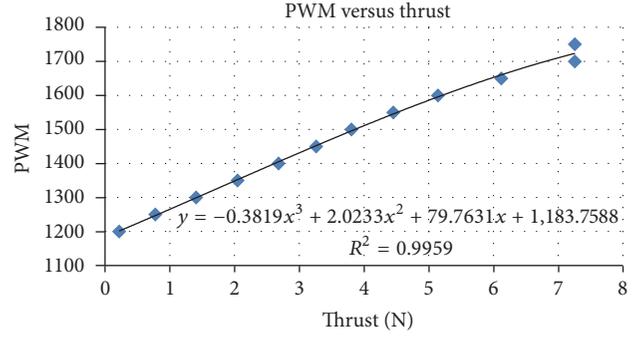


FIGURE 12: Regression of PWM from thrust.

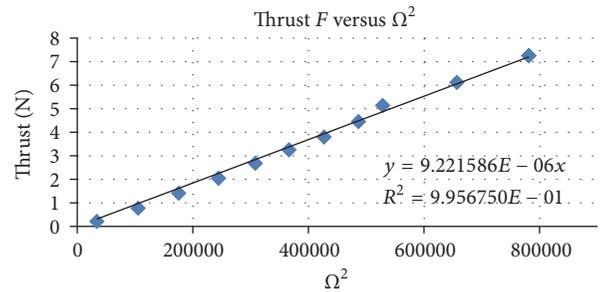


FIGURE 13: Regression of thrust with respect to squared speed.

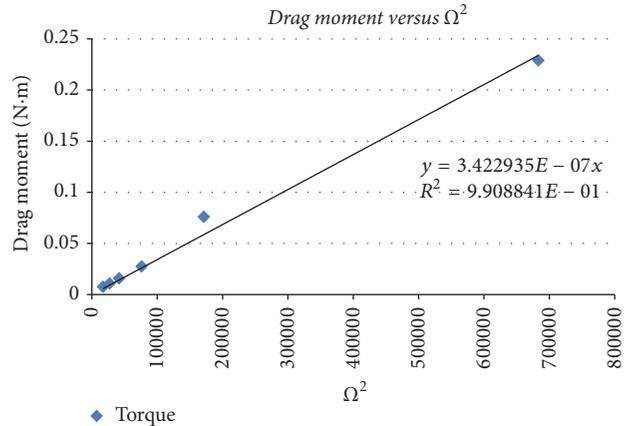


FIGURE 14: Regression of drag moment with respect to squared speed.

Using the equivalent reference profile, into the twin model, for the simultaneously numerical simulation there is a series of method designs to ensure the "apple to apple" comparison. In this work, the input was applied to the altitude channel in the ramp and constant height profile. This combination of reference profiles was designed to excite the responses of DIC-ANN and PID controller at the transient and steady condition.

Although there are many studies that compared the PID and ANN-based control systems, most of them are not directly comparing the results. Each controller's performance was plotted in the separate graph and separately analyzed [64, 65]. However, a direct comparison will produce a sharp analysis and unambiguous results. Thus, by placing the

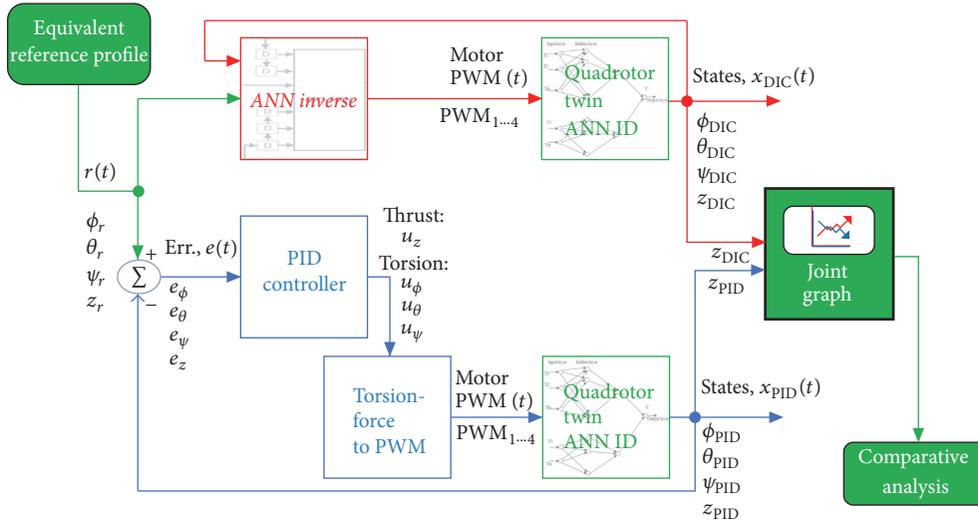


FIGURE 15: The DIC-ANN and PID comparative simulation scheme.

results of DIC-ANN and PID controller in the, respectively, mutual plot, they will be directly compared and identify their advantages or disadvantages to each other.

5. Comparison Result of DIC-ANN and PID Control Method

The comparison of DIC-ANN and PID control method is performed in the following description: first, each controller is using the twin model, second, for PID, the controlling force and torsion converted to obtain PWM values, third, the numerical simulation was executed simultaneously, and fourth, both controllers were excited with equivalent altitude profile as their reference and the responses plotted in joint graphs. Since the early descriptions have been revealed, this section compared the results in “head to head” fashion. Several published studies analyzed their comparison in the joint graph [62, 67, 68], as adopted in the following analysis.

5.1. Flight Profile for the Altitude Response Comparison. The flight simulation aimed to mimic the flight test of a quadrotor as illustrated in Figure 16. The simulated flight consists of four phases, first, the “take-off and climb” phase at $0 < t < 10$ s, second, the “hovering” phase at $10 < t < 20$ s, third the “climb in ramp” phase at $20 < t < 22.5$ s, and fourth, the “final altitude” phase at $22.5 < t < 50$ s. Initially, the quadrotor is simulated to take-off from the ground and climb to a safety altitude preferable for a test pilot to execute required maneuvers. In the numerical quantities, the quadrotor is simulated to fly from zero to 5 m of height in the “take-off and climb” phase and then “hovering” to maintain the 5 m of height. Such flight profile had enabled both controllers to achieve a steady hovering condition; thus, an equivalent excitation can be induced to stimulate the time response especially in the altitude to be observed. The excitation was enforced by lifting the quadrotor in the “climb in ramp” phase to achieve the 7.5 m height of “final altitude” phase. The slope in the “climb in ramp” phase is 1 m/s to give a 2.5 m of altitude

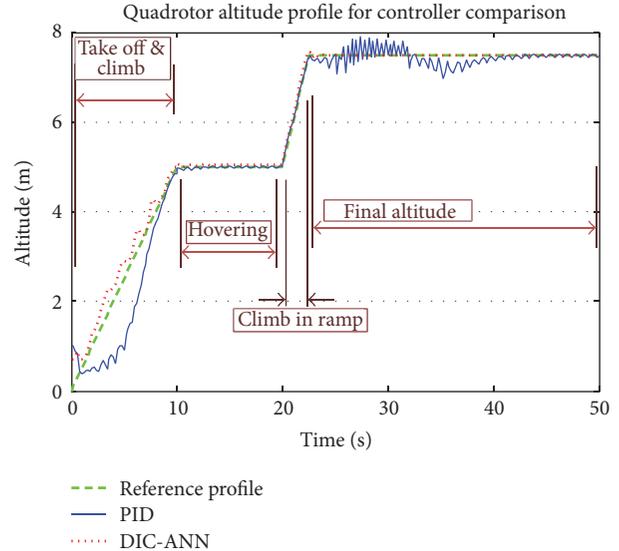


FIGURE 16: Flight profile for altitude’s response comparison.

increment from $t = 20$ s to $t = 22.5$ s. Finally, the simulation stopped at $t = 50$ s after the quadrotor altitude achieved the steady state of the “final altitude” phase.

Although, in the first 10 seconds of simulation, the DIC-ANN showed a remarkable response in following the take-off profile, no further comparison will be studied in this segment since the maneuver was meant to bring the quadrotor to the safety altitude. The comparisons are started after $t = 20$ s when the hovering quadrotor was excited by the 1m/s of ramp. The comparisons also covered the transient response at $t = 22.5$ s and the steady state response right after the ramp ended at the 7.5 m of constant height.

5.2. Comparison of the Ramp Response and Steady State. The ramp maneuver was chosen to obtain a more realistic simulation of flight compared to the standard step function

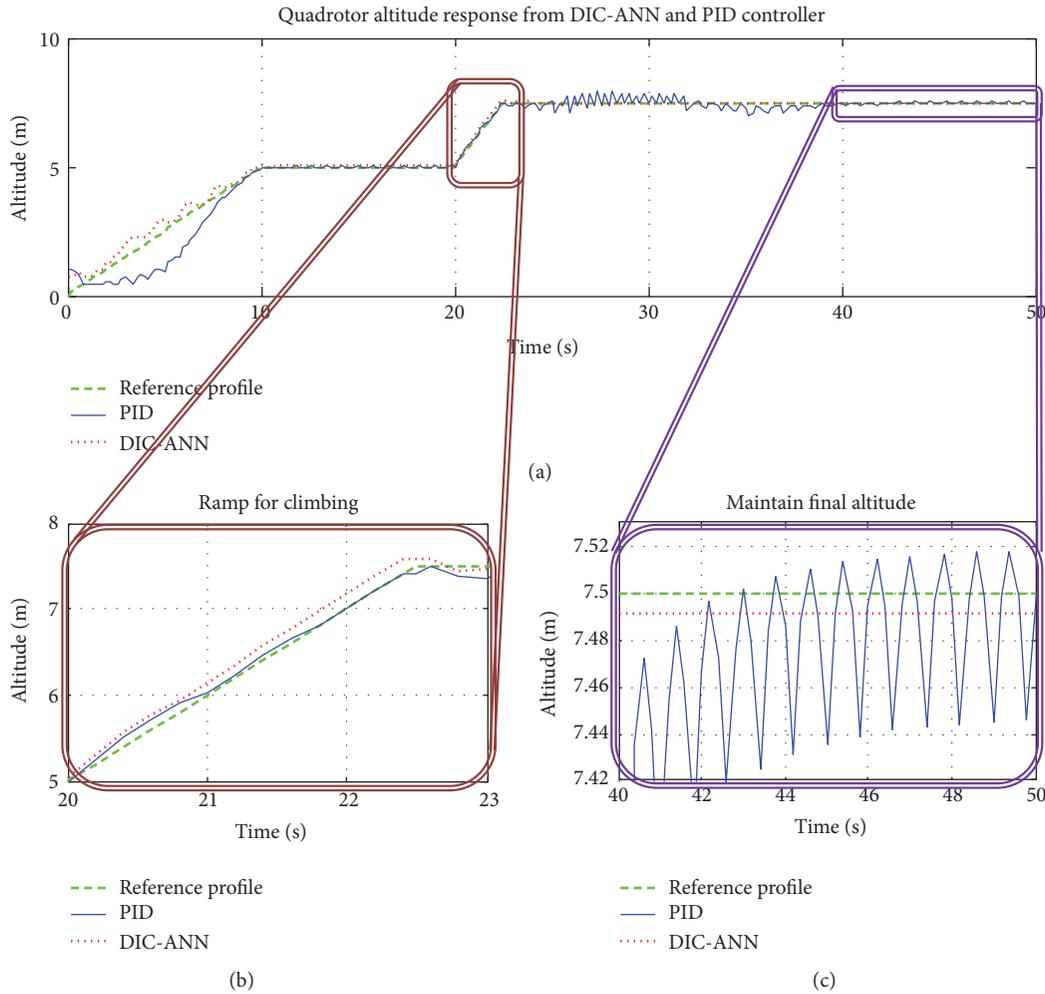


FIGURE 17: Comparison of response in ramp and steady state.

for exciting the modes of quadrotor motion. The ramp response of quadrotor altitude is shown in Figure 17(b) entitled “ramp for climbing.” The steady state response was shown in Figure 17(c) entitled “maintain final altitude.” The reference altitude profile is displayed as the green dashed line, and the DIC-ANN response was represented by the red dotted curve while the PID response was presented by the solid blue curve.

Along with the reference profile, the DIC-ANN altitude response was quickly damped and rising to follow the reference slope without significant oscillation observed. While the oscillation in PID controller response was observed for 2.5 s until the ramp ended. A closer examination in the Figure 17(b) showed that the PID controller produced a lower error when following the ramp maneuver compared to the DIC-ANN altitude response.

After the ramp completed at $t = 22.5$ s and becomes constant altitude in the reference profile, the altitude response shows the transient oscillatory mode until it damped and achieved the steady state condition. The steady state achievement can be seen in Figure 17(c), while the transient response is discussed in the following subsection. Although the

simulation runs 37.5 s after the transient occurrence at the edge of the ramp, the PID controller is still unable to damp the oscillatory mode. This residual oscillatory mode forms the PID resulting in a vertical vibration with 0.04 m or 4 cm of amplitude and 1 Hz of frequency, that is, significant enough with the quadrotor dimension and its potential mission performance. On the contrary, the DIC has a shorter transient response and achieves its steady state for 3 s after the ramp is completed, shown in Figure 18(b) at the following subsection. Thus, the DIC-ANN has settling time equal to 3 s while the PID controller fails to settle after 37.5 s due to the residual oscillatory mode with 0.04 m or 4 cm of amplitude and 1 Hertz of frequency.

5.3. Comparison of the Transient Responses. The quadrotor was brought up from 5 m of altitude to 7.5 m by a 1 m/s ramp maneuver. Hence, the ramp is completed after 2.5 s and the altitude is being constantly maintained. However, a sudden discontinuity at the end of ramp maneuver excites the oscillatory mode of the quadrotor flight motion. These oscillatory motions, known as the transient response, occurred at both of DIC-ANN and PID controllers. The transient

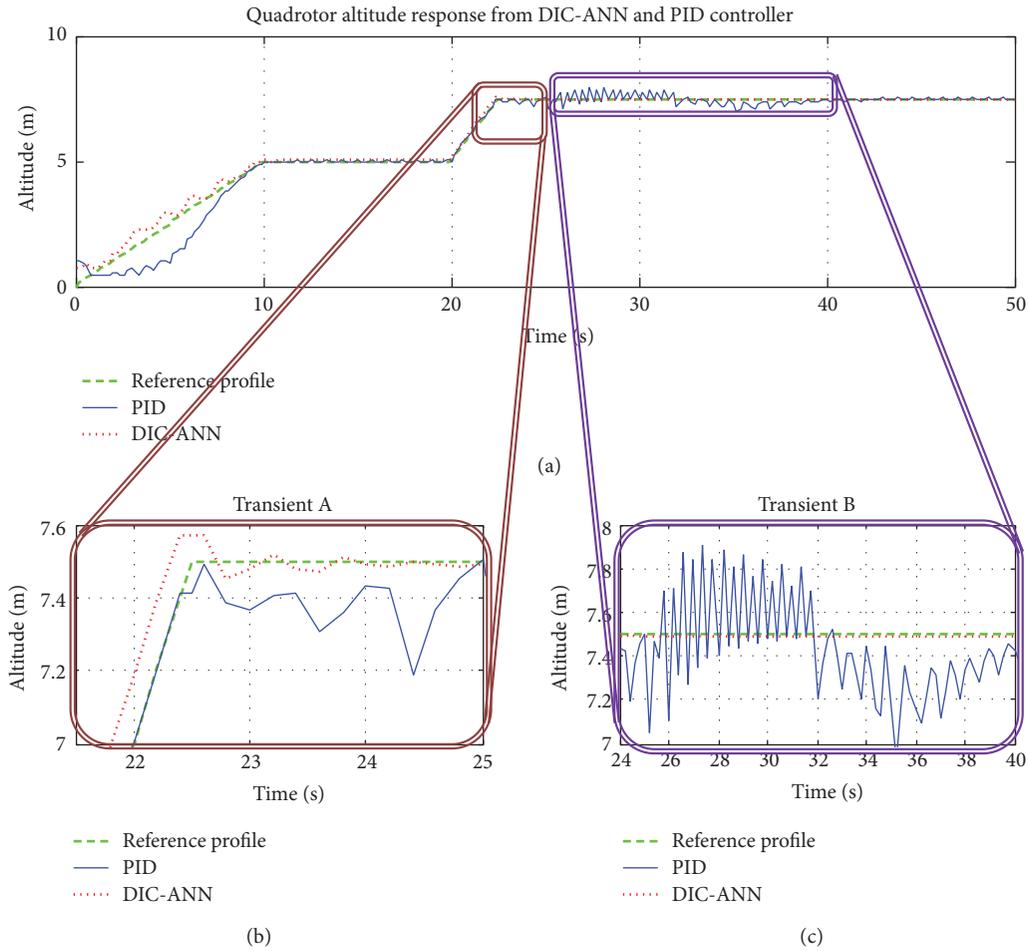


FIGURE 18: Comparison of the transient response.

responses of the quadrotor altitude are displayed in Figures 18(b) and 18(c). Although the DIC-ANN damped the transient response immediately, the PID controller still exhibits its transient response involving multiple frequencies. These plural frequencies were exposed by splitting the display into two sequential graphs, that is, Figure 18(b) entitled “Transient A” for $21.5\text{ s} < t < 25\text{ s}$ and Figure 18(b) entitled “Transient B” for $24\text{ s} < t < 40\text{ s}$. The reference altitude profile is depicted as the green dashed line, and the DIC-ANN response is represented by the red dotted curve while the PID response is represented by the blue solid curve.

As interpreted from Figure 18(b), the DIC-ANN successfully damped the transient response and brought the quadrotor altitude to its desired height. On the other hand, the PID experienced 1 Hz transient oscillation which was undamped after 3 s. As the observation continued, the PID controller in Figure 18(c) showed the superposition of the short period (large frequencies) oscillation modulated with the long period (rare frequencies), for example, the swing with 16 s period. To be concluded, the DIC-ANN have better performance in handling the transient response of quadrotor altitude compared to the PID controller, since DIC-ANN damped the oscillation after 3 s while PID was unable to do so for 17.5 s after excitation occurred.

The absolute error of both controllers presented in Figures 19(a), 19(b), and 19(c), while Figure 19(d) showed the altitude profile to relate itself with the corresponding error characteristic. The DIC-ANN absolute error is represented by the red dash-dotted curve while the PID absolute error is represented by the diamond-marked blue curve. The absolute error of ramp execution displayed in Figure 19(b) provides the maximum error of DIC-ANN which does not exceed 0.2 m. After $t = 22\text{ s}$, the DIC-ANN absolute error tends to be reduced as shown in the slight decrement which ended at $t = 22.5\text{ s}$ when the ramp is completed. Since most of the quadrotor mission required altitude precision in the hovering maneuver, thus, the DIC-ANN ramp errors are still acceptable although the PID have smaller error, i.e., a half of the DIC-ANN. On the contrary, Figure 19(c) highlighted the PID absolute error to 0.5 m from the desired altitude in the final altitude aimed. After the quadrotor completed the ramp, the DIC-ANN ramp error decreased immediately in less than 3 s as shown in Figure 19(c) while the PID ramp error increased and oscillated between 0 and 0.5 m until 17.5 s after the ramp completed. The absolute error characteristic enhanced the DIC-ANN capability to overcome a sudden change in the reference profile and following it with lower oscillation than the PID.

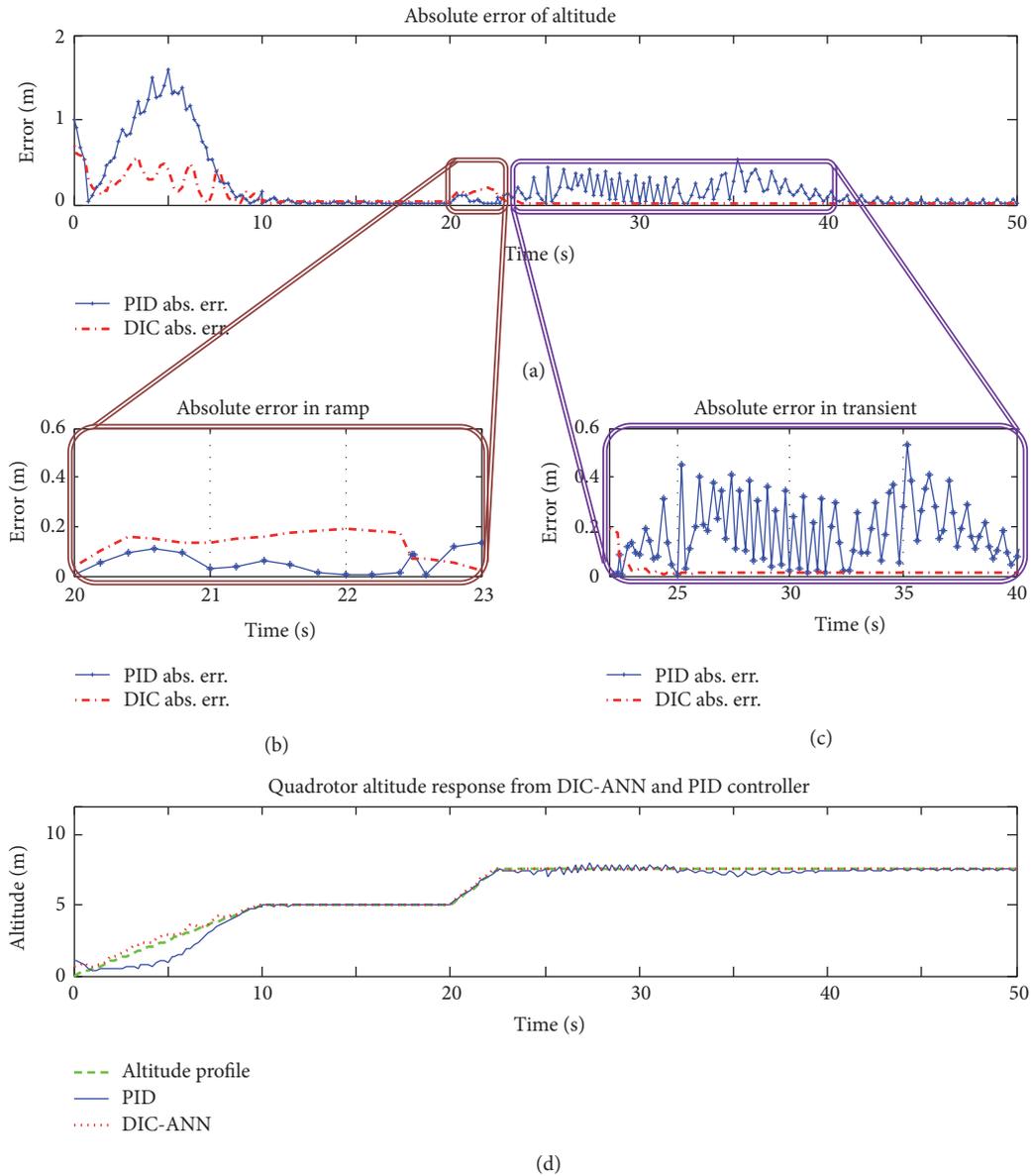


FIGURE 19: Absolute error of DIC-ANN and PID in altitude response.

6. Concluding Remarks

The method of comparison built in this article originated from the satisfactory results of DIC-ANN controller in UAVs. Thus, the gap in the comparative research between DIC-ANN and PID for quadrotor has been elaborated to pull out a proper comparative method between these control systems. The comparison composed the twin ANN model usage, the conversion into PWM values, the simultaneous numerical simulation, and the equivalent reference profile to be plotted on a joint graph. These steps ensured the comparison to be *apple to apple, head to head, and ceteris paribus*.

The comparison result showed that the DIC-ANN performance was better than the PID controller in handling the quadrotor altitude dynamics. Although the DIC experienced

slight overshoot in following the ramp input given, the error is not exceeding 0.2 m while it tends to be decreased.

The sudden changes from climbing to hover exhibit the lower steady state error in the DIC-ANN controller and the transient oscillation damped faster by the DIC-ANN. These results showed that DIC-ANN was better in controlling the nonlinear vehicle than the PID controller did. Equipped with the learning algorithm, the DIC-ANN performed better compared to the PID that involved efforts of tuning. Such results will support the DIC-ANN application for the autonomous technology. For further development, the comparison will be suitable to be expanded to analyze the translation responses. By analyzing the UAV responses in the x and y direction, the trajectory can be compared and validated.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] T. Prucksakorn, K. Wachiraratnanakornkul, and I. Nilkhamhang, "Unmanned aerial vehicle for observing landslide with iterative feedback tuning," in *Proceedings of the 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2013*, May 2013.
- [2] A. Sargent, Ed., *Drone tested by Albany sea rescuers at Salmon Holes set to help in emergencies*, ABC News, 2017.
- [3] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, South Carolina, 2007.
- [4] X. Zhang, X. Li, K. Wang, and Y. Lu, "A survey of modelling and identification of quadrotor robot," *Abstract and Applied Analysis*, vol. 2014, Article ID 320526, 2014.
- [5] J. Mange, S. Pace, and A. Dunn, "Optimization of PID Controller Parameters for Automated Ground Vehicle Control on Dynamic Terrain," in *International Conference on Scientific Computing (CSC)*, p. 3, Las Vegas, Nev, USA, 2015.
- [6] C. S. Chin, M. W. S. Lau, and E. Low, "Supervisory cascaded controller design: experiment test on a remotely operated vehicle," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 225, no. 3, pp. 584–603, 2011.
- [7] C. S. Chin, M. W. S. Lau, E. Low, and G. G. L. Seet, "Robust and decoupled cascaded control system of underwater robotic vehicle for stabilization and pipeline tracking," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 222, no. 4, pp. 261–278, 2008.
- [8] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive Neural Network Control of AUVs With Control Input Nonlinearities Using Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, 2017.
- [9] R. Cui, L. Chen, C. Yang, and M. Chen, "Extended state observer-based integral sliding mode control for an underwater robot with unknown disturbances and uncertain nonlinearities," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6785–6795, 2017.
- [10] C. Tian, J. Wang, Z. Yin, and G. Yu, "Integral backstepping based nonlinear control for quadrotor," in *Proceedings of the 35th Chinese Control Conference, CCC 2016*, pp. 10581–10585, July 2016.
- [11] Q. Wei, H. Wang, Q. Wu, and M. Chen, "Backstepping-based attitude control for a quadrotor UAV using nonlinear disturbance observer," in *Proceedings of the 34th Chinese Control Conference, CCC 2015*, pp. 771–776, July 2015.
- [12] D. Matouk, O. Gherouat, F. Abdessemed, and A. Hassam, "Quadrotor position and attitude control via backstepping approach," in *Proceedings of the 8th International Conference on Modelling, Identification and Control, ICMIC 2016*, pp. 73–79, November 2016.
- [13] Y. Yu, Y. Guo, X. Pan, and C. Sun, "Robust backstepping tracking control of uncertain MIMO nonlinear systems with application to quadrotor UAVs," in *Proceedings of the 2015 IEEE International Conference on Information and Automation, ICIA 2015 - In conjunction with 2015 IEEE International Conference on Automation and Logistics*, pp. 2868–2873, August 2015.
- [14] M. Reinoso, L. I. Minchala, J. P. Ortiz, D. Astudillo, and D. Verdugo, "Trajectory tracking of a quadrotor using sliding mode control," *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2157–2166, 2016.
- [15] M. K. Shaik and J. F. Whidborne, "Robust sliding mode control of a quadrotor," in *Proceedings of the 11th UKACC United Kingdom Automatic Control Council International Conference on Control, UKACC Control 2016*, September 2016.
- [16] J. Anwar and F. M. Malik, "Performance of non-linear observers for closed loop terminal sliding mode control of quadrotor UAV," in *Proceedings of the 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 244–252, Islamabad, Pakistan, January 2017.
- [17] P. Dey, S. R. Kurode, and R. Ramachandran, "Robust attitude control of quadrotor using sliding mode," in *Proceedings of the 1st International Conference on Automatic Control and Dynamic Optimization Techniques, ICACDOT 2016*, pp. 268–272, September 2016.
- [18] L. Li, L. Sun, and J. Jin, "Survey of advances in control algorithms of quadrotor unmanned aerial vehicle," in *Proceedings of the 16th IEEE International Conference on Communication Technology, ICCT 2015*, pp. 107–111, October 2015.
- [19] B. Kusumoputro, H. Suprijono, M. A. Heryanto, and B. Y. Suprpto, "Development of an attitude control system of a heavy-lift hexacopter using Elman recurrent neural networks," in *Proceedings of the 22nd International Conference on Automation and Computing, ICAC 2016*, pp. 27–31, September 2016.
- [20] Y. Li and S. Song, "A survey of control algorithms for quadrotor unmanned helicopter," in *Proceedings of the IEEE 5th International Conference on Advanced Computational Intelligence (ICACI '12)*, pp. 365–369, October 2012.
- [21] E. R. Rene, M. E. López, J. H. Kim, and H. S. Park, "Back propagation neural network model for predicting the performance of immobilized cell biofilters handling gas-phase hydrogen sulphide and ammonia," *BioMed Research International*, vol. 2013, Article ID 463401, 2013.
- [22] D. Sutarya and B. Kusumoputro, "Identification of industrial furnace temperature for sintering process in nuclear fuel fabrication using NARX neural networks," *Science and Technology of Nuclear Installations*, vol. 2014, Article ID 854569, 8 pages, 2014.
- [23] H. Suprijono and B. Kusumoputro, "Direct inverse control based on neural network for unmanned small helicopter attitude and altitude control," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, no. 2-2, pp. 99–102, 2017.
- [24] M. A. Heryanto, H. Suprijono, B. Y. Suprpto, and B. Kusumoputro, "Attitude and Altitude control of a quadcopter using neural network based direct inverse control scheme," *Advanced Science Letters*, vol. 23, no. 5, pp. 4060–4064, 2017.
- [25] A. Julkananusart and I. Nilkhamhang, "Quadrotor tuning for attitude control based on double-loop PID controller using fictitious reference iterative tuning (FRIT)," in *Proceedings of the 41st Annual Conference of the IEEE Industrial Electronics Society, IECON 2015*, pp. 4865–4870, November 2015.
- [26] C. K. Tan and J. Wang, "A novel PID controller gain tuning method for a quadrotor landing on a ship deck using the invariant ellipsoid technique," in *Proceedings of the 2014 14th*

- International Conference on Control, Automation and Systems, ICCAS 2014*, pp. 1339–1344, October 2014.
- [27] M. marc81, “Quadcopter PID output, Message to Robotics Stack Exchange Online Discussion Forum,” 2014, <http://robotics.stackexchange.com/questions/2964/quadcopter-pid-output/2965#2965>.
- [28] O. Liang, “Quadrotor PID Explained and Tuning, Share Knowledge and Ideas, March 19,” <http://oscarliang.com/quadcopter-pid-explained-tuning/>.
- [29] E. H. T. Teisberg, “Scratch-Built Quadcopter Project, Thomas Teisberg, March 19,” <http://www.thomasteisberg.com/quadcopter/>.
- [30] hauptmech. and J. Baker, “What are good strategies for tuning PID loops?, Message to Robotics Stack Exchange Online Discussion Forum, 2012,” <http://robotics.stackexchange.com/questions/167/what-are-good-strategies-for-tuning-pid-loops>.
- [31] R. Allen and M. Pavone, “A Real-Time Framework for Kinodynamic Planning with Application to Quadrotor Obstacle Avoidance,” in *AIAA Guidance, Navigation, and Control Conference*, Gaylord Palms, Kissimmee, Florida, USA, 2016.
- [32] K. N. Dang, G. Lee, and T. Kang, “Linear quadrotor modelling and attitude controller design based on experimental data,” in *Proceedings of the 15th International Conference on Control, Automation and Systems, ICCAS 2015*, pp. 472–476, October 2015.
- [33] A. Das, K. Subbarao, and F. Lewis, “Dynamic inversion with zero-dynamics stabilisation for quadrotor control,” *IET Control Theory & Applications*, vol. 3, no. 3, pp. 303–314, 2009.
- [34] A. Honglei, L. Jie, W. Jian, W. Jianwen, and M. Hongxu, “Backstepping-based inverse optimal attitude control of quadrotor,” *International Journal of Advanced Robotic Systems*, vol. 10, article no. 223, 2013.
- [35] S. Bouabdallah and R. Siegwart, “Full control of a quadrotor,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 153–158, San Diego, Calif, USA, November 2007.
- [36] T. Bresciani, *Modelling, identification and control of a quadrotor helicopter [M.S. theses]*, 2008.
- [37] R. Beard, “Quadrotor dynamics and control, All Faculty Publications, 2008,” <http://scholarsarchive.byu.edu/facpub/1325>.
- [38] H. Zheng, Q. Zeng, W. Chen, H. Zhu, and C. Chen, “Improved PID control algorithm for quadrotor based on MCS,” in *Proceedings of the 7th IEEE Chinese Guidance, Navigation and Control Conference, CGNCC 2016*, pp. 1780–1783, August 2016.
- [39] J. O. Pedro, M. Dangor, and P. J. Kala, “Differential evolution-based PID control of a quadrotor system for hovering application,” in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation, CEC 2016*, pp. 2791–2798, July 2016.
- [40] K. Issam and G. Qingbo, “Research on control strategies for the stabilization of quadrotor UAV,” in *Proceedings of the 5th International Conference on Intelligent Control and Information Processing, ICICIP 2014*, pp. 286–292, August 2014.
- [41] M. Elsamanty, A. Khalifa, M. Fanni, A. Ramadan, and A. Abo-Ismael, “Methodology for identifying quadrotor parameters, attitude estimation and control,” in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '13)*, pp. 1343–1348, Wollongong, Australia, July 2013.
- [42] C. Wang, M. Nahon, and M. Trentini, “Controller development and validation for a small quadrotor with compensation for model variation,” in *Proceedings of the 2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014*, pp. 902–909, May 2014.
- [43] M. Reyad, M. Arafa, and E. A. Sallam, “An optimal PID controller for a quadrotor system based on de algorithm,” in *Proceedings of the 11th International Conference on Computer Engineering and Systems, ICCES 2016*, pp. 444–451, December 2016.
- [44] A. O’Dwyer, “An Overview of Tuning Rules for the PI and PID Continuous-Time Control of Time-Delayed Single-Input, Single-Output (SISO) Processes,” in *PID Control in the Third Millennium, Advances in Industrial Control*, pp. 3–44, Springer, London, UK, 2012.
- [45] A. Mukarram, U. A. Fiaz, and U. I. Khan, *Altitude Control of a Quadcopter*, Department of Electrical Engineering, Pakistan Institute of Engineering and Applied Sciences, Islamabad, Pakistan, 2015.
- [46] G. Bo, L. Xin, Z. Hui, and W. Ling, “Quadrotor helicopter Attitude Control using cascade PID,” in *Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016*, pp. 5158–5163, May 2016.
- [47] Mathworks., “PID Controller, Discrete PID Controller,” <https://www.mathworks.com/help/simulink/sref/pidcontroller.html>.
- [48] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, John Wiley and Sons, Inc, 3rd edition, 2015.
- [49] A. Jokar, A. A. Godarzi, M. Saber, and M. B. Shafii, “Simulation and optimization of a pulsating heat pipe using artificial neural network and genetic algorithm,” *Heat and Mass Transfer*, vol. 52, no. 11, pp. 2437–2445, 2016.
- [50] A. Manonmani, T. Thyagarajan, M. Elango, and S. Sutha, “Modelling and control of greenhouse system using neural networks,” *Transactions of the Institute of Measurement and Control*, 2016.
- [51] P. Sinha, L. Hadjiiski, and K. Mutib, “Neural Networks in Autonomous Vehicle Control,” *IFAC Proceedings Volumes*, vol. 26, no. 1, pp. 335–340, 1993.
- [52] B. Kusumoputro, K. Priandana, and W. Wahab, “System identification and control of pressure process rig® system using back-propagation neural networks,” *ARNP Journal of Engineering and Applied Sciences*, vol. 10, no. 16, pp. 7190–7195, 2015.
- [53] R. Syam, “Dynamics and fuzzy logic method for controlling quadcopter,” *Research Journal of Applied Sciences*, vol. 11, no. 6, pp. 251–260, 2016.
- [54] W. Hussein, M. El-khatib, A. Elruby, and H. Haleem, “Quad Rotor Design, Simulation And Implementation,” in *International Conference on Computer Science from Algorithms to Applications, (CSAA09)*, JW Marriott, Mirage City, Cairo, Egypt, 2009.
- [55] H. Oh, D.-Y. Won, S.-S. Huh, D. H. Shim, M.-J. Tahk, and A. Tsourdos, “Indoor UAV Control Using Multi-Camera Visual Feedback,” *Journal of Intelligent Robotic Systems*, vol. 61, no. 1, pp. 57–84, 2011.
- [56] J. Yang, Z. Cai, Q. Lin, and Y. Wang, “Self-tuning PID control design for quadrotor UAV based on adaptive pole placement control,” in *Proceedings of the 2013 Chinese Automation Congress, CAC 2013*, pp. 233–237, November 2013.
- [57] A. Teimourian and D. Firouzbakht, “A Practical Method for Determination of the Moments of Inertia of Unmanned Aerial Vehicles,” in *XXII Conference of Italian Association of Aeronautics and Astronautics*, Napoli, Italy, 2013.

- [58] M. R. Jardin and E. R. Mueller, "Optimized measurements of UAV mass moment of inertia with a bifilar pendulum," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, pp. 4790–4812, Monterey, Calif, USA, August 2007.
- [59] A. E. Ruano, P. M. Ferreira, and C. M. Fonseca, "An overview of nonlinear identification and control with neural networks," in *Intelligent Control Systems using Computational Intelligence Techniques, IET Control Engineering Series*, A. E. Ruano, Ed., p. 37, The Institution of Engineering and Technology, London, UK, 2008.
- [60] C.-H. Chen, "Intelligent transportation control system design using wavelet neural network and PID-type learning algorithms," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6926–6939, 2011.
- [61] Y. Du, C. Zhao, F. Li, and X. Yang, "An Open Data Platform for Traffic Parameters Measurement via Multirotor Unmanned Aerial Vehicles Video," *Journal of Advanced Transportation*, vol. 2017, Article ID 8324301, 12 pages, 2017.
- [62] P. Ryckaert, B. Salaets, A. S. White, and M. Stoker, "A comparison of classical, state-space and neural network control to avoid slip," *Mechatronics*, vol. 15, no. 10, pp. 1273–1288, 2005.
- [63] M. Azizur Rahman and M. Ashraful Hoque, "On-line self-tuning ANN-based speed control of a PM DC motor," *IEEE/ASME Transactions on Mechatronics*, vol. 2, no. 3, pp. 169–178, 1997.
- [64] F. M. Dias and A. M. Mota, "A comparison between a PID and internal model control using neural networks," in *5th World Multi-Conference on Systemics, Cybernetics and Informatics*, vol. 268, p. 273, Orlando, EUA, 2001.
- [65] Y. V. Parkale, "Comparison of ANN Controller and PID Controller for Industrial Water Bath Temperature Control System using MATLAB Environment," *International Journal of Computer Applications*, vol. 53, no. 2, pp. 1–6, 2012.
- [66] M. I. Mohammed, Z. B. Noussier, R. M. Abdel Maksoud, and F. Z. Amer, "Surge avoidance using speed and valve methodologies controlled by PID, fuzzy and neural networks approaches," in *The 2011 International Conference on Computer Engineering and Systems*, pp. 73–78, IEEE, Cairo, Egypt, December 2011.
- [67] M. Y. Azwar, M. A. Hussain, A. K. Abdul Wahab, and M. F. Zanil, "A Comparative study between Neural Networks (NN)-based and Adaptive-PID Controllers for the Optimal Bio-Hydrogen Gas Production in Microbial Electrolysis Cell Reactor," *Computer Aided Chemical Engineering*, vol. 37, pp. 1529–1534, 2015.
- [68] P. Kumar and A. P. K. Singh, "Comparative Analysis of Air Conditioning System Using PID and Neural Network Controller," *International Journal of Scientific and Research Publications*, vol. 795, 2013.
- [69] D. Bokan, N. Cetic, J. Kovacevic, and V. Ilic, "Control of unmanned aerial vehicle based on XMOS platform," in *Proceedings of the 2013 21st Telecommunications Forum Telfor, TELFOR 2013*, pp. 893–896, November 2013.
- [70] K. Li, S. K. Phang, B. M. Chen, and T. H. Lee, "Platform design and mathematical modeling of an ultralight quadrotor micro aerial vehicle," in *Proceedings of the 2013 International Conference on Unmanned Aircraft Systems, ICUAS 2013*, May 2013.
- [71] P. S. King, *System Design Methodology and Implementation of Micro Aerial Vehicles*, Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore, 2014.
- [72] M. Dupuis, J. Gibbons, M. Hobson-Dupont et al., *Design Optimization of a Quad-Rotor Capable of Autonomous Flight*, Aerospace Engineering, Worcester Polytechnic Institute, Worcester, Mass, USA, 2008.

Research Article

State-of-Charge Estimation and Active Cell Pack Balancing Design of Lithium Battery Power System for Smart Electric Vehicle

Z. C. Gao,¹ C. S. Chin,² W. D. Toh,¹ J. Chiew,¹ and J. Jia¹

¹Clean Energy Research Centre, School of Engineering, Temasek Polytechnic, 21 Tampines Avenue 1, Singapore 529757

²Faculty of Science, Agriculture, and Engineering, Newcastle University in Singapore, Singapore

Correspondence should be addressed to C. S. Chin; cheng.chin@ncl.ac.uk

Received 7 July 2017; Revised 26 October 2017; Accepted 13 November 2017; Published 31 December 2017

Academic Editor: Emanuele Crisostomi

Copyright © 2017 Z. C. Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an integrated state-of-charge (SOC) estimation model and active cell balancing of a 12-cell lithium iron phosphate (LiFePO₄) battery power system. The strong tracking cubature extended Kalman filter (STCEKF) gave an accurate SOC prediction compared to other Kalman-based filter algorithms. The proposed groupwise balancing of the multiple SOC exhibited a higher balancing speed and lower balancing loss than other cell balancing designs. The experimental results demonstrated the robustness and performance of the battery when subjected to current load profile of an electric vehicle under varying ambient temperature.

1. Introduction

Lithium-ion battery storage system plays a vital role in electric vehicle (EV) applications [1–5]. Portable lithium batteries are commonly used for their high energy density and low cost. However, the voltages of these battery cells are quite low and require many battery cells in series to meet the voltage requirement for real applications. In addition, there exist many problems with the battery management system (BMS) such as inaccurate state-of-charge (SOC) estimation due to multiple charging and discharge of the cells. Hence, the SOC [6] was one of the essential parameters to estimate in order to prevent damage to the battery. Unfortunately, the estimation of SOC is not a simple process as it depends on factors such as battery's capacitance, resistance, internal temperature, ambient temperature [7, 8], and other cell characteristics. The ratio of the charge delivered to the battery over the total charge of the battery was typically used to obtain the SOC.

The common ampere-hour integral method caused biases due to the integration. On the other hand, the voltage method used the battery voltage and SOC relationship or discharge curve to determine the SOC. The battery cell needs to disconnect from the load in order to determine the

open-circuit voltage during actual battery operation. Another approach based on equivalent circuit model (ECM) [9–12] was used to estimate the SOC. The resistors and capacitors were used for modeling the battery cell in ECM. Although it requires an experiment to validate the parameters, it is relatively easy to use and implement. However, the battery is a nonlinear time-varying system with capacity changes due to aging, and ambient temperature variation, an accurate estimation of the SOC on the ECM, is therefore required.

Another approach using fuzzy logic [13] was proposed to estimate the SOC of the lithium-ion battery in EV. It utilized both battery terminal voltages to overcome the problem of overdischarging. The rules of combining membership functions were not robust due to the types of the battery cell used. The active cell balancing for multicell battery was not discussed.

As a result, a nonmodel based approach such as neural network [14–16] and its combination with fuzzy logic named fuzzy-neural network [17] was used. The support vector machine [18] was also used to estimate the battery dynamics. Although the neural network based approach and SVM were both a nonlinear estimation method that does not require the battery model [19, 20], a significant dataset and

computational time for training the SOC value were required. To circumvent the issue of training time, the extreme machine learning (ELM) [21–25] was proposed that uses regularized least squares to compute faster than the conventional quadratic programming approach without tuning the hidden to output neurons. Nevertheless, a significant dataset for an accurate SOC estimation was required despite the short training time. Moreover, the application of a multicell SOC estimation and cell balancing was not proposed.

Instead of nonmodel based approach, a more accurate physical modeling using electrochemical model was used [26–29]. The electrochemical model provided an accurate physical meaning of the electrical and chemical properties of the battery cell. The nonlinear partial differential equations increased the model complexity and computational time during SOC estimation. As a result, a linear model using fewer parameters in the electrochemical model was adopted [30] with Kalman filter (KF) to estimate the SOC. The extended Kalman filter (KF) [31–33], sliding-mode observer [34, 35], and Luenberger observer [36, 37] were applied to estimate the SOC using the ECM model. However, it required an accurate model of the battery and higher computing resource with correct initialization of parameters that changed rapidly.

The extended KF (EKF) was then proposed to estimate the SOC using a nonlinear ordinary differential equation model [38]. The unscented KF was utilized [39] to avoid such linearization of the nonlinear equation in EKF. A nonlinear SOC estimator [40] was then employed on the electrochemical model of the battery instead of the ECM. But the computation time and numerical error increased due to the numerical approach used in solving the partial differential equations. In addition, it could not track the sudden change in the SOC value. A strong tracking cubature Kalman filter (CKF) [41] that outperformed the EKF was proposed to track the sudden change of SOC value accurately. It was followed by the multirate strong tracking extended Kalman filter (STEKF) [42] to handle a higher dimensional state estimation of SOC. However, the paper demonstrated on a six-cell battery stack without using the CKF to track the fast-changing SOC value. Hence, the use of the strong tracking cubature on extended Kalman filter (STCEKF) with cell balancing is desired. Furthermore, most of the literature that involved SOC estimation was mainly dealt with theoretical development and comparisons with other SOC estimation methods on a single cell.

In addition to SOC estimation, an active cell balancing (AB) circuit [43–45] is another important element in the battery management system. Most batteries are made up of multiple cells that required AB to prevent each cell from overcharge or discharge after the SOC was determined. Although the topics of SOC estimation and active balancing were separately analyzed or published, it is essential to examine how a battery power system with both SOC estimation and AB can be realized.

With cell balancing, the battery cell lifetime and capacity can be further extended. In general, there exist two different AB circuits. The first type consumed the redundant energy of parallel resistance to maintain the terminal voltage of the cells. The second type used inductors, converters, or

transformers to realize energy transfer between cells. The energy in the cells with higher SOC or terminal voltage can be transferred to other cells to maintain the same SOC and voltage among the cells. The disadvantage was the complexity of controlling the converters [41, 46–48]. A multistage equalization was used to simplify the circuit and increase the balancing speed.

There exist different cell balancing methods using a resistor, capacitor, inductor/transformer, and energy converter balancing method [44, 45]. For example, the shuttling capacitors cell balancing method named “charge shuttling cells equalization” used capacitors as external energy storage for alternating the energy between the cells for charge balancing. The control strategy was straightforward and efficient. But the disadvantages were the relatively long balancing time and high cost as compared to the passive balancing method. However, the shuttling capacitors can be optimized using a double-tiered switched capacitor (STSC) based on the switched capacitor to decrease the balancing time.

Another energy conversion cell balancing topology using inductors was used to move energy from a cell to another cell. The relative high balancing current provided a shorter balancing time. However, the inductor topology was relatively costly with high balancing losses. The energy converters such as Cuk, Buck-Boost, full-bridge PWM energy, and Quasi-Resonant Converter were used for cell balancing. When the imbalance was detected, the converters allowed the energy to transfer between cells. This method was used for high power applications. But they can be expensive and quite complex to control.

Hence, a proposed multicell battery power system with both SOC estimation and AB has the following contributions. The novelty will come from the proposed AB circuit to accomplish the groupwise balancing of the multiple SOC obtained via the STCEKF. The AB circuit consists of a switching circuit using a DC-DC converter, IGBT drivers, inductors, and diodes to balance the SOC among the cells. Each cell uses a single fast driver to reduce the balancing loss by controlling the cells’ discharge, charge, and cut-off. The battery packs will be divided into groups to balance the energy simultaneously by transferring the energy from the cell with higher to lower SOC. In summary, a new integrated STCEKF model-based SOC estimation and the active cell balancing will be implemented on a multicell LiFePO₄ battery power system to improve its performance and robustness in an electric vehicle.

The paper is organized as followed. In Section 2, a proposed battery design is presented followed by Section 3 of SOC estimation. In Section 4, describes the experimental results of the battery pack using actual load followed by the conclusion in Section 5.

2. Proposed Battery Power System Design

The nominal voltage and continuous discharge current of a single LiFePO₄ cell (ANR26650M1-B) are limited to 3.3 V and 50 A, respectively. As shown in Figure 1, the battery pack consists of 12 cells connected in series to produce 2 kWh.

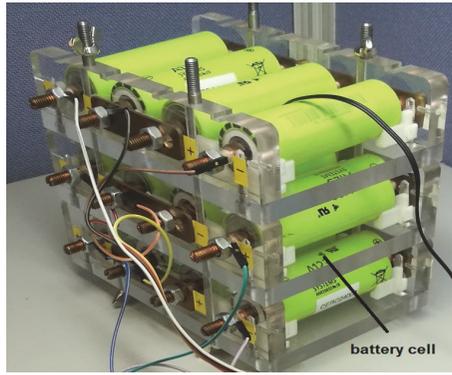


FIGURE 1: Proposed lithium iron phosphate battery prototype.

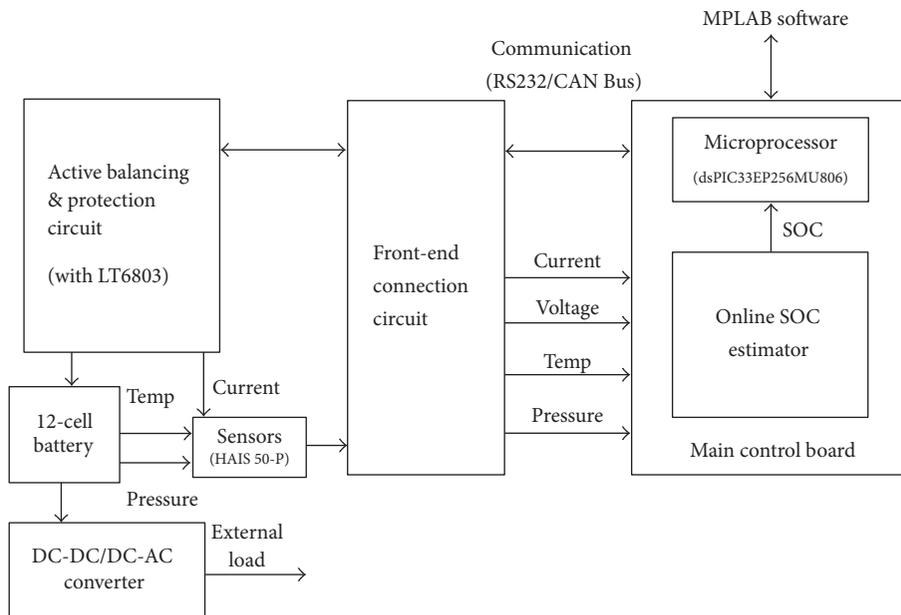


FIGURE 2: Proposed battery management system architecture with several subsystems for SOC estimation and active cell balancing.

An open frame design was used to accommodate all circuit boards, sensors, and battery cells in the prototyping stage.

The battery management system (BMS) will perform SOC estimation and active cell balancing on a single board. Figure 2 shows the proposed subsystems used in the architecture of the BMS board. Based on the power consumption, the main control board will compute the desired voltage and current. A lookup table will be used to record the optimal current and voltage of the battery at different power and voltage condition. A DC/DC converter will be used to interface between the cells and external load. The active balancing and protection (ABP) circuit will balance the SOC among the 12 cells via the multicell battery stack monitoring using the LT6803 chip. The analog signals from the current, voltage, temperature, and pressure sensor will be processed in the front-end connection (FEC) circuit. The measured data from the sensor will be transmitted via the serial port of the microprocessor. The control signal will send to the 12 cells

via the ABP and FEC circuits. The online SOC estimator in Figure 2 can estimate the SOC based on the sampled data obtained via the front-end connection circuit. The SOC estimation and cell balancing algorithms were coded in C++ via MPLAB software on the host computer or laptop.

The active balancing circuit consists of relays, DC-DC converters, and current transducers. The relays and DC-DC converters were connected to the batteries as shown in Figure 3. The relays are represented by the switch symbols. One DC-DC converter (DC/DC2) was used to balance Cell #1 to Cell #6 (i.e., Pack #1) and another DC-DC converter (DC/DC1) was used to balance Cell #7 to Cell #12 (i.e., Pack #2). As shown in the inputs of the DC/DCs in Figure 3, the DC/DC2 was powered by Cell #8 to Cell #12 while DC/DC1 was powered by Cell #1 to Cell #5. Relays were employed to make these connections and enable the connections from DC/DC2 output to Cell #1 to Cell #6 and DC/DC1 output to Cell #7 to 12 independently. There exist four current

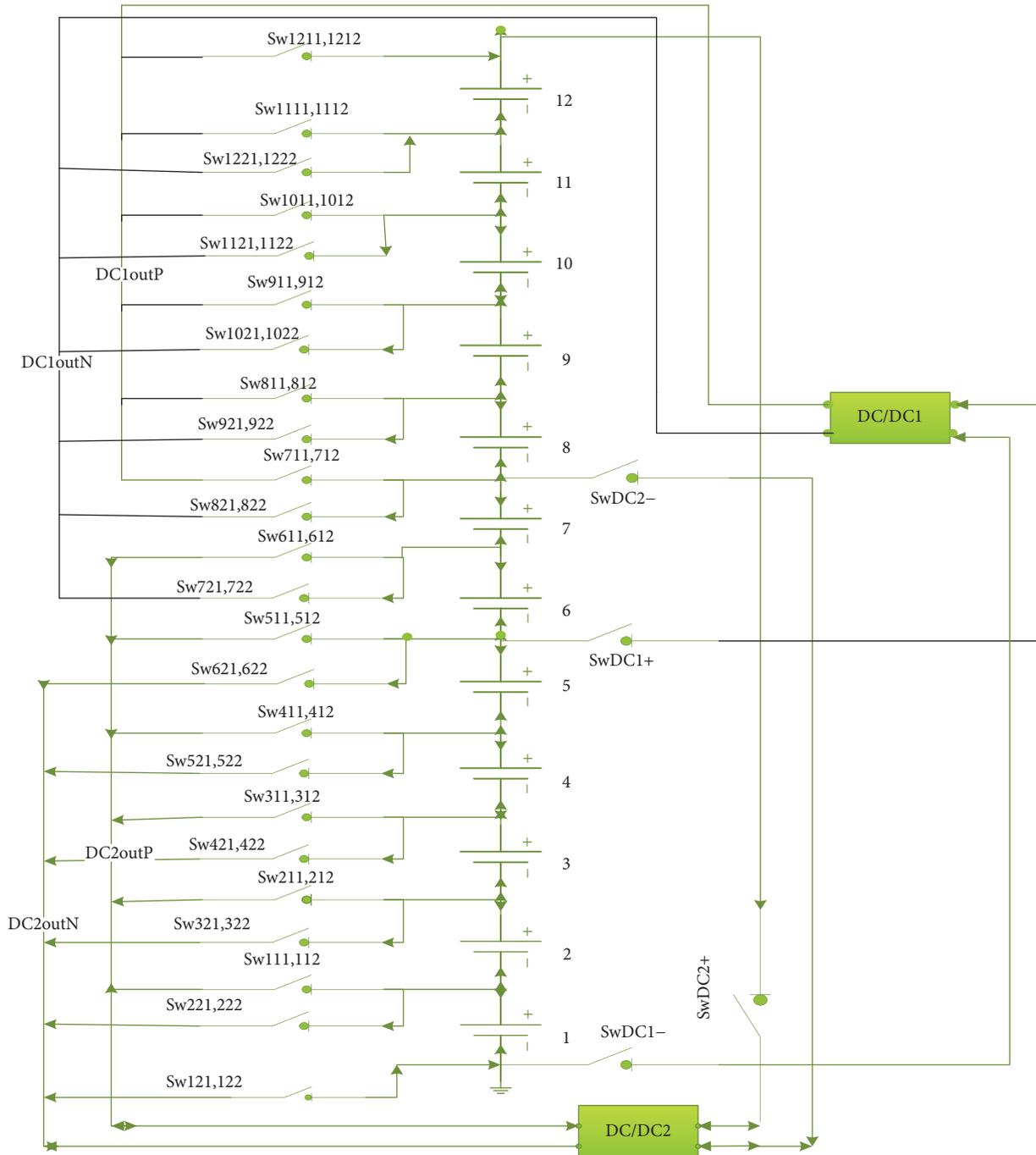


FIGURE 3: Active cell balancing scheme.

transducers to monitor the current drawn and delivered by both the DC-DC converters. The current transducers were configured to measure ± 2 A, ± 3 A, or ± 6 A.

The flowchart of the active cell balancing is shown in Figure 4. Firstly, the system checks for any abnormalities condition using the estimated SOC. If the SOC is lower or higher than a preset limit, the cell will not be used. The remaining normal cells will be deployed. The SOC for each cell in a group of six will be ranked. There exist two groups of six cells, namely, Pack #1 and Pack #2. The cells with unequal

voltages will be balanced. For example, the cell with highest SOC will be discharged while the cell with lowest SOC will be charged sequentially until the cells have equal voltage levels. The duration of the cycle depends on the time taken for the cells' balancing.

3. SOC Estimation

Many techniques were proposed for SOC estimation. However, the equivalent circuit model in Figure 5 was used. This

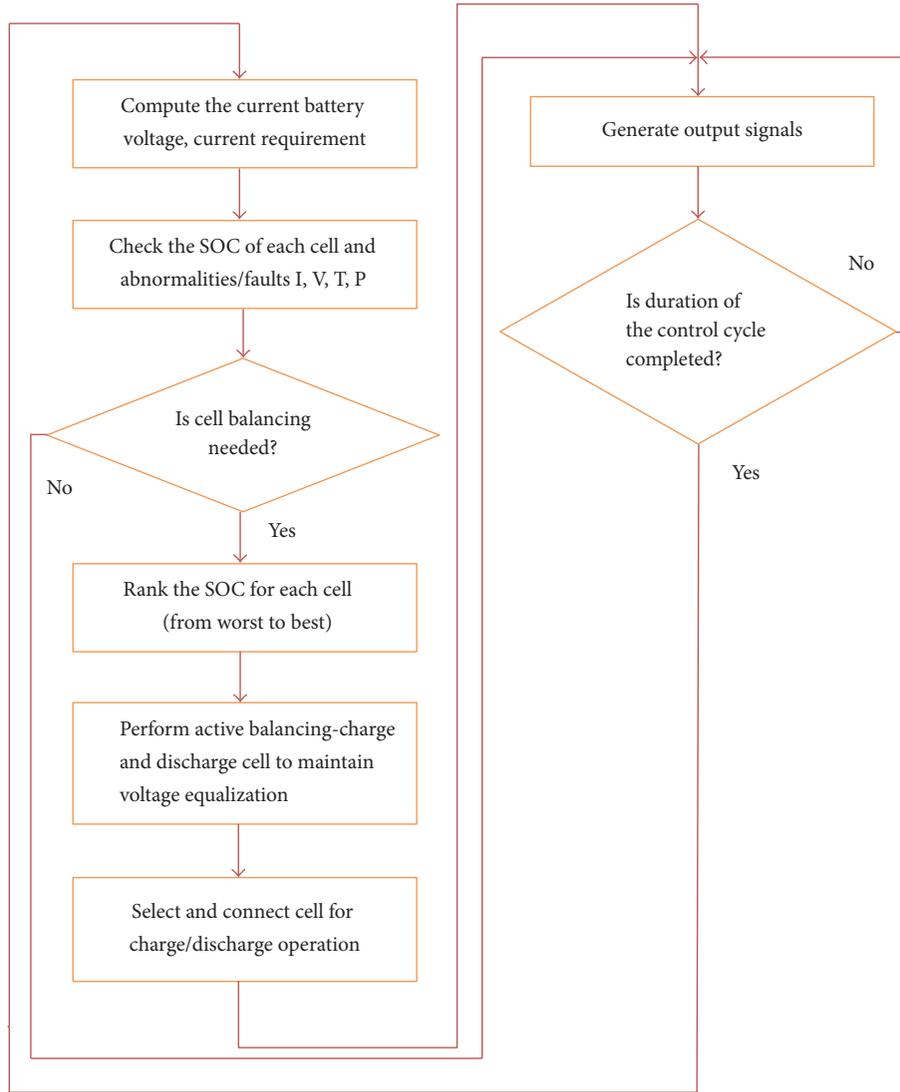


FIGURE 4: Active cell balancing algorithm flowchart.

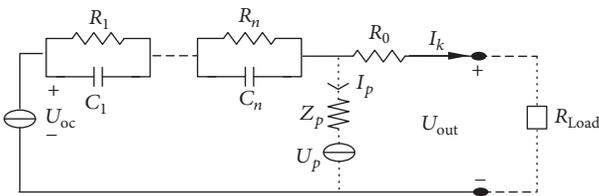


FIGURE 5: Classical topology of Thevenin equivalent circuit model.

model consists of open-circuit voltage (OCV) source U_{oc} with an indefinite number of parallel RC branches and a series resistance R_0 . The model also contains a parasitic branch that represents self-charge and charging losses denoted by I_p . Instead of using indefinite RC branches as seen in Figure 5, a 2-RC equivalent circuit was used. The MATLAB/Simscap was utilized to model the cell's SOC using the 2-RC equivalent circuit. As seen in Figure 6, the Simscap model consists

of five lookup tables, namely, R_0 , R_1 , C_1 , R_2 , and C_2 at a different ambient temperature from 5°C to 45°C . The data in the lookup data was further optimized using the Parameter Estimation Toolbox at different SOC. The different 2-RC parameters at various SOC are tabulated in Table 1.

The dynamic equation of SOC can be given as

$$\dot{\text{SOC}} = -\frac{\eta I(t)}{C_n}, \quad (1)$$

where C_n is the battery cell capacity and η is the coulomb efficiency.

From the structure in Figure 5, the dynamic equations of the voltages U_1 and U_2 can be expressed as

$$\begin{aligned} \dot{U}_1 &= \frac{I}{C_1} - \frac{U_1}{R_1 C_1}, \\ \dot{U}_2 &= \frac{I}{C_2} - \frac{U_2}{R_2 C_2}. \end{aligned} \quad (2)$$

TABLE 1: Parameters of 2-RC equivalent circuit model.

SOC	R_0 (ohm)	R_1 (ohm)	R_2 (ohm)	C_1 (F)	C_2 (F)
0.05	0.0215	0.0152	0.0148	38848.99	90150.93
0.10	0.0026	0.0032	0.0020	1294.020	179973.3
0.19	0.0042	0.0046	0.0023	20618.27	636416.8
0.28	0.0033	0.0030	0.0024	10690.21	207895.2
0.37	0.0033	0.0018	0.0030	3880.350	117578.1
0.46	0.0026	0.0035	0.0013	67312.19	12303.82
0.54	0.0045	0.0001	0.0044	3643.250	111455.1
0.63	0.0056	0.0027	0.0006	318347.7	85798.68
0.72	0.0032	0.0036	0.0010	47656.71	11604.30
0.81	0.0022	0.0012	0.0019	1026.310	14113.7
0.90	0.0033	0.0027	0.0019	23379.15	432980.6
0.99	0.0101	0.0026	0.0027	12565.53	66708.01

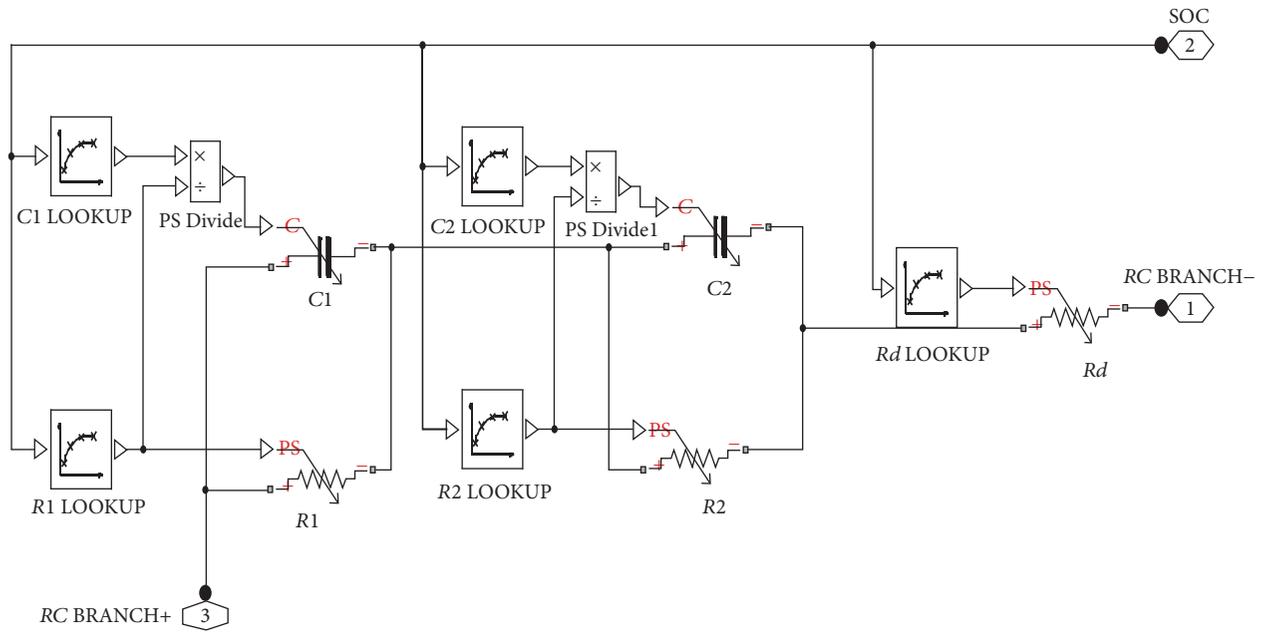


FIGURE 6: 2-RC equivalent circuit model of battery cell.

The battery terminal voltage can be written as

$$U_L = U_{oc} - IR_0 - U_1 - U_2, \quad (3)$$

where U_{oc} can be estimated by polynomial fitting. It is a standard method to determine the relationship between U_{oc} and SOC. But the errors are apparent during the end of the charging and discharging period. To compensate the fitting error, an exponential function and natural logarithm function are used

$$U_{oc}(\text{SOC}) = a_0 + a_1 \text{SOC} + a_2 \log \text{SOC} + a_3 \log(1 - \text{SOC}), \quad (4)$$

where a_0 , a_1 , a_2 , and a_3 are identified through open-circuit voltage- (OCV-) SOC curve fitting method.

The battery model has to be discretized in order to program the SOC algorithm into the microcontroller. In discrete time, the SOC dynamics can be given as follows:

$$\text{SOC}(k) = \text{SOC}(k-1) - \frac{\eta I(k) \Delta t}{C_n}, \quad (5)$$

where Δt is the sampling time interval.

The discrete time versions of (2) can be obtained by using zero-order hold (ZOH) process. The transformed equations are expressed as follows:

$$\begin{aligned} U_1(k+1) &= e^{-\Delta t/R_1 C_1} U_1(k) + R_1 (1 - e^{-\Delta t/R_1 C_1}) I(k), \\ U_2(k+1) &= e^{-\Delta t/R_2 C_2} U_2(k) + R_2 (1 - e^{-\Delta t/R_2 C_2}) I(k). \end{aligned} \quad (6)$$

Define the state vector $x_k = [\text{SOC}(k) \ U_1(k) \ U_2(k)]^T$; the following state space model of the battery cell in discrete matrix form can be obtained:

$$\begin{aligned} & \begin{bmatrix} \text{SOC}(k) \\ U_1(k) \\ U_2(k) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & e^{-\Delta t/R_1 C_1} & 0 \\ 0 & 0 & e^{-\Delta t/R_2 C_2} \end{bmatrix} \begin{bmatrix} \text{SOC}(k-1) \\ U_1(k-1) \\ U_2(k-1) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{\eta I(k) \Delta t}{C_n} \\ R_1 (1 - e^{-\Delta t/R_1 C_1}) \\ R_2 (1 - e^{-\Delta t/R_2 C_2}) \end{bmatrix} I(k-1) + w(k-1). \end{aligned} \quad (7)$$

The battery terminal voltage at any time sample k is written as such

$$U_L(k) = U_{oc}(\text{SOC}(k)) - I(k) R_0 - U_1(k) - U_2(k). \quad (8)$$

In this model, the term $U_{oc}(\text{SOC}(k))$ function is a nonlinear equation. The state and output equations in discrete time is expressed as follows:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + w_k, \\ y_k &= g(x_k, u_k) + v_k, \end{aligned} \quad (9)$$

where $f(x_k, u_k)$ and $g(x_k, u_k)$ are continuously differentiable nonlinear functions, w_k is the process noise with zero means, and v_k is the measurement noise, which is independent of w_k with zero mean value.

The Strong Tracking Filter (STF) with online adaptively modified Kalman gain matrix and prior state error covariance to track the sudden change in the state vectors was used. The critical feature of the STF is the method to rearrange the prior error covariance \mathbf{P}_k^- by multiplying it by a diagonal matrix Λ_k in which the differing diagonal entries optimize the propagation of components in state vector by diminishing the impacts of old data on current parameter estimation

$$\mathbf{P}_k^- = \Lambda_k \mathbf{G}_k \mathbf{P}_{k-1}^+ \mathbf{G}_k + \beta_k \mathbf{Q}_k \beta_k, \quad (10)$$

where Λ_k denotes multiple fading factors matrix that is determined as

$$\Lambda_k = \text{diag}(\lambda_1, \lambda_1, \dots, \lambda_n). \quad (11)$$

The proportion of λ and its constraint can be realized by prior knowledge of the system

$$\begin{aligned} \lambda_1 : \lambda_2 : \dots : \lambda_n &= c_k (a_1 : a_2 : \dots : a_n), \\ \lambda_i &= \max(1, c_k (a_i)), \end{aligned} \quad (12)$$

where $a_i \geq 1$, $i = 1, 2, \dots, n$, are predetermined constants which reflect the distinctive fading rate of the state estimation,

a_i is set to be of relatively larger value when it comes to circumstance that i th component in the state vector changes much faster as compared to the others, and c_k is the standard factor of λ_n that is given as follows:

$$\begin{aligned} V_k &= \begin{cases} S_1 S_1^T, & k = 0 \\ \frac{\rho V_{k-1} + S_k S_k^T}{1 + \rho}, & k \geq 1, \end{cases} \\ N_k &= V_k - \beta R_k - H_k \beta_k Q_k \beta_k^T H_k^T, \\ M_k &= \text{diag}(a_1, a_2, \dots, a_n) G_k P_{k-1}^+ G_k^T H_k^T H_k, \\ c_k &= \frac{\text{trace}(N_k)}{\text{trace}(M_k)}, \end{aligned} \quad (13)$$

where S_k denotes the difference of output calculated by STF and measurement value. $\rho = 0.95$ and $\beta \geq 1$ are forgetting and weakening factors, respectively.

The STEKF was developed by applying the strong tracking algorithm on the EKF. STEKF benefits the estimation process by taking the advantages of EKF that minimizes the estimation error covariance and STF that track the state vector variation accurately. In addition, cubature Kalman filter generates a set of cubature points propagated by system equations to approximate the posterior estimate which was used. It can also track the fast-changing SOC value. As a result, the strong tracking with cubature on extended Kalman filter algorithm (STCEKF) was formed. For clarity, the steps in the algorithm of STCEKF are given as follows.

In the time update, the cubature points are generated

$$\begin{aligned} X_{i,k-1}^+ &= S_{k-1}^+ \varepsilon_i + \hat{x}_{k-1}^+, \quad i = 1, 2, \dots, n, \\ \varepsilon_i &= \begin{cases} \sqrt{n} [1]_i, & i = 1, 2, \dots, n \\ -\sqrt{n} [1]_i, & i = n+1, n+2, \dots, 2n, \end{cases} \end{aligned} \quad (14)$$

where $S_{k-1}^+ (S_{k-1}^+)^T = P_{k-1}^+$ and n is the dimension of the state vector.

The cubature points are propagated

$$\bar{X}_{i,k} = g(X_{i,k-1}^+, u_{k-1}). \quad (15)$$

The predicted state is estimated

$$\hat{x}_k^- = \frac{1}{2n} \sum_{i=1}^n \bar{X}_{i,k}. \quad (16)$$

The predicted error covariance is calculated

$$P_k^- = \frac{1}{2n} \sum_{i=1}^n \bar{X}_{i,k} (\bar{X}_{i,k})^T - \hat{x}_k^- (\hat{x}_k^-)^T + \beta_k Q_k \beta_k. \quad (17)$$

In the measurement update, the cubature points are generated

$$\bar{X}_{i,k}^- = S_k^- \varepsilon_i + \hat{x}_k^-, \quad i = 1, 2, \dots, n. \quad (18)$$

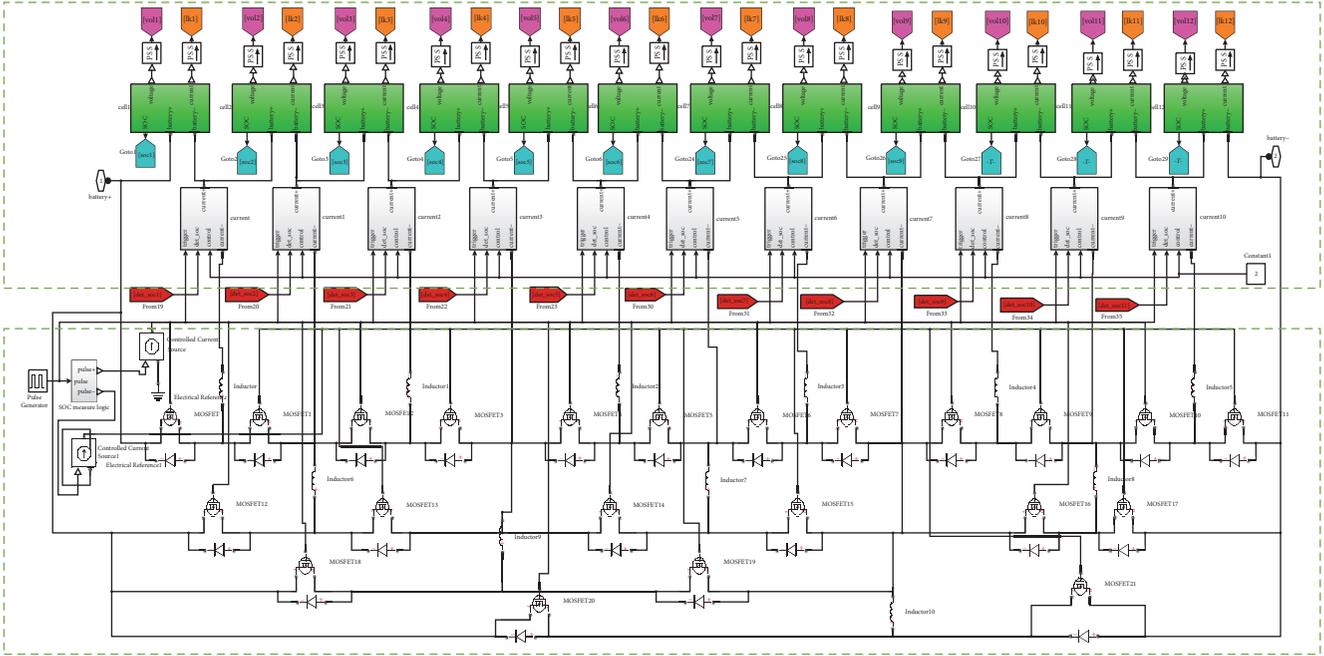


FIGURE 7: Example of battery model including SOC estimator (top) and cell balancing (bottom).

The propagated cubature points are evaluated

$$Z_{i,k} = h(\tilde{X}_{i,k}, u_k). \quad (19)$$

The predicted measurement is estimated

$$\hat{z}_k = \frac{1}{2n} \sum_{i=1}^n Z_{i,k}. \quad (20)$$

The innovation covariance matrix is obtained

$$P_{zz,k} = \frac{1}{2n} \sum_{i=1}^n Z_{i,k} (Z_{i,k})^T - z_k (z_k)^T + \gamma_k R_k \gamma_k. \quad (21)$$

The cross-covariance matrix is determined

$$P_{xz,k} = \frac{1}{2n} \sum_{i=1}^n \tilde{X}_{i,k} (Z_{i,k})^T - \hat{x}_k^- (\hat{z}_k)^T. \quad (22)$$

The Kalman gain is calculated

$$K_k = P_{xz,k} (P_{zz,k})^{-1}. \quad (23)$$

The updated state is estimated

$$x_k^+ = x_k^- + K_k (z_k - \hat{z}_k). \quad (24)$$

The corresponding error covariance is computed

$$P_k^+ = P_k^- - K_k P_{zz,k} (K_k)^T. \quad (25)$$

The simulation model that includes the SOC estimation using STCEKF and the active cell balancing can be seen in Figure 7.

4. Experimental Tests

The battery was tested in a laboratory setup [8] as shown in Figure 8. The twelve lithium iron phosphate battery cells (ANR26650M1-B) were used during the test. The specifications of the cell can be obtained from a123batteries.com datasheet. The load current was generated by a DC electronic load while the battery cells were charged by a programmable DC power supply. It was used to control the voltage or current source with the output voltage (maximum at 36 V) and current (maximum at 20 A). The host PC communicates with the DAQ device to measure the charging and discharging of each cell. The NI DAQ device controlled the outputs and inputs data with acquisition rate set to 1 sample per second. A current sensor measured the current during the charging and discharge operation for one battery cell. As all the battery cells are unique due to the manufacturing variation, the test was repeated for other 11 cells. The experimental data for the 2-RC model for the 12 cells were obtained. The validated 2-RC models were updated into the series of lookup tables as shown in Figure 6.

4.1. Proposed Battery Model Validation. The prototype of the battery power system can be seen in Figure 9. The prototype was used to validate the 2-RC cell models at different ambient temperatures from 5°C, 15°C, 25°C, 35°C, and 45°C in an environmental chamber. For example, the root means square or RMS errors between the simulation and experimental results at 25°C for the 12-cell are shown in Table 2. The RMS error of Cell #1 terminal voltage between the simulation and experiment was around 7.20×10^{-5} at 25°C (see Table 2). The experimental data at a different ambient temperature of 25°C was compared with the model to test the robustness

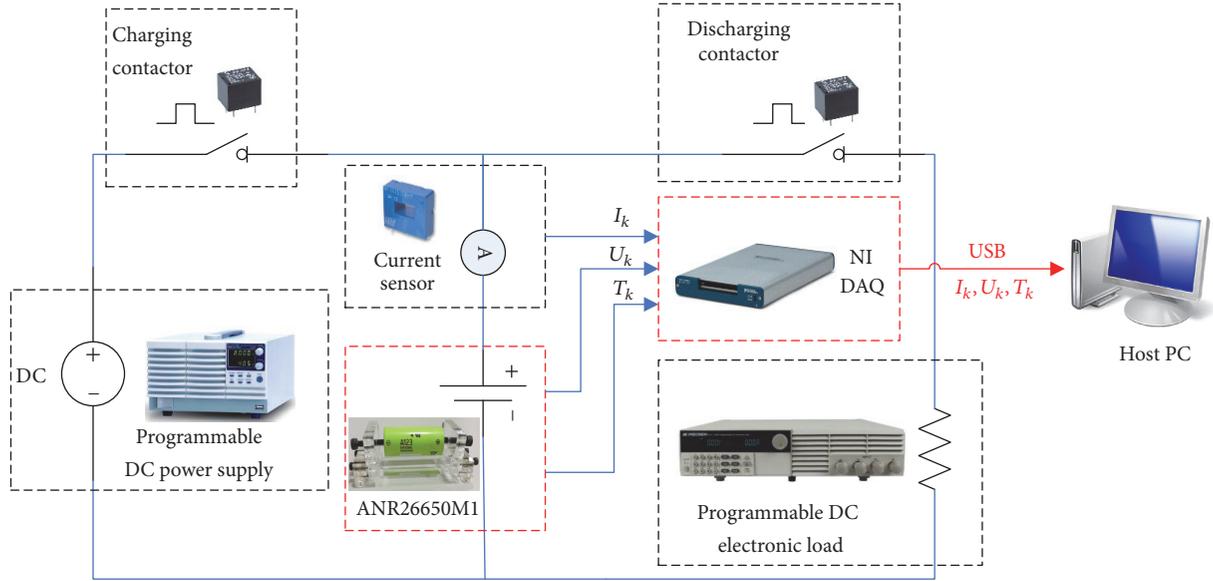


FIGURE 8: Test bench setup for battery cell testing [8].

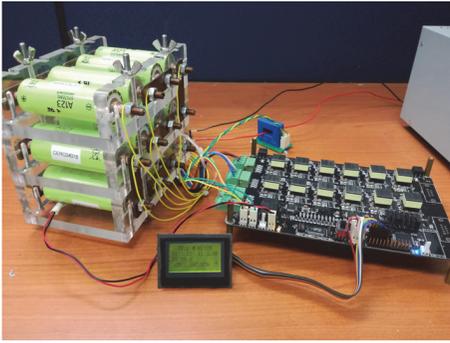


FIGURE 9: 12-cell battery pack prototype for multicell testing.

of the numerical model in Figure 10. The result shows that the battery model output can estimate the terminal voltage with small error. The errors of the terminal voltage due to the current fluctuation can converge to zero within 1.20×10^{-5} s. The result showed that the proposed battery model was accurate. Note that the results are valid for the type of battery cells used.

4.2. SOC Estimation Algorithms Comparison. Several SOC estimation algorithms such as EKF, STEKF, CKF, STCEKF were compared using the 2-RC equivalent circuit battery cell models obtained at 25°C. The following test process was used to evaluate the proposed SOC estimation algorithm with other algorithms.

(1) *Initialization.* To determine the benchmark of the SOC, the battery cell needs to be fully charged followed by a 15-hour relaxation period before running the algorithms. It helped to

TABLE 2: RMSE between simulation and experiment of terminal voltage at 25°C.

Cell number	Root means square error (RMSE) of terminal voltage
Cell 01	2.26×10^{-5}
Cell 02	8.98×10^{-6}
Cell 03	1.88×10^{-5}
Cell 04	2.34×10^{-5}
Cell 05	2.54×10^{-5}
Cell 06	1.99×10^{-5}
Cell 07	2.37×10^{-5}
Cell 08	2.51×10^{-5}
Cell 09	1.92×10^{-5}
Cell 10	2.00×10^{-5}
Cell 11	1.74×10^{-5}
Cell 12	2.19×10^{-5}

ensure that the initial SOC of the battery cell is 100% in its equilibrium state.

(2) *Compute Actual SOC.* A reference ampere-hour method was used to determine the actual SOC for comparison.

(3) *Test Different Algorithms.* An initial SOC error was set to 20% for testing the convergence performance using different SOC estimation algorithms.

(4) *Comparison Process.* The root means square error (RMSE) was computed to evaluate the estimation performance for various SOC estimation algorithms. As shown in Figure 11, the initial transient stage of the SOC estimation was quite

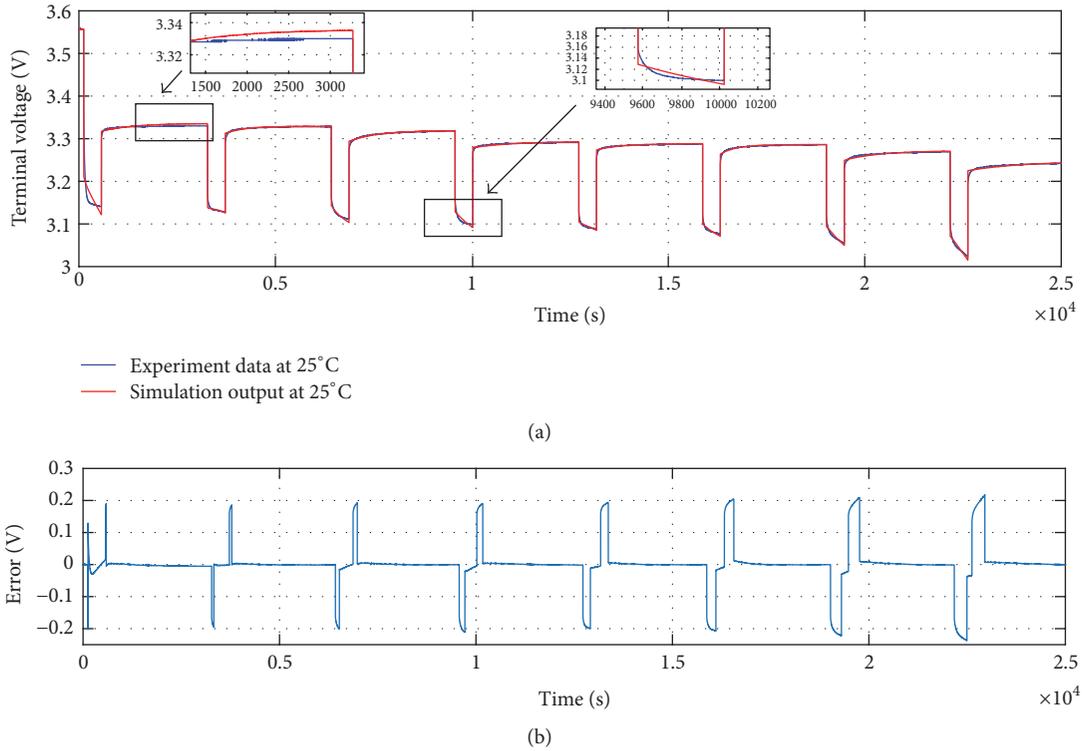


FIGURE 10: (a) Cell #1 model validated at 25°C. (b) Cell #1 error between simulation and experiment results at 25°C.

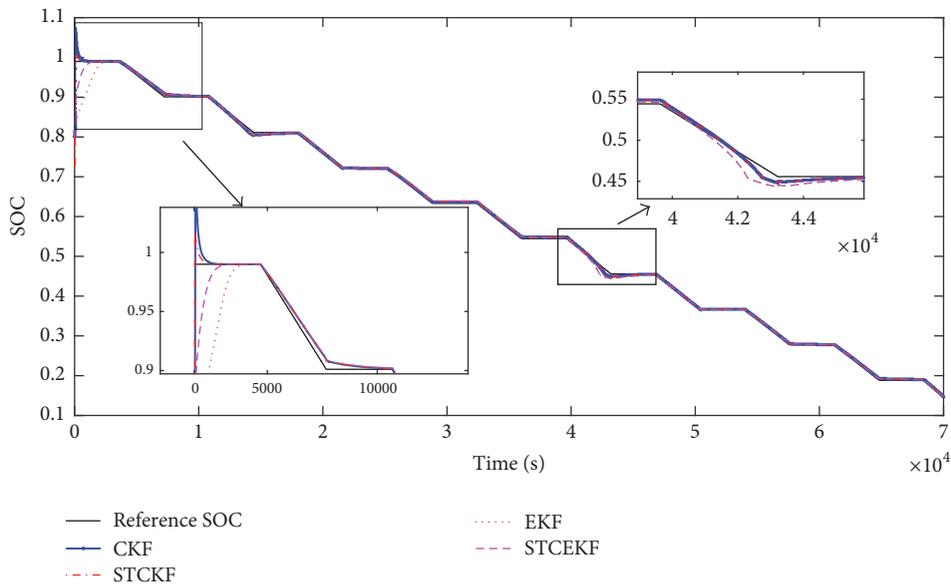


FIGURE 11: SOC estimation results using EKF, STEKF, CKF, and STCEKF at 25°C.

different for each method. The CKF exhibits an overshoot while EKF was quite sluggish in the response. As compared to the STCEKF, there was a smaller overshoot with a faster response as compared to the other methods. However, the STEKF deviated from the reference SOC curve during the discharging. Most of the SOC estimation results could converge to the reference SOC. The RMSE values and the

convergence time of the SOC estimation are tabulated in Table 3. It can be seen that STCEKF exhibits the smallest RMSE and the shorter convergence time as compared to other methods. Therefore, the STCEKF has faster and more accurate SOC estimation as compared to EKF, STEKF, and CKF. Note that similar behavior can be found at a different temperature.

TABLE 3: RMSE for SOC estimation using EKF, STEKF, CKF, and STCEKF at 25°C.

Descriptions	Extended Kalman filter (EKF)	Strong tracking extended Kalman filter (STEKF)	Cubature Kalman filter (CKF)	Strong tracking with cubature on extended Kalman filter (STCEKF)
Root means square error (RMSE)	0.02735	0.01455	0.01223	0.01176
Convergence time (min)	35	20	11.7	10

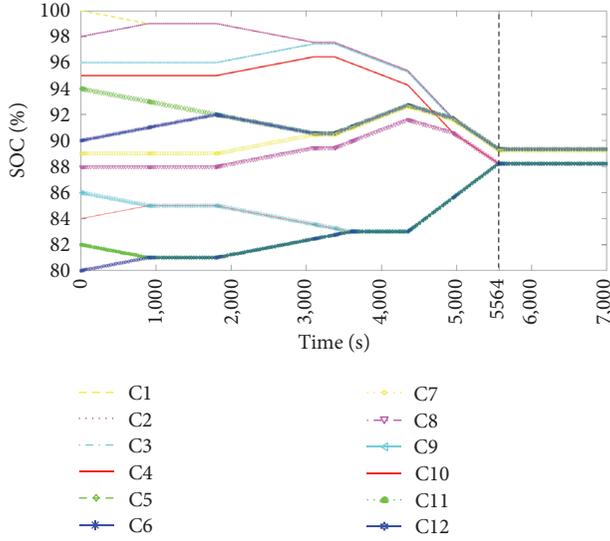


FIGURE 12: Active cell balancing results using proposed active cell balancing circuit for all cells at 25°C.

As mentioned in the above section, the balancing routine was used to balance the imbalanced cells. Firstly, the balancing routine will seek for the lowest, highest, and average SOC of the cells. The cell number with the lowest and highest SOC will be logged. Secondly, the balancing routine will determine whether the BMS has been balanced by using the lowest, highest, and average SOC of the cells. If all the cells were balanced, the balancing routine would turn off all the relays that control the active cell balancing. Thirdly, the balancing routine will execute the balancing for the cell with the lowest SOC first. Note that one DC-DC converter will be used to balance Cell #1 to #6, and another DC-DC converter will be used to balance Cell #7 to #12.

Figure 12 shows the experimental results of SOC for each cell (labeled as C1 to C12) after the active cell balancing. As seen in Figure 12, the SOC of each cell was maintained at the same value at the end of the cell balancing. Most of the cells exhibited same SOC value at the end of the balancing time frame of 5564 s. The SOC of each cell converged to similar value while the battery pack was running. It did not have to wait for 5564 s for powering. A comparative study between different active balancing topologies was shown in Table 4 to further examine the balancing speed and the balancing loss between these methods. In Table 4, the capacitors showed a slower balancing speed than the transformer. Although the proposed algorithm for the active cell balancing method was slightly complicated than other methods, it demonstrated

some merits of higher balancing speed and lower balancing loss for the battery management system. For brevity, only the cell balancing test at 25°C was shown. The similar test at different ambient temperatures can be repeated.

4.3. SOC Estimation under Realistic Load Profile and Varying Ambient Temperature. To further validate the proposed ambient temperature model and SOC estimation algorithm under the practical and dynamic situation, the New European Driving Cycle load profile [49] was applied to the 12-cell series battery pack prototype to simulate the electric vehicle applications under varying the ambient temperature. The ambient temperature variation can be seen in Figure 13(a). The current load profile of the electric vehicle subjected to the temperature disturbance is depicted in Figure 13(b). The load profile was scaled down to fit the battery pack. However, it can be adjusted to different current level. The programmable DC electronic load was used to run the preprogrammed load profile. Before the experiment, the battery pack was fully charged to 100% in Figure 13(c). The instantaneous error was determined based on the difference between the respective methods (i.e., STCEKF and EKF) and the actual SOC value obtained by the Ah counting method. As observed in Figure 13(c), the proposed SOC estimation can follow the actual SOC value by the Ah counting method under the changing operating ambient temperature. Despite the ambient temperature perturbation, the SOC error of the STCEKF method (see Figure 13(d)) settled to a smaller steady-state error after a short period of drift as compared to EKF-based SOC model that was unable to recover from the ambient temperature disturbances. In summary, the proposed STCEKF is more robust than EKF in the SOC estimation. The graphs for the STEKF and CKF were not available at the time of comparison. In addition, the tests can be repeated for different cell type and current load profiles at different temperatures.

5. Conclusions

The first strong tracking cubature extended Kalman filter (STCEKF) and active cell balancing for the lithium iron phosphate battery system model were jointly developed. The SOC estimation using the STCEKF produced the lowest error and faster computational time as compared with the extended Kalman filter (EKF). A new cell balancing circuit and algorithm showed a higher balancing speed and less balancing loss during the charging and discharging. The proposed battery power system design was validated by both model simulation and experiment using the actual battery prototype. The New European Driving Cycle load profile

TABLE 4: Comparison of different active cell balancing methods at 25°C.

	Capacitor	Switched capacitor (STSC)	Inductor	Proposed method	Transformer
Balancing speed	11330 s	8550 s	6833 s	5564 s	6064 s
Average state-of-charge (efficiency)	90.0%	90.0%	89.2%	89.5%	89.3%
Cost	4	3	3	3	1
Size	3	2	3	3	1
Complexity	4	4	1	2	3

Note. Cost (4: cheap, 1: expensive); size (4: small, 1: big); complexity (4: low, 1: high).

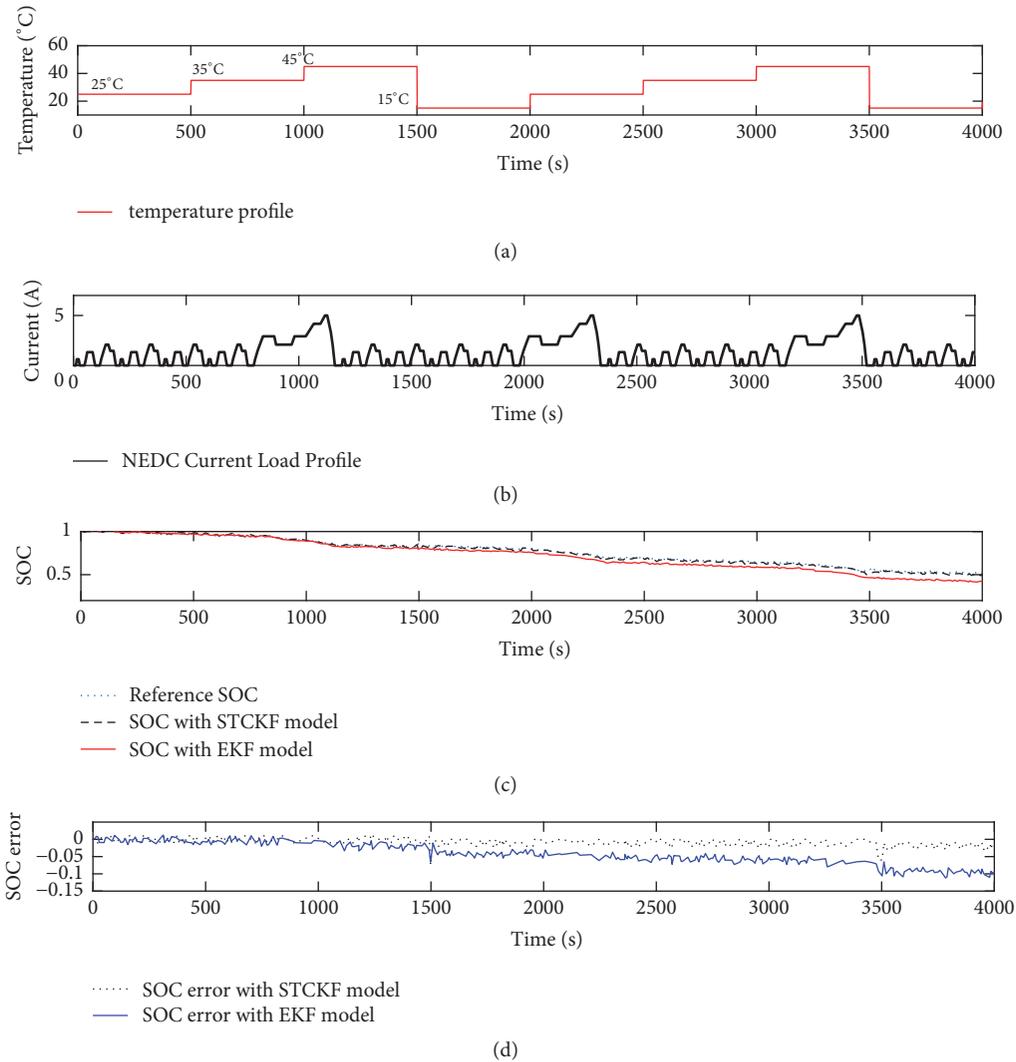


FIGURE 13: (a) Ambient temperature profile; (b) current profile; (c) SOC comparison of proposed STCKF and EKF-based model; (d) SOC error comparison.

for an electric vehicle was used to verify and compare the proposed SOC estimation performance by the STCKEF with the EKF. The experimental results showed that the STCKEF performed better than the EKF under the varying ambient temperature and current load profile (that was equivalent to

the actual load profile in the electric vehicle). In summary, the proposed STCKEF was successfully implemented and validated in the actual 12-cell battery pack with less SOC estimation error, faster balancing time, and less balancing loss for real-time application. For future works, the battery

model will be fine-tuned and further developed to detect and diagnose uncertain faults subjected to external disturbances with less computation burden.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge Newcastle University in Singapore and Temasek Polytechnic for providing the necessary financial support and equipment.

References

- [1] P. Germano and Y. Perriard, "Battery Charger for Electric Vehicle Based on a Wireless Power Transmission," in *Proceedings of the 13th IEEE Vehicle Power and Propulsion Conference, VPPC 2016*, China, October 2016.
- [2] O. Gomofov, J. P. Trovao, X. Kestelyn, and M. R. Dubois, "Adaptive Energy Management System Based on a Real-Time Model Predictive Control With Nonuniform Sampling Time for Multiple Energy Storage Electric Vehicle," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 5520–5530, 2017.
- [3] E. A. Grunditz and T. Thiringer, "Performance Analysis of Current BEVs Based on a Comprehensive Review of Specifications," *IEEE Transactions on Transportation Electrification*, vol. 2, no. 3, pp. 270–289, 2016.
- [4] H. Cave, "Charging ahead? the bid for better batteries," *Journal of Engineering and Technology*, vol. 12, no. 2, pp. 58–61, 2017.
- [5] A. Castaings, W. Lhomme, R. Trigui, and A. Bouscayrol, "Practical control schemes of a battery/supercapacitor system for electric vehicle," *IET Electrical Systems in Transportation*, vol. 6, no. 1, pp. 20–26, 2016.
- [6] Y. Wang, H. Fang, Z. Sahinoglu, T. Wada, and S. Hara, "Adaptive Estimation of the State of Charge for Lithium-Ion Batteries: Nonlinear Geometric Observer Approach," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 3, pp. 948–962, 2015.
- [7] Y. Xing, W. He, M. Pecht, and K. L. Tsui, "State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures," *Applied Energy*, vol. 113, pp. 106–115, 2014.
- [8] Z. Gao, C. S. Chin, W. L. Woo, and J. Jia, "Integrated equivalent circuit and thermal model for simulation of temperature-dependent LiFePO₄ battery in actual embedded application," *Energies*, vol. 10, no. 1, article 85, 2017.
- [9] T. K. Dong, A. Kirchev, F. Mattera, J. Kowal, and Y. Bultel, "Dynamic modeling of Li-ion batteries using an equivalent electrical circuit," *Journal of The Electrochemical Society*, vol. 158, no. 3, pp. A326–A336, 2011.
- [10] D. Andre, M. Meiler, K. Steiner, H. Walz, T. Soczka-Guth, and D. U. Sauer, "Characterization of high-power lithium-ion batteries by electrochemical impedance spectroscopy. II: Modelling," *Journal of Power Sources*, vol. 196, no. 12, pp. 5349–5356, 2011.
- [11] X. S. Hu, S. B. Li, and H. Peng, "A comparative study of equivalent circuit models for Li-ion batteries," *Journal of Power Sources*, vol. 198, pp. 359–367, 2012.
- [12] W. Waag, C. Fleischer, and D. U. Sauer, "Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles," *Journal of Power Sources*, vol. 258, pp. 321–339, 2014.
- [13] S. G. Li, S. M. Sharkh, F. C. Walsh, and C. N. Zhang, "Energy and battery management of a plug-in series hybrid electric vehicle using fuzzy logic," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 8, pp. 3571–3585, 2011.
- [14] W. He, N. Williard, C. Chen, and M. Pecht, "State of charge estimation for Li-ion batteries using neural network modeling and unscented Kalman filter-based error cancellation," *International Journal of Electrical Power & Energy Systems*, vol. 62, pp. 783–791, 2014.
- [15] S.-C. Wang and Y.-H. Liu, "A PSO-Based Fuzzy-Controlled Searching for the Optimal Charge Pattern of Li-Ion Batteries," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 2983–2993, 2015.
- [16] P. Melin and O. Castillo, "Intelligent control of complex electrochemical systems with a neuro-fuzzy-genetic approach," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 5, pp. 951–955, 2001.
- [17] H. Chen and G. Wu, "Compensation fuzzy neural network power management strategy for hybrid electric vehicle," *Tongji Daxue Xuebao/Journal of Tongji University*, vol. 37, no. 4, pp. 525–530, 2009.
- [18] V. Klass, M. Behm, and G. Lindbergh, "Capturing lithium-ion battery dynamics with support vector machine-based battery model," *Journal of Power Sources*, vol. 298, pp. 92–101, 2015.
- [19] P. Gao, C. Zhang, and G. Wen, "Equivalent circuit model analysis on electrochemical impedance spectroscopy of lithium metal batteries," *Journal of Power Sources*, vol. 294, pp. 67–74, 2015.
- [20] J. Smekens, J. Paulsen, W. Yang et al., "A Modified Multiphysics model for Lithium-Ion batteries with a Li_xNi_{1/3}Mn_{1/3}Co_{1/3}O₂ electrode," *Electrochimica Acta*, vol. 174, Article ID 25140, pp. 615–624, 2015.
- [21] H. Fang, Y. Wang, Z. Sahinoglu, T. Wada, and S. Hara, "State of charge estimation for lithium-ion batteries: An adaptive approach," *Control Engineering Practice*, vol. 25, no. 1, pp. 45–54, 2014.
- [22] J. Du, Z. Liu, and Y. Wang, "State of charge estimation for Li-ion battery based on model from extreme learning machine," *Control Engineering Practice*, vol. 26, no. 1, pp. 11–19, 2014.
- [23] J. Du, Z. Liu, C. Chen, and Y. Wang, "Li-ion battery SOC estimation using EKF based on a model proposed by extreme learning machine," in *Proceedings of the 7th IEEE Conference on Industrial Electronics and Applications*, pp. 1648–1653, Singapore, Singapore, 2012.
- [24] A. Densmore and M. Hanif, "Modeling the condition of lithium ion batteries using the extreme learning machine," in *Proceedings of the 2016 IEEE PES PowerAfrica Conference, PowerAfrica 2016*, pp. 184–188, Livingstone, Zambia, July 2016.
- [25] R. Razavi-Far, S. Chakrabarti, and M. Saif, "Multi-step parallel-strategy for estimating the remaining useful life of batteries," in *Proceedings of the 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1–4, Windsor, ON, Canada, April 2017.
- [26] M. Doyle, T. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *Journal of The Electrochemical Society*, vol. 140, no. 6, pp. 1526–1533, 1993.
- [27] T. F. Fuller, M. Doyle, and J. Newman, "Simulation and optimization of the dual lithium ion insertion cell," *Journal of The Electrochemical Society*, vol. 141, no. 1, pp. 1–10, 1994.

- [28] S. Atlung, K. West, and T. Jacobsen, "Dynamic Aspects of Solid Solution Cathodes for Electrochemical Power Sources," *Journal of The Electrochemical Society*, vol. 126, no. 8, pp. 1311–1321, 1979.
- [29] S. Khaleghi Rahimian, S. Rayman, and R. E. White, "Extension of physics-based single particle model for higher charge-discharge rates," *Journal of Power Sources*, vol. 224, pp. 180–194, 2013.
- [30] K. A. Smith, C. D. Rahn, and C.-Y. Wang, "Model-based electrochemical estimation of lithium-ion batteries," in *Proceedings of the 17th IEEE International Conference on Control Applications, CCA*, pp. 714–719, USA, September 2008.
- [31] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—part 1. Background," *Journal of Power Sources*, vol. 134, no. 2, pp. 252–261, 2004.
- [32] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—part 2. Modeling and identification," *Journal of Power Sources*, vol. 134, no. 2, pp. 262–276, 2004.
- [33] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: part 3. State and parameter estimation," *Journal of Power Sources*, vol. 134, no. 2, pp. 277–292, 2004.
- [34] X. Chen, W. Shen, Z. Cao, and A. Kapoor, "A novel approach for state of charge estimation based on adaptive switching gain sliding mode observer in electric vehicles," *Journal of Power Sources*, vol. 246, pp. 667–678, 2014.
- [35] D. Kim, K. H. Koo, J. J. Jeong, T. D. Goh, and S. W. Kim, "Second-order discrete-time sliding mode observer for state of charge determination based on a dynamic resistance Li-ion battery model," *Energies*, vol. 6, no. 10, pp. 5538–5551, 2013.
- [36] X. Hu, F. Sun, and Y. Zou, "Estimation of state of charge of a Lithium-Ion battery pack for electric vehicles using an adaptive luenberger observer," *Energies*, vol. 3, no. 9, pp. 1586–1603, 2010.
- [37] T. He, D. Li, Z. Wu, Y. Xue, and Y. Yang, "A modified luenberger observer for SOC estimation of lithium-ion battery," in *Proceedings of the 2017 36th Chinese Control Conference (CCC)*, pp. 924–928, Dalian, China, July 2017.
- [38] Y. Wang, H. Fang, L. Zhou, and T. Wada, "Revisiting the State-of-Charge Estimation for Lithium-Ion Batteries: A Methodical Investigation of the Extended Kalman Filter Approach," *IEEE Control Systems Magazine*, vol. 37, no. 4, pp. 73–96, 2017.
- [39] S. Santhanagopalan and R. E. White, "State of charge estimation using an unscented filter for high power lithium ion cells," *International Journal of Energy Research*, vol. 34, no. 2, pp. 152–163, 2010.
- [40] R. Klein, N. A. Chaturvedi, J. Christensen, J. Ahmed, R. Findisen, and A. Kojic, "Electrochemical model based observer design for a lithium-ion battery," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 289–301, 2013.
- [41] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.
- [42] J. Jia, P. Lin, C. S. Chin et al., "Multirate strong tracking extended Kalman filter and its implementation on lithium iron phosphate (LiFePO₄) battery system," in *Proceedings of the 11th IEEE International Conference on Power Electronics and Drive Systems, PEDS 2015*, pp. 640–645, Australia, June 2015.
- [43] P. Malysz, R. Gu, J. Ye, H. Yang, and A. Emadi, "State-of-charge and state-of-health estimation with state constraints and current sensor bias correction for electrified powertrain vehicle batteries," *IET Electrical Systems in Transportation*, vol. 6, no. 2, pp. 136–144, 2016.
- [44] Z. Chen, S. Qiu, M. A. Masrur, and Y. L. Murphey, "Battery state of charge estimation based on a combined model of extended kalman filter and neural networks," in *Proceedings of the 2011 International Joint Conference on Neural Network, IJCNN 2011*, pp. 2156–2163, USA, August 2011.
- [45] A. Sidhu, A. Izadian, and S. Anwar, "Adaptive nonlinear model-based fault diagnosis of li-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 2, pp. 1002–1011, 2015.
- [46] X. He, Z. Wang, X. Wang, and D. H. Zhou, "Networked strong tracking filtering with multiple packet dropouts: algorithms and applications," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1454–1463, 2014.
- [47] T. Kim, W. Qiao, and L. Qu, "Power electronics-enabled self-X multicell batteries: A design toward smart batteries," *IEEE Transactions on Power Electronics*, vol. 27, no. 11, pp. 4723–4733, 2012.
- [48] M. Daowd, N. Omar, P. van den Bossche, and J. van Mierlo, "A review of passive and active battery balancing based on MATLAB/Simulink," *International Review of Electrical Engineering*, vol. 6, no. 7, pp. 2974–2989, 2011.
- [49] Z. Gao, C. Chin, J. Chiew, J. Jia, and C. Zhang, "Design and Implementation of a Smart Lithium-Ion Battery System with Real-Time Fault Diagnosis Capability for Electric Vehicles," *Energies*, vol. 10, no. 10, p. 1503, 2017.

Research Article

Autonomous Path Planning for Road Vehicles in Narrow Environments: An Efficient Continuous Curvature Approach

Domokos Kiss and Gábor Tevesz

Department of Automation and Applied Informatics, Budapest University of Technology and Economics, Budapest, Hungary

Correspondence should be addressed to Domokos Kiss; domokos.kiss@aut.bme.hu

Received 5 May 2017; Accepted 29 August 2017; Published 31 October 2017

Academic Editor: Mohammad Hamdan

Copyright © 2017 Domokos Kiss and Gábor Tevesz. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper we introduce a novel method for obtaining good quality paths for autonomous road vehicles (e.g., cars or buses) in narrow environments. There are many traffic situations in urban scenarios where nontrivial maneuvering in narrow places is necessary. Navigating in cluttered parking lots or having to avoid obstacles blocking the way and finding a detour even in narrow streets are challenging, especially if the vehicle has large dimensions like a bus. We present a combined approximation-based approach to solve the path planning problem in such situations. Our approach consists of a global planner which generates a preliminary path consisting of straight and turning-in-place primitives and a local planner which is used to make the preliminary path feasible to car-like vehicles. The approximation methodology is well known in the literature; however, both components proposed in this paper differ from existing similar planning methods. The approximation process with the proposed local planner is proven to be convergent for any preliminary global paths. The resulting path has continuous curvature which renders our method well suited for application on real vehicles. Simulation experiments show that the proposed method outperforms similar approaches in terms of path quality in complicated planning tasks.

1. Introduction

Looking around in the automotive industry, we can see a great amount of automation and an accelerating pursuit to develop intelligent self-driving vehicles. According to the popularity of the field, research is considerably active in the last decades [1–5]. The navigation task of autonomous robotic vehicles consists of many subproblems, involving sensing, perception and recognition of situations, localization, motion planning, and execution. Amongst these, we treat here the problem of generating appropriate paths for robotic cars in the presence of obstacles. As usual in case of cars, we assume the workspace of interest being a planar surface. From a control theoretic view, the rolling without slipping constraint of wheels induces nonholonomic constraints, which do not prohibit the controllability of a system but cause remarkable difficulties (it is enough to think of parallel parking a car). Although controlling a nonholonomic vehicle is a control theoretic challenge, it is worth incorporating knowledge about the specific system in the path planning task as well.

This paper describes a path planning algorithm which is intended to solve navigation problems where the vehicle has to cross narrow or cluttered areas in order to reach the goal. Since cars have a nonzero minimum turning radius, such environments require in many cases nontrivial maneuvering (including more reversals) between narrowing. Our solution is appropriate for the continuous steering car, which means that the finite steering rate is taken into account as well beyond the turning radius. In other words, steering at standstill is avoided by forcing the path to have a continuous curvature profile. This is achieved by using clothoid curves between straight and circular path segments. The proposed algorithm is an approximation-based method consisting of a local and a global planner. The local planner generates feasible paths in the absence of obstacles. The global planner delivers a preliminary collision-free path by omitting the minimum turning radius constraint of the vehicle. It is shown that the proposed local and global planners are appropriate to be applied together and guarantee finding a feasible solution in the presence of obstacles if the planning task is solvable.

Moreover, the local paths are designed to be similar to those a human driver would find.

Although different human drivers and passengers prefer different types of paths, we try to summarize the main characteristics of a “human-like” path as follows. Existing optimal planners strive to return the shortest possible solution which results in sharp (mostly maximal curvature) turns and many reversals in most cases. In our opinion optimal length is not as important for a passenger as feeling comfortable. If a path is longer than optimal (to a meaningful extent, of course) but contains less sharp turns and reversals, then we treat it better from the passenger’s perspective. Our proposed approach is designed with these preferences in mind: the preliminary global path consists only of straight segments (with in-place turning between them), and the local planner has the task of connecting these segments with smooth and not necessarily maximal curvature turns.

The remaining part of the paper is organized as follows. The next section summarizes related work in path planning for car-like vehicles. Section 3 introduces the main characteristics of the proposed approach and the previous work of the authors in this field. In Section 4 we describe the local planning component (TTS-planner) in detail, in which we discuss the applied path elements, the properties of local paths, and a topological admissible steering method based on these. In Section 5 the global planning tool (RTR-planner) is described which generates the collision-free preliminary path. Finally, simulation examples are presented which confirm the advantageous characteristics of the proposed approach.

2. Related Work

The path planning task for nonholonomic systems is difficult, caused by the concurrent presence of global geometric (environmental) and local kinematic constraints. For this reason most available approaches decouple the treatment of these in two separate phases of the planning process. Basically we can distinguish between global and local path planning methods, which account for the two subproblems. In some approaches the configuration space of the vehicle is sampled and a graph-based description is used to grasp the topological structure of the environment [6]. For example, the Probabilistic Roadmap Method [7] uses general graphs with an offline sampling; the Rapidly Exploring Random Tree approach [8–10] builds one or more trees online to cover the free space. These sampling-based frameworks make use of a vehicle-specific local planner module (also denoted as a *steering method*), which accounts for connecting two arbitrary configurations of the nonholonomic car. Graph vertices represent configurations, while edges in the graphs correspond to local paths in these approaches. The global admissibility of these is checked by an appropriate collision detector module.

Other approaches are based on the Holonomic Path Approximation Algorithm [11], which recommends planning a preliminary holonomic (kinematically unconstrained) path first and approximating it using a steering method in a second step. In this framework, the holonomic global path is

iteratively subdivided and the steering method together with a collision detector is applied to replace the parts. To ensure the convergence of this algorithm, the steering method has to verify the *topologically property*, which is defined in [12].

Steering nonholonomic systems is a difficult task even in the absence of obstacles. Exact algorithms are not available in general but only for special classes, for example, nilpotentizable [13], chain-formed [14–16], and differentially flat systems [17]. Common examples for the above-mentioned system classes include wheeled vehicles with or without trailers. There exist elegant solutions of path planning and control especially for differentially driven and car-like robots [18–21] and mobile manipulators [22–24]. Most of the available work investigates differentially driven robots with one trailer [19] or more trailers [18, 20]. The steering methods presented in [19, 20] are shown to verify the topological property as well. Since the kinematics of a car is equivalent to a differentially driven robot towing a trailer hitched on top of the wheel axis, these flatness-based methods were shown to be applicable for cars [21] as well. In these approaches a sufficiently smooth planar curve is planned for the flat output of the system, which is the axle midpoint of the last trailer, or simply the rear axle midpoint in case of a single car. Although the method proposed in [21] is a simple and elegant way of generating local paths for cars with continuous curvature, it does not take into account the upper-bounded curvature constraint (except at the start and end configurations).

An important and widely used family of steering approaches is based on optimal control. For general systems only approximate methods exist [25]. However, exact methods for computing optimal (e.g., shortest length) local paths are available for cars moving only forward (Dubins-car [26]) or both forward and backward (Reeds–Shepp-car [27–30]). For the Reeds–Shepp-car, it is shown in [27] that for any pair of configurations the shortest path can be chosen from 48 possible sets of paths, each of which consisting of maximum five circular or straight segments and having maximum two cusps. The number of sets was reduced to 46 in [28] and to 26 in [30]. Because these optimal paths consist of linear and circular segments, the resulting curvature profile contains abrupt changes, which can be traversed precisely only when the car is stopped at points where the curvature changes, in order to reorient its wheels. To overcome this limitation, a steering method is introduced in [31] which approximates Reeds–Shepp paths using continuous curvature turns, which contain clothoid curves to connect linear and circular parts. It was shown that although these paths are not optimal, they converge to the optimal Reeds–Shepp paths when the allowed sharpness of the clothoids tends to infinity. Both the Reeds–Shepp planner and its continuous curvature generalization (let us denote it as CCRS planner in the sequel) is proved to be topological admissible.

Other continuous curvature planning approaches fall into the “path smoothing” class, where a sequence of configurations or a preliminary polygonal route is given and the goal is turning it to a smooth curve. These make use of clothoids [32] and higher order polynomial curves [33–35], or curves with closed form expressions such as Bézier curves [36, 37]

or *B*-splines [38]. Unfortunately these are not treating issues such as topological admissibility or convergence guarantees in general planning tasks.

Beyond these approaches, there are specialized algorithms, which were designed to solve a specific planning task, for example, parallel or garage parking. These are not suited to solve more general planning problems in case of arbitrary obstacle distributions. For example, [39, 40] use circle segments and straight lines to generate trajectories for car parking. In [40] these are denoted as translation (TM) and rotation movements (RM), and a number of TM–RM–TM combinations are linked together, depending on the maneuvering complexity of the parking task. An exhaustive search is performed in every step; thus the computational complexity increases heavily with the number of necessary maneuvers. The method of Müller et al. [41] proposes solving the parking task by the general two-step path approximation approach, where they use the CCRS local planner [31] and the global planner described in [42]. Although a general framework is applied, they exploit some simplifications and assumptions (e.g., regarding the shape of obstacles and symmetry of their distribution) in order to make computations tractable for real-time applications.

A recent publication [43] introduces a planning approach which generalizes a parking planner to handle more difficult obstacle distributions, for example, navigation in a narrow parking lot containing some additional obstacles. It is shown that the algorithm can be applied in real time thanks to the fast optimization strategy it is based on. However, completeness or guaranteed convergence is not proven.

3. The RTR + TTS-Planner

Many of the mentioned planning methods were seminal for the development of our approach presented in this paper. Our goal was to design an algorithm with the following properties:

- (i) It should solve the general planning task; that is, no assumptions are made regarding the obstacle distribution or the starting and goal configurations of the vehicle. At the same time, it should perform well also in the specialized parking situations.
- (ii) The resulting paths should have continuous curvature profile.
- (iii) The aim is to generate “human-like” paths, which are simple and contain a “meaningful” number of maneuvers needed by the task at hand.

We chose the following methodology during the design of this planning method. A two-step approximation approach similar to [11, 41] was chosen; however, our global planner already takes some nonholonomic properties of the vehicle into account. The main contributions of this paper are the introduction of our novel topological admissible continuous curvature local planning method (TTS-planner) which facilitates the construction of “human-like” paths, as well as the detailed description of the global RTR-planner.

This paper extends our previous work [45] giving a thorough theoretical treatment, detailed derivation, and proofs

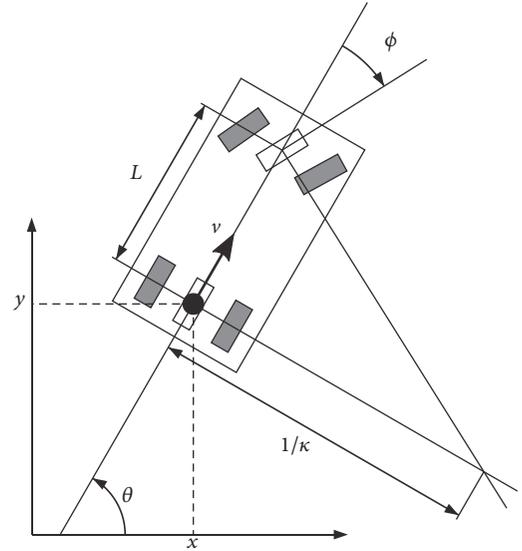


FIGURE 1: Kinematic model of a car.

regarding the local planning method, as well as a more detailed description of the global planner.

Previous versions of the proposed two-step planning approach can be found in [46–49], where the local planner used only straight and circular segments, resulting in a discontinuous curvature profile. The currently proposed generalized version uses clothoids as well (similarly to [31]). In the derivation we rely on the results of Wilde [50] regarding the parametrization of clothoids. Our approach is mostly comparable to the CCRS planner [31] or the two-step planning method [41] based on it, because they have similar theoretical properties. However, the simulation examples in Section 6 show that our local planner generates better quality paths and has a higher success rate in narrow environments than a CCRS-based approach.

In the sequel we introduce the local TTS-planner first, by deriving it from the kinematic properties of the car and choosing path primitives with clothoids as building blocks. The global RTR-planner is introduced after that where we explain the reason of using only straight movement and in-place turning primitives in the preliminary planning phase.

4. Local Planning: The TTS-Planner

Before going into details of the local path planning method, we give mathematical definitions of the solvable problem. First the vehicle model is introduced, followed by basic properties of the feasible paths, which will serve as a basis for choosing the geometric building elements of the local planner module.

4.1. Vehicle Model. The geometrical model of a car can be seen in Figure 1. In this approach we use the one-track (or bicycle) motion model of a continuous steering kinematic car [41]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \frac{1}{L} \tan \phi \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \Omega, \quad (1)$$

where x and y are the position coordinates of the back axle midpoint (treated as the reference point of the model), θ is the orientation of the car, ϕ denotes the steering angle, and L is the distance of the front and rear wheel axles. The signed velocity of the reference point is v (positive means forward motion), and Ω stands for the steering rate. The absolute values of the steering angle and its rate of change are limited to

$$\begin{aligned} |\phi| &\leq \phi_{\max} < \frac{\pi}{2} \\ |\Omega| &\leq \Omega_{\max} < \infty. \end{aligned} \quad (2)$$

The instantaneous turning radius (the reciprocal of the path curvature κ) is given by

$$\frac{1}{\kappa} = \frac{L}{\tan \phi}. \quad (3)$$

The steering angle limit implies an upper bound on the path curvature

$$|\kappa| \leq \kappa_{\max} = \frac{\tan \phi_{\max}}{L}. \quad (4)$$

The more widespread form of the motion equation is the following [51]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\kappa} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \kappa \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \sigma, \quad (5)$$

where σ denotes the curvature change rate. The vector $q = (x, y, \theta, \kappa)$ is called the configuration (or state) of the vehicle and $u = (v, \sigma)$ is the input vector of the system. The connection between (1) and (5) is given by

$$\begin{aligned} \kappa &= \frac{\tan \phi}{L} \\ \sigma &= \dot{\kappa} = \frac{\dot{\phi}}{L \cos^2 \phi} = \frac{\Omega}{L \cos^2 \phi}. \end{aligned} \quad (6)$$

An upper bound on the curvature change rate can be expressed as

$$|\sigma| \leq \sigma_{\max} = \frac{\Omega_{\max}}{L} \leq \frac{\Omega_{\max}}{L \cos^2 \phi}. \quad (7)$$

4.2. Feasible Paths. To elaborate the properties of feasible local paths of our car model, let the translational velocity of the car be written as $v = \gamma|v|$, where $|v|$ stands for the

magnitude of the velocity and γ represents the local direction of motion:

$$\gamma = \begin{cases} 1, & \text{if } v \geq 0 \text{ (forward motion),} \\ -1, & \text{if } v < 0 \text{ (backward motion).} \end{cases} \quad (8)$$

Furthermore, let us introduce the translational displacement of the car along its path

$$s(t) = \int_0^t |v(\tau)| d\tau \quad (9)$$

and express the translational velocity as

$$v = \gamma \frac{ds}{dt}. \quad (10)$$

Using (10) and the chain rule of differentiation, we reformulate the motion equation (5) as follows:

$$\frac{d}{ds} \begin{bmatrix} x \\ y \\ \theta \\ \kappa \end{bmatrix} \cdot \frac{ds}{dt} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \kappa \\ 0 \end{bmatrix} \gamma \frac{ds}{dt} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \sigma. \quad (11)$$

After dividing by (10) and defining

$$\alpha(s) = \frac{\sigma}{v} = \gamma \frac{d\kappa}{ds}(s) = \frac{d^2\theta}{ds^2}(s), \quad (12)$$

the *sharpness* of the path, we get the spatial description of feasible paths:

$$\gamma \frac{d}{ds} \begin{bmatrix} x \\ y \\ \theta \\ \kappa \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \kappa \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \alpha. \quad (13)$$

We can establish the following properties of these paths for our car model:

- (i) The orientation θ at s is constrained by the local displacements along the x - and y -axes and by the motion direction γ . The path curvature κ is given by the first (spatial) derivative of θ and the motion direction. Thus every feasible path can be fully described by a planar curve labeled by the motion direction at every point. This fact actually arises from the differential flatness property of the car with the rear axle midpoint position as flat output [17, 21].
- (ii) The sharpness of the path is the second derivative of θ (the spatial change rate of the curvature along the path) and determines the maximal allowed velocity at every point on the path:

$$|v(s)| \leq \frac{\sigma_{\max}}{|\alpha(s)|}. \quad (14)$$

If a nonzero minimal traveling speed v_{\min} is required, then the sharpness has an upper bound:

$$|\alpha(s)| \leq \alpha_{\max} = \frac{\sigma_{\max}}{|v_{\min}|}. \quad (15)$$

In other words, if we would like to obtain a path which can be followed by the continuous steering car (1) and we allow the car to stop only if the motion direction is reversed, then the path should be a planar curve with

- (i) bounded curvature and
- (ii) bounded sharpness (continuous curvature).

In Sections 4.3, 4.4, and 4.5 we are about to present specific paths fulfilling these requirements and the local path planner for generating such paths.

4.3. Path Elements. Our aim is to choose geometric primitives which can serve as building elements in the process of path construction. We assume that the motion direction is the same along a primitive, but it can change between the primitives (which means a cusp in the path). The most simple planar curves with bounded curvature are straight lines ($\kappa = 0$, called *S* segments in the sequel) and curvature-limited circular arcs ($|\kappa(s)| = \text{const} \leq \kappa_{\max}$, called *C* segments). Paths constructed by these primitives have piecewise constant curvature with abrupt curvature changes (i.e., infinite sharpness) between the pieces. To obtain continuous curvature, we are about to use *clothoids*, which are the most simple parametric curves with changing curvature and bounded sharpness.

4.3.1. Clothoids. A clothoid is a curve with constant sharpness

$$|\alpha(s)| = |\alpha| \leq \alpha_{\max} < \infty, \quad \forall s. \quad (16)$$

According to (13), the equations describing a clothoid path can be obtained by integration. If we assume that $q(0) = 0$, we get

$$x(s) = \gamma \sqrt{\frac{\pi}{|\alpha|}} C_F \left(\sqrt{\frac{|\alpha|}{\pi}} s \right), \quad (17a)$$

$$y(s) = \gamma \text{sgn}(\alpha) \sqrt{\frac{\pi}{|\alpha|}} S_F \left(\sqrt{\frac{|\alpha|}{\pi}} s \right), \quad (17b)$$

where $\text{sgn}(\alpha)$ denotes the sign of α and C_F and S_F stand for the Fresnel cosine and sine integrals:

$$C_F(r) = \int_0^r \cos\left(\frac{\pi}{2}\mu^2\right) d\mu, \quad (18a)$$

$$S_F(r) = \int_0^r \sin\left(\frac{\pi}{2}\mu^2\right) d\mu. \quad (18b)$$

4.3.2. Clothoid Types. From the point of view of path planning, eight different types of clothoids can be distinguished, based on the following properties:

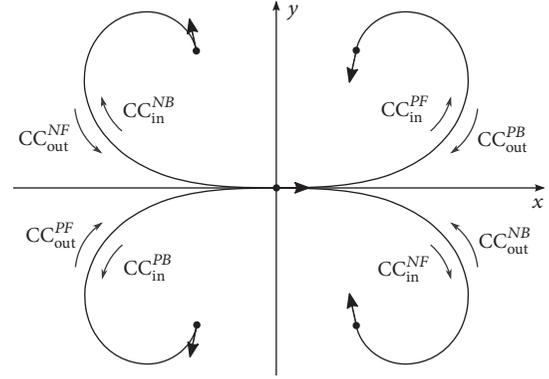


FIGURE 2: Clothoid types.

- (i) Sign of the sharpness: $\text{sgn}(\alpha) = \gamma \text{sgn}(d\kappa/ds)$, it can be positive (*P*, $\text{sgn}(\alpha) = 1$) or negative (*N*, $\text{sgn}(\alpha) = -1$). Zero sharpness is omitted because it results in a straight or circular segment instead a clothoid.
- (ii) Direction of motion: γ , it can be forward (*F*) or backward (*B*)
- (iii) Sign of the absolute curvature change: $\text{sgn}(d|\kappa|/ds)$. The absolute curvature grows if the vehicle is entering a turn (called *in*-type clothoid) and vanishes if it returns to a straight segment (*out*-type clothoid).

Equations (17a) and (17b) describe in-type clothoids parametrized by the constants α , γ and the running variable $s \in [0, s_{CC}]$, starting from $q_{\text{in}}(0) = 0$ and arriving at some $q_{\text{in}}(s_{CC})$, where s_{CC} denotes the whole curve length. From every in-type clothoid we can derive an out-type counterpart by preserving the same geometry but reversing the motion direction:

$$x_{\text{out}}(s) = -\gamma \sqrt{\frac{\pi}{|\alpha|}} C_F \left(\sqrt{\frac{|\alpha|}{\pi}} (s_{CC} - s) \right), \quad (19a)$$

$$y_{\text{out}}(s) = -\gamma \text{sgn}(\alpha) \sqrt{\frac{\pi}{|\alpha|}} S_F \left(\sqrt{\frac{|\alpha|}{\pi}} (s_{CC} - s) \right). \quad (19b)$$

Note that these out-type clothoids start from some $q_{\text{out}}(0) = q_{\text{in}}(s_{CC})$ and arrive at $q_{\text{out}}(s_{CC}) = 0$. The eight different types of clothoids are illustrated in Figure 2.

Note that we deal here only with such clothoid path segments which have either their starting or their final configuration at the origin with zero orientation and curvature ($q(0) = 0$ for in-type and $q(s_{CC}) = 0$ for out-type clothoids). However, this causes no serious limitations. On the one hand, any $q = (x, y, \theta, 0)$ can be transformed to $q' = (0, 0, 0, 0)$ by translation and rotation. On the other hand, $\kappa(0) = 0$ or $\kappa(s_{CC}) = 0$ means that our final concatenated paths can consist of *S* segments, $CC_{\text{in}}-C-CC_{\text{out}}$ triplets, and $CC_{\text{in}}-CC_{\text{out}}$ pairs. Practically, the only limitation is that transitions between two *C* segments of different curvatures are allowed only with a curvature profile crossing or touching zero, that is, by a $C-CC_{\text{out}}-CC_{\text{in}}-C$ sequence. As we will see later, this does not prohibit the creation of a complete path planner.

From now on, according to the nomenclature of [31], we will denote $CC_{in}-C-CC_{out}$ triplets as *CC-turns* or *T segments*, and $CC_{in}-CC_{out}$ pairs as *elementary paths* or *E segments*. Note that an elementary path is a special case of a CC-turn.

4.3.3. Reparametrization of Clothoids. By looking at (17a) and (17b), we can state that any point on an in-type clothoid starting from $q(0) = 0$ can be described by the pair $(\alpha, \gamma s)$, the sharpness and the signed displacement along the curve. It was shown by Wilde [50] that an equivalent representation is available using the pair (κ, δ) , the curvature and the deflection of the curve at a certain point:

$$x(\kappa, \delta) = \text{sgn}(\delta) \frac{\sqrt{2\pi|\delta|}}{\kappa} C_F \left(\sqrt{\frac{2|\delta|}{\pi}} \right), \quad (20a)$$

$$y(\kappa, \delta) = \frac{\sqrt{2\pi|\delta|}}{\kappa} S_F \left(\sqrt{\frac{2|\delta|}{\pi}} \right). \quad (20b)$$

As it can be seen, κ appears only as a multiplier in the expressions; thus it can be interpreted as a scaling factor. In this representation δ is responsible for the “shape” of the clothoid segment, and κ determines its “size.”

From a computational perspective, this (κ, δ) parametrization is less convenient than the parametrization by $(\alpha, \gamma s)$. The reason is that α and γ are constant values and only s is variable along the path segment; thus the curve is described by some fixed parameters and exactly one free parameter, as usual. This makes the explicit computation of these curves simple, for example, for visualization or collision checking. Opposed to this, both κ and δ are varying along the curve, and coherency has to be maintained between them to obtain proper results. The coherency is ensured by the constant sharpness, which is related to κ and δ as

$$\alpha = \frac{\kappa^2}{2\delta}. \quad (21)$$

For simplicity, let us choose fixed curvature and deflection values κ_{CC} and δ_{CC} at the endpoint of curve. This is the point with maximal absolute curvature and deflection:

$$\kappa_{CC} = \arg \max_{\kappa} |\kappa|, \quad (22a)$$

$$\delta_{CC} = \arg \max_{\delta} |\delta|. \quad (22b)$$

Based on (20a), (20b), (22a), and (22b), in-type clothoids can be fully described by

$$x_{in}(\delta) = \text{sgn}(\delta_{CC}) \frac{\sqrt{2\pi|\delta_{CC}|}}{\kappa_{CC}} C_F \left(\sqrt{\frac{2|\delta|}{\pi}} \right), \quad (23a)$$

$$y_{in}(\delta) = \frac{\sqrt{2\pi|\delta_{CC}|}}{\kappa_{CC}} S_F \left(\sqrt{\frac{2|\delta|}{\pi}} \right), \quad (23b)$$

$$\theta_{in}(\delta) = \delta, \quad (23c)$$

$$\kappa_{in}(\delta) = \kappa_{CC} \sqrt{\frac{\delta}{\delta_{CC}}}, \quad (23d)$$

where $\delta \in [0, \delta_{CC}]$ is the variable parameter and δ_{CC} and κ_{CC} are constant values. A shorter form for $x_{in}(\delta)$ and $y_{in}(\delta)$ can be obtained by defining the following functions:

$$X(\beta) = \text{sgn}(\beta) \sqrt{\pi|\beta|} \cdot C_F \left(\sqrt{\frac{|\beta|}{\pi}} \right) \quad (24)$$

$$Y(\beta) = \sqrt{\pi|\beta|} \cdot S_F \left(\sqrt{\frac{|\beta|}{\pi}} \right). \quad (25)$$

Using these, (23a), (23b), (23c), and (23d) will look like

$$x_{in}(\delta) = \frac{X(2\delta)}{\kappa_{CC}} \sqrt{\frac{\delta_{CC}}{\delta}} = \frac{X(2\delta)}{\kappa_{in}(\delta)}, \quad (26a)$$

$$y_{in}(\delta) = \frac{Y(2\delta)}{\kappa_{CC}} \sqrt{\frac{\delta_{CC}}{\delta}} = \frac{Y(2\delta)}{\kappa_{in}(\delta)}, \quad (26b)$$

$$\theta_{in}(\delta) = \delta, \quad (26c)$$

$$\kappa_{in}(\delta) = \kappa_{CC} \sqrt{\frac{\delta}{\delta_{CC}}}. \quad (26d)$$

Note that if we look at the endpoint, $X(2\delta_{CC})$ and $Y(2\delta_{CC})$ can be treated as the end position of an “unscaled” in-type clothoid of δ_{CC} full deflection. The “scaling factor” $1/\kappa_{CC}$ can be used to obtain the real endpoint coordinates $x_{in}(\delta_{CC})$ and $y_{in}(\delta_{CC})$.

Now, let us consider a pair of in-type and out-type clothoids having the same geometry (such pairs were shown in Figure 2, e.g., CC_{in}^{PF} and CC_{out}^{PB}). In order to ensure the above-mentioned coherency between κ_{CC} and δ_{CC} , we define them according to (22a) and (22b) for out-type clothoids as well. However, in this case κ_{CC} belongs to the starting point, while δ_{CC} to the endpoint of the curve. Since an out-type clothoid has the same geometry as its in-type pair but opposite motion direction, we can obtain its equation as

$$x_{out}(\delta) = -\text{sgn}(\delta_{CC}) \frac{\sqrt{2\pi|\delta_{CC}|}}{\kappa_{CC}} C_F \left(\sqrt{\frac{2|\delta - \delta_{CC}|}{\pi}} \right), \quad (27a)$$

$$y_{out}(\delta) = \frac{\sqrt{2\pi|\delta_{CC}|}}{\kappa_{CC}} S_F \left(\sqrt{\frac{2|\delta - \delta_{CC}|}{\pi}} \right), \quad (27b)$$

$$\theta_{out}(\delta) = \delta - \delta_{CC}, \quad (27c)$$

$$\kappa_{out}(\delta) = \kappa_{CC} \sqrt{\frac{\delta_{CC} - \delta}{\delta_{CC}}}. \quad (27d)$$

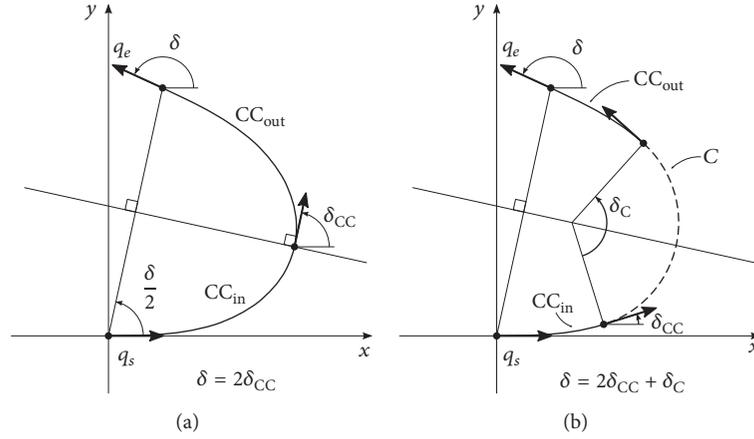


FIGURE 3: Structure of (a) elementary paths and (b) CC-turns.

4.3.4. *Elementary Paths and CC-Turns.* As already mentioned, elementary paths (E segments) are symmetric curves, consisting of a pair of compatible CC_{in} and CC_{out} clothoids. The curvature of the first curve changes from 0 to κ_{CC} , while the second from κ_{CC} back to 0. Both curves have the same deflection δ_{CC} . Compatibility is assured by the same motion direction γ and the same curvature κ_{CC} at the end of the in-type part and at the beginning of the out-type part. The compatible pairs are

- (i) CC_{in}^{PF} and CC_{out}^{NF} ,
- (ii) CC_{in}^{NF} and CC_{out}^{PF} ,
- (iii) CC_{in}^{PB} and CC_{out}^{NB} ,
- (iv) CC_{in}^{NB} and CC_{out}^{PB} .

A CC-turn consists of a pair of compatible CC_{in} and CC_{out} curves, with an additional circular arc between them, which has $\kappa_C = \kappa_{CC}$ curvature and δ_C deflection. As mentioned before, an elementary path is a special case of a CC-turn with $\delta_C = 0$ (see Figure 3).

Let us characterize these path components. They can be described by a parametric configuration curve $q(\delta) = (x(\delta), y(\delta), \theta(\delta), \kappa(\delta))$ with $\delta \in [0, \delta_{end}]$. The curve is fully determined by the following parameters:

- (i) A starting configuration $q(0) = q_s = (x_s, y_s, \theta_s, 0)$.
- (ii) The maximal curvature κ_{CC} (in an absolute sense: $|\kappa_{CC}| \geq |\kappa(\delta)|, \forall \delta$).
- (iii) Deflections of the clothoid (δ_{CC}) and circular (δ_C) parts.

We use the following building elements to synthesize elementary paths and CC-turns:

- (i) An in-type clothoid $q_{in}(\delta)$ of deflection δ_{CC} , starting from $q_{in}(0) = (0, 0, 0, 0)$ and arriving at $q_{in}(\delta_{CC}) = (x_{in}(\delta_{CC}), y_{in}(\delta_{CC}), \delta_{CC}, \kappa_{CC})$, as described by (23a), (23b), (23c), and (23d).

- (ii) A circular arc $q_c(\delta)$ of deflection δ_C and curvature κ_{CC} , starting from $q_c(0) = (0, 0, 0, 0)$, described by

$$x_c(\delta) = \frac{\sin \delta}{\kappa_{CC}} \quad (28a)$$

$$y_c(\delta) = \frac{1 - \cos \delta}{\kappa_{CC}} \quad (28b)$$

$$\theta_c(\delta) = \delta \quad (28c)$$

$$\kappa_c(\delta) = \kappa_{CC}. \quad (28d)$$

- (iii) A compatible out-type clothoid $q_{out}(\delta)$ of deflection δ_{CC} , starting from $q_{out}(0) = (x_{out}(0), y_{out}(0), -\delta_{CC}, \kappa_{CC})$ and arriving at $q_{out}(\delta_{CC}) = (0, 0, 0, 0)$, as described by (27a), (27b), (27c), and (27d).

The starting and end configurations of these parts have to be aligned in order to obtain the resulting elementary path or CC-turn. The alignment is performed by rotation and translation, described by homogeneous transformation matrices. Let us assign a transformation matrix to any configuration q_0 as follows:

$$q_0 = \begin{pmatrix} x_0 \\ y_0 \\ \theta_0 \\ \kappa_0 \end{pmatrix} \rightarrow T(q_0) = \begin{bmatrix} \cos \theta_0 & -\sin \theta_0 & 0 & 0 & x_0 \\ \sin \theta_0 & \cos \theta_0 & 0 & 0 & y_0 \\ 0 & 0 & 1 & 0 & \theta_0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (29)$$

This transformation matrix represents the pose of the local coordinate frame attached to q_0 . The transformation of

another configuration q by $T(q_0)$ can be performed as usual:

$$\begin{pmatrix} q' \\ 1 \end{pmatrix} = T(q_0) \begin{pmatrix} q \\ 1 \end{pmatrix}. \quad (30)$$

This means that we interpret the coordinates of q as local, relative to q_0 . Multiplying by $T(q_0)$ we get q' , which represents the global coordinates of the transformed q . Notice that the transformation does not affect the curvature part of the configuration. It follows that the original configuration q_0 can be obtained by applying the transformation $T(q_0)$ to $q = (0, 0, 0, \kappa_0)$:

$$\begin{pmatrix} q_0 \\ 1 \end{pmatrix} = T(q_0) \begin{pmatrix} 0 \\ 0 \\ 0 \\ \kappa_0 \\ 1 \end{pmatrix}. \quad (31)$$

The inverse transformation is given by

$$T^{-1}(q_0) = \begin{bmatrix} \cos \theta_0 & \sin \theta_0 & 0 & 0 & -x_0 \cos \theta_0 - y_0 \sin \theta_0 \\ -\sin \theta_0 & \cos \theta_0 & 0 & 0 & x_0 \sin \theta_0 - y_0 \cos \theta_0 \\ 0 & 0 & 1 & 0 & -\theta_0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (32)$$

These transformation matrices have the following chaining property:

$${}^H q' = T(q_0) \cdot {}^H q \iff T(q') = T(q_0) \cdot T(q), \quad (33)$$

where the upper “ H ” index stands for the homogenous coordinate representation ${}^H q = [q, 1]^T$.

Now, we have all the tools to assemble CC-turns and elementary paths. A general description of these is obtained as follows:

$${}^H q(\delta) = T(q_s) \cdot {}^H q_{in}(\delta), \quad \text{if } \delta \in [0, \delta_{CC}], \quad (34a)$$

$${}^H q(\delta) = T(q(\delta_{CC})) \cdot {}^H q_c(\delta - \delta_{CC}), \quad (34b)$$

$$\text{if } \delta \in (\delta_{CC}, \delta_{CC} + \delta_C].$$

$${}^H q(\delta) = T(q(\delta_{CC} + \delta_C)) \cdot T^{-1}(q_{out}(0)) \cdot {}^H q_{out}(\delta - \delta_{CC} - \delta_C), \quad (34c)$$

$$\text{if } \delta \in (\delta_{CC} + \delta_C, 2\delta_{CC} + \delta_C],$$

where q_s stands for a given arbitrary starting configuration with zero curvature and $q_{in}(\delta)$ and $q_{out}(\delta)$ are given by (23a), (23b), (23c), (23d), (27a), (27b), (27c), and (27d), respectively. This can be treated as a “direct” approach for generating a CC-turn or elementary path. One specifies the

starting configuration q_s , the maximal curvature κ_{CC} , and the deflections δ_{CC} and δ_C , and the end configuration $q_e = q(2\delta_{CC} + \delta_C)$ is obtained directly using (34a), (34b), and (34c). Without loss of generality we can assume that $q_s = (0, 0, 0, 0)$. In this case we get

$${}^H q(\delta_{CC}) = {}^H q_{in}(\delta_{CC}) = T(q_{in}(\delta_{CC})) \cdot {}^H 0, \quad (35a)$$

$${}^H q(\delta_{CC} + \delta_C) = T(q_{in}(\delta_{CC})) \cdot {}^H q_c(\delta_C) = \underbrace{T(q_{in}(\delta_{CC})) \cdot T(q_c(\delta_C))}_{T(q(\delta_{CC} + \delta_C))} \cdot {}^H 0, \quad (35b)$$

$${}^H q_e = {}^H q(2\delta_{CC} + \delta_C) = \underbrace{T(q(\delta_{CC} + \delta_C)) \cdot T^{-1}(q_{out}(0))}_{T(q_e)} \cdot {}^H 0, \quad (35c)$$

$$\cdot {}^H q_{out}(\delta_{CC}) = T(q_e) \cdot {}^H 0,$$

because the endpoint of a standard out-type clothoid is the origin, according to (27a), (27b), (27c), and (27d). The components of

$$T(q_e) = T(q_{in}(\delta_{CC})) \cdot T(q_c(\delta_C)) \cdot T^{-1}(q_{out}(0)) \quad (36)$$

have the following form:

$$T(q_{in}(\delta_{CC})) = \begin{bmatrix} \cos \delta_{CC} & \sin \delta_{CC} & 0 & 0 & \frac{X(2\delta_{CC})}{\kappa_{CC}} \\ \sin \delta_{CC} & \cos \delta_{CC} & 0 & 0 & \frac{Y(2\delta_{CC})}{\kappa_{CC}} \\ 0 & 0 & 1 & 0 & \delta_{CC} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (37)$$

$$T(q_c(\delta_C)) = \begin{bmatrix} \cos \delta_C & -\sin \delta_C & 0 & 0 & \frac{\sin \delta_C}{\kappa_{CC}} \\ \sin \delta_C & \cos \delta_C & 0 & 0 & \frac{1 - \cos \delta_C}{\kappa_{CC}} \\ 0 & 0 & 1 & 0 & \delta_C \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

In order to obtain $T^{-1}(q_{out}(0))$, we express $q_{out}(0)$ first:

$$q_{out}(0) = \begin{pmatrix} -x_{in}(\delta_{CC}) \\ y_{in}(\delta_{CC}) \\ -\delta_{CC} \\ \kappa_{CC} \end{pmatrix} = \begin{pmatrix} -\frac{X(2\delta_{CC})}{\kappa_{CC}} \\ \frac{Y(2\delta_{CC})}{\kappa_{CC}} \\ -\delta_{CC} \\ \kappa_{CC} \end{pmatrix}, \quad (38)$$

and it follows from (32) that

$$T^{-1}(q_{\text{out}}(0)) = \begin{bmatrix} \cos \delta_{CC} & -\sin \delta_{CC} & 0 & 0 & t_{1,5} \\ \sin \delta_{CC} & \cos \delta_{CC} & 0 & 0 & t_{2,5} \\ 0 & 0 & 1 & 0 & \delta_{CC} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (39)$$

where

$$\begin{aligned} t_{1,5} &= \frac{X(2\delta_{CC})}{\kappa_{CC}} \cos \delta_{CC} + \frac{Y(2\delta_{CC})}{\kappa_{CC}} \sin \delta_{CC}, \\ t_{2,5} &= \frac{X(2\delta_{CC})}{\kappa_{CC}} \sin \delta_{CC} - \frac{Y(2\delta_{CC})}{\kappa_{CC}} \cos \delta_{CC}. \end{aligned} \quad (40)$$

After multiplying the three matrices we get

$$T(q_e) = \begin{bmatrix} \cos(\delta_{\text{end}}) & -\sin(\delta_{\text{end}}) & 0 & 0 & \frac{A(2\delta_{CC}, \delta_C)}{\kappa_{CC}} \\ \sin(\delta_{\text{end}}) & \cos(\delta_{\text{end}}) & 0 & 0 & \frac{B(2\delta_{CC}, \delta_C)}{\kappa_{CC}} \\ 0 & 0 & 1 & 0 & \delta_{\text{end}} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (41)$$

where

$$\delta_{\text{end}} = 2\delta_{CC} + \delta_C, \quad (42)$$

$$\begin{aligned} A(2\delta_{CC}, \delta_C) &= X(2\delta_{CC})(1 + \cos(2\delta_{CC} + \delta_C)) \\ &\quad + Y(2\delta_{CC}) \sin(2\delta_{CC} + \delta_C) \\ &\quad + \sin(\delta_{CC} + \delta_C) - \sin \delta_{CC}, \end{aligned} \quad (43)$$

$$\begin{aligned} B(2\delta_{CC}, \delta_C) &= X(2\delta_{CC}) \sin(2\delta_{CC} + \delta_C) \\ &\quad + Y(2\delta_{CC})(1 - \cos(2\delta_{CC} + \delta_C)) \\ &\quad - \cos(\delta_{CC} + \delta_C) + \cos \delta_{CC}. \end{aligned} \quad (44)$$

The end configuration of the CC-turn is given by the elements in the last column of $T(q_e)$:

$$x_e = \frac{A(2\delta_{CC}, \delta_C)}{\kappa_{CC}}, \quad (45a)$$

$$y_e = \frac{B(2\delta_{CC}, \delta_C)}{\kappa_{CC}}, \quad (45b)$$

$$\theta_e = 2\delta_{CC} + \delta_C \quad (45c)$$

$$\kappa_e = 0 \quad (45d)$$

It is important to note that, in order to fully define a CC-turn, two of the deflections δ_{CC} , δ_C , and δ_{end} should be

given, together with the maximal curvature κ_{CC} . The third deflection parameter is determined by (42).

The equations of an elementary path are similar to (45a), (45b), (45c), and (45d); we only need to substitute $\delta_C = 0$:

$$x_e = \frac{A(2\delta_{CC}, 0)}{\kappa_{CC}} = \frac{A(2\delta_{CC})}{\kappa_{CC}}, \quad (46a)$$

$$y_e = \frac{B(2\delta_{CC}, 0)}{\kappa_{CC}} = \frac{B(2\delta_{CC})}{\kappa_{CC}}, \quad (46b)$$

$$\theta_e = 2\delta_{CC}, \quad (46c)$$

$$\kappa_e = 0, \quad (46d)$$

where

$$\begin{aligned} A(2\delta_{CC}) &= X(2\delta_{CC})(1 + \cos(2\delta_{CC})) \\ &\quad + Y(2\delta_{CC}) \sin(2\delta_{CC}), \end{aligned} \quad (47)$$

$$\begin{aligned} B(2\delta_{CC}) &= X(2\delta_{CC}) \sin(2\delta_{CC}) \\ &\quad + Y(2\delta_{CC})(1 - \cos(2\delta_{CC})). \end{aligned}$$

4.4. TTS Paths for Local Planning. Algorithms that generate feasible local paths for a wheeled system—such as our car model (5)—in the absence of obstacles are called local planners or steering methods. The exact definition of these is the following.

Definition 1 (steering method). Let us denote by Λ_f the set of all feasible local paths λ_f regarding system equation $\dot{q} = f(q, u)$. A steering method (or local planner module) is a function

$$\text{Steer} : \mathcal{C} \times \mathcal{C} \longrightarrow \Lambda_f, \quad \text{i.e., } \lambda_f(\cdot) = \text{Steer}(q_0, q_1) \quad (48)$$

such that

$$\lambda_f(0) = \text{Steer}(q_0, q_1)(0) = q_0, \quad (49)$$

$$\lambda_f(S_\lambda) = \text{Steer}(q_0, q_1)(S_\lambda) = q_1.$$

As already mentioned in Section 2, the most widely used local planners for the simple kinematic car model (without the requirement of curvature continuity) are based on the optimal length Reeds–Shepp (RS) paths [27]. These local paths consist of maximum five circular or straight segments and have at most two cusps, where the curvature of circular arcs is exactly $\pm\kappa_{\text{max}}$. If the continuous steering car model (5) is used, the continuous curvature generalization proposed in [31] can be applied. This approximates Reeds–Shepp paths by replacing circular segments with CC-turns. It was shown that although the resulting CCRS paths are not optimal, they converge to the optimal RS paths when the allowed sharpness of the clothoids tends to infinity. The CC-turns in this approach consist of clothoids having the maximal allowed sharpness and circle segments of maximal curvature.

In the sequel we are doing something similar: introduce a new set of local paths for the continuous steering car. The main ideas which motivated the invention of our approach instead using the CCRS paths are the following:

- (i) The main goal is global planning in the presence of obstacles. Most existing planning methods use a concatenation of local paths, which has the consequence that the result will be not optimal, even if the parts were generated by an optimal planner. Even the CCRS paths are suboptimal. Thus, having the final application in our mind, we lifted the requirement of optimality of our local paths.
- (ii) Instead minimizing the global path length, we had the goal to achieve “natural” or “human-like” paths. What this means is informally described in the following points.
- (iii) The RS (resp., CCRS) paths consist of up to five circular (resp., CC-turn) and straight segments. We show in the sequel that three path segments are sufficient to connect any two configurations.
- (iv) The RS (resp., CCRS) paths use circular segments (resp., CC-turns) having the maximal allowed curvature (the steering wheel is turned to the end position in every curve). Our planner is not restricted to this. Only the limit is taken into account, but smaller curvatures are allowed.
- (v) The last segment of RS (resp., CCRS) paths is always circular (resp., CC-turn). We argue that this contradicts the behavior of human drivers. Because the car can only move forward or backward, in most cases the more natural behavior is to arrive at a given goal configuration by a straight movement. There are a few exceptions, for example, the case of parallel parking, but these can usually be transformed to the mentioned case by reversing the roles of initial and goal configurations. Consequently, we choose a straight segment as the last section of our local paths.

To simplify notation in the sequel, we will denote local paths as a concatenation of S, T, and E letters, corresponding straight segments, CC-turns, and elementary paths, respectively. In our approach we use *TTS* (as well as *EES*) paths (note that in our previous work [45] we used the notations *T*TS* and *E*ES* where *T** and *E** represented “CC-turn or straight segment” and “elementary path or straight segment,” although we omit here the possibility of having a straight segment as the first element because our experiments showed that *STS* and *SES* paths are chosen very rarely by the implemented planning algorithm). In the sequel we present the construction of *EES* paths first (where we use only elementary paths in place of CC-turns) and show that this class of paths is sufficient to connect an arbitrary pair of configurations. After that we generalize *E* segments to CC-turns.

4.4.1. Construction of EES Paths. Let us assume without loss of generality that the path starts from a given initial configuration $q_I = (x_I, y_I, \theta_I, 0)$ and arrives at the origin as goal configuration $q_G = (0, 0, 0, 0)$. Any query pair (q_I, q_G) can be transformed to this form by translation and rotation, and the designed path can be transformed back. Let us denote the intermediate configurations between the path segments

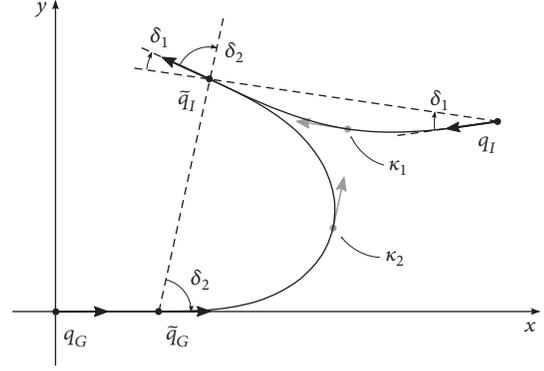


FIGURE 4: *EES* path.

by \tilde{q}_I and \tilde{q}_G . For every segment, i denotes its index (1, 2 or 3) inside the path. A straight segment is parametrized by its absolute length s_i and direction γ_i and an elementary path segment by its curvature $\kappa_{i,CC}$ and deflection $\delta_{i,CC}$ at the middle point between the CC_{in} and CC_{out} parts. To simplify the notation, let us neglect the “CC” subscript from now on: let $\kappa_i = \kappa_{i,CC}$ and $\delta_i = \delta_{i,CC}$.

The scheme of an *EES* path (illustrated in Figure 4) is the following:

$$q_I \xrightarrow[\delta_1, \kappa_1]{E} \tilde{q}_I \xrightarrow[\delta_2, \kappa_2]{E} \tilde{q}_G \xrightarrow[s_3, \gamma_3]{S} q_G. \quad (50)$$

The coordinates of \tilde{q}_I can be obtained from q_I by a transformation belonging to the first elementary path

$${}^H\tilde{q}_I = T(q_I) \cdot T(q_e(\delta_1, \kappa_1)) \cdot {}^H0, \quad (51)$$

where

$$T(q_e(\delta_1, \kappa_1)) = \begin{bmatrix} \cos(2\delta_1) & -\sin(2\delta_1) & 0 & 0 & \frac{A(2\delta_1)}{\kappa_1} \\ \sin(2\delta_1) & \cos(2\delta_1) & 0 & 0 & \frac{B(2\delta_1)}{\kappa_1} \\ 0 & 0 & 1 & 0 & 2\delta_1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (52)$$

$$T(q_I) = \begin{bmatrix} \cos(\theta_I) & -\sin(\theta_I) & 0 & 0 & x_I \\ \sin(\theta_I) & \cos(\theta_I) & 0 & 0 & y_I \\ 0 & 0 & 1 & 0 & \theta_I \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

By applying the matrix product we obtain

$$\tilde{x}_I = \frac{A(2\delta_1)}{\kappa_1} \cos(\theta_I) - \frac{B(2\delta_1)}{\kappa_1} \sin(\theta_I) + x_I \quad (53a)$$

$$\tilde{y}_I = \frac{A(2\delta_1)}{\kappa_1} \sin(\theta_I) + \frac{B(2\delta_1)}{\kappa_1} \cos(\theta_I) + y_I \quad (53b)$$

$$\tilde{\theta}_I = 2\delta_1 + \theta_I \quad (53c)$$

$$\tilde{\kappa}_I = 0. \quad (53d)$$

Let us define the auxiliary functions

$$C(\beta, \psi) = A(\beta) \cos \psi - B(\beta) \sin \psi \quad (54)$$

$$D(\beta, \psi) = A(\beta) \sin \psi + B(\beta) \cos \psi$$

and write (53a), (53b), (53c), and (53d) in a shorter form:

$$\tilde{x}_I = \frac{C(2\delta_1, \theta_I)}{\kappa_1} + x_I \quad (55a)$$

$$\tilde{y}_I = \frac{D(2\delta_1, \theta_I)}{\kappa_1} + y_I \quad (55b)$$

$$\tilde{\theta}_I = 2\delta_1 + \theta_I \quad (55c)$$

$$\tilde{\kappa}_I = 0. \quad (55d)$$

Alternatively, we can get \tilde{q}_I from $\tilde{q}_G = (\tilde{x}_G, 0, 0, 0)$ by “traveling backward” along the elementary path. This corresponds to a normal E segment with inverse deflection. We can obtain \tilde{q}_I this way by applying the transformation

$${}^H\tilde{q}_I = T(\tilde{q}_G) \cdot T(q_e(-\delta_2, \kappa_2)) \cdot {}^H0, \quad (56)$$

where $T(q_e(-\delta_2, \kappa_2))$ is the same as (41) with parameters $\delta_{CC} = -\delta_2$ and $\delta_C = 0$:

$$T(q_e(-\delta_2, \kappa_2)) = \begin{bmatrix} \cos(2\delta_2) & \sin(2\delta_2) & 0 & 0 & \frac{A(-2\delta_2)}{\kappa_2} \\ -\sin(2\delta_2) & \cos(2\delta_2) & 0 & 0 & \frac{B(-2\delta_2)}{\kappa_2} \\ 0 & 0 & 1 & 0 & -2\delta_2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (57)$$

and $T(\tilde{q}_G)$ looks like

$$T(\tilde{q}_G) = \begin{bmatrix} 1 & 0 & 0 & 0 & \tilde{x}_G \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (58)$$

The result of the transformation is

$$\tilde{x}_I = \frac{A(-2\delta_2)}{\kappa_2} + \tilde{x}_G \quad (59a)$$

$$\tilde{y}_I = \frac{B(-2\delta_2)}{\kappa_2} \quad (59b)$$

$$\tilde{\theta}_I = -2\delta_2 \quad (59c)$$

$$\tilde{\kappa}_I = 0. \quad (59d)$$

By comparing (55a), (55b), (55c), (55d), (59a), (59b), (59c), and (59d) we can conclude that the path parameters δ_1 , κ_1 , δ_2 , κ_2 , and $\tilde{x}_G = -\gamma_3 s_3$ of an EES path fulfill the following constraints:

$$\frac{C(2\delta_1, \theta_I)}{\kappa_1} + x_I = \frac{A(-2\delta_2)}{\kappa_2} - \gamma_3 s_3 \quad (60a)$$

$$\frac{D(2\delta_1, \theta_I)}{\kappa_1} + y_I = \frac{B(-2\delta_2)}{\kappa_2} \quad (60b)$$

$$2\delta_1 + \theta_I = -2\delta_2. \quad (60c)$$

One can conclude that in an EES planning problem only one path segment can be parametrized freely; the others are determined. In the sequel we will use δ_1 and κ_1 as free parameters.

Let us pay some attention to the ranges of angle parameters in the above equations. The orientation parameter θ_I is treated as being in the interval $[-\pi, \pi]/\sim$, where “/” means that the two boundary values are identified. Similarly to the geometric derivation of CC-turns, we introduce an upper bound $|\delta_{\text{end},i}| \leq \pi$ on the whole deflection of both elementary paths, which results in $\delta_i \in [-\pi/2, \pi/2]$. In order to ensure that δ_1 , δ_2 , and θ_I being in the given intervals fulfill the constraint (60c), we have to choose the set carefully from which δ_1 and δ_2 can be taken. Let us examine two cases:

(i) $\theta_I \in [0, \pi]$: choosing $\delta_1 \in [-\pi/2, \pi/2 - \theta_I/2] \subseteq [-\pi/2, \pi/2]$ results in

$$\delta_1 + \frac{\theta_I}{2} = -\delta_2 \in \left[-\frac{\pi}{2} + \frac{\theta_I}{2}, \frac{\pi}{2}\right] \subseteq \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]. \quad (61)$$

(ii) $\theta_I \in [-\pi, 0]$: choosing $\delta_1 \in [-\theta_I/2 - \pi/2, \pi/2] \subseteq [-\pi/2, \pi/2]$ results in

$$\delta_1 + \frac{\theta_I}{2} = -\delta_2 \in \left[-\frac{\pi}{2}, \frac{\pi}{2} + \frac{\theta_I}{2}\right] \subseteq \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]. \quad (62)$$

These choices ensure both δ_1 and δ_2 being in $[-\pi/2, \pi/2]$ for any values of θ_I . We can write these conditions in a single compact form:

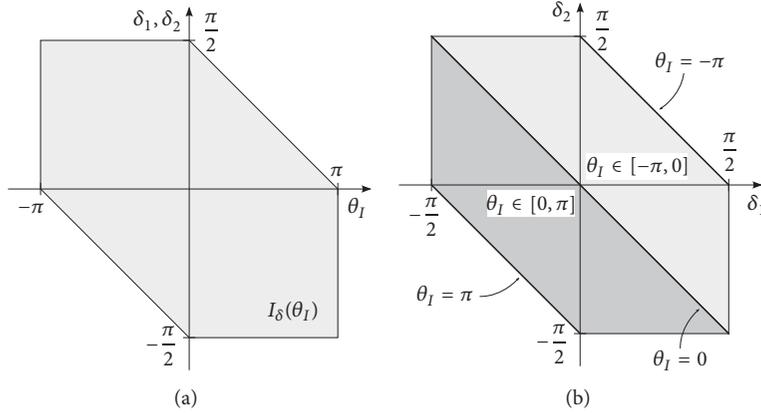
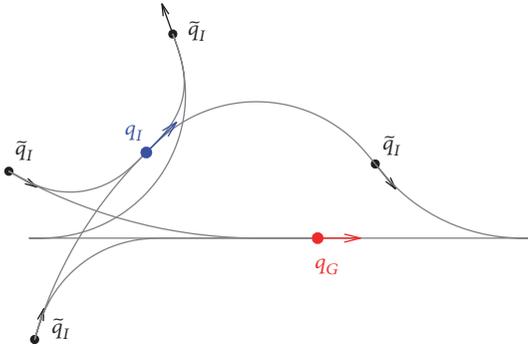
$$\delta_1, \delta_2 \in I_\delta(\theta_I) = \left[-\text{sgn}_p(\theta_I) \frac{\pi}{2}, \text{sgn}_p(\theta_I) \frac{\pi}{2} - \frac{\theta_I}{2}\right], \quad (63)$$

where $\text{sgn}_p(\cdot)$ is a special sign function with $\text{sgn}_p(0) = 1$:

$$\text{sgn}_p(x) = \begin{cases} \text{sgn}(x), & \text{if } x \neq 0 \\ 1, & \text{if } x = 0 \end{cases}. \quad (64)$$

An illustration of this set can be seen in Figure 5.

Up to now we have not taken the curvature bound into account. It can be proven that an arbitrary configuration pair can be connected by an EES path even if an upper bound on the absolute curvature is given. The following Lemma states this fact; the proof is elaborated in Appendix A.

FIGURE 5: Set of valid (δ_1, δ_2) pairs for *EES* paths.FIGURE 6: There is an infinite number of *TTS* solutions between two configurations.

Lemma 2. *In the absence of obstacles, any (q_I, q_G) pair of initial and goal configurations of a continuous steering car can be connected by an *EES* path.*

Furthermore, it turns out that infinitely many solutions can be found even with bounded curvature, due to the infinitely many choices of δ_1 and κ_1 parameters. This means that \tilde{q}_I can be chosen in infinitely many ways, as illustrated in Figure 6.

4.4.2. From *EES* to *TTS* Paths. As we have already seen in Section 4.3.4, elementary paths are special *CC*-turns with $\delta_C = 0$. It follows that the results for elementary paths used in *EES* planning can be generalized to *CC*-turns as well. The equations of an elementary path starting from any \tilde{q}_I and arriving at the x -axis—the middle segment of an *EES* path—are given by (59a), (59b), (59c), and (59d). This can easily be generalized to the case when the deflection of the circular part is nonzero:

$$\tilde{x}_I = \frac{A(-2\delta_{2,CC} - \delta_{2,C})}{\kappa_2} + \tilde{x}_G \quad (65a)$$

$$\tilde{y}_I = \frac{B(-2\delta_{2,CC} - \delta_{2,C})}{\kappa_2} \quad (65b)$$

$$\tilde{\theta}_I = -2\delta_{2,CC} - \delta_{2,C} \quad (65c)$$

$$\tilde{\kappa}_I = 0, \quad (65d)$$

where $\delta_{2,CC}$ and $\delta_{2,C}$ stand for the deflections of the clothoid and the circular parts, respectively. The full deflection of the *CC*-turn is

$$\delta_{2,end} = 2\delta_{2,CC} + \delta_{2,C} = -\tilde{\theta}_I. \quad (66)$$

Let us introduce the relative amount of the circular part of a *CC*-turn as

$$R_\delta = \frac{\delta_C}{\delta_{end}}. \quad (67)$$

Using this, we can express (65b) as

$$\tilde{y}_I = \frac{B((1 - R_{\delta,2})\delta_{2,end}, R_{\delta,2}\delta_{2,end})}{\kappa_2}. \quad (68)$$

The maximal curvature and the sharpness of the *CC*-turn are given by

$$\kappa_2 = \frac{1}{\tilde{y}_I} B((1 - R_{\delta,2})\delta_{2,end}, R_{\delta,2}\delta_{2,end}) \quad (69a)$$

$$\alpha_2 = \frac{\kappa_2^2}{2\delta_{2,CC}} = \frac{B^2((1 - R_{\delta,2})\delta_{2,end}, R_{\delta,2}\delta_{2,end})}{\tilde{y}_I^2 (1 - R_{\delta,2})\delta_{2,end}}. \quad (69b)$$

Note that if we fix the full deflection $\delta_{2,end}$ and change the ratio $R_{\delta,2}$ only, then for a given \tilde{q}_I the end configuration \tilde{q}_G on the x -axis remains the same; only κ_2 and α_2 are affected. In other words, we can tune the shape of a *CC*-turn with R_δ without affecting the starting and final configurations. This is illustrated in Figure 7, where clothoids are drawn with solid black lines and the red dashed arcs represent circular segments.

When R_δ grows, the curvature is decreasing but the sharpness is increasing at the same time, independently from the full deflection of the path segment. This property can be examined in Figure 8, which shows surface and contour plots of the (absolute) curvature and sharpness of a *CC*-turn (with

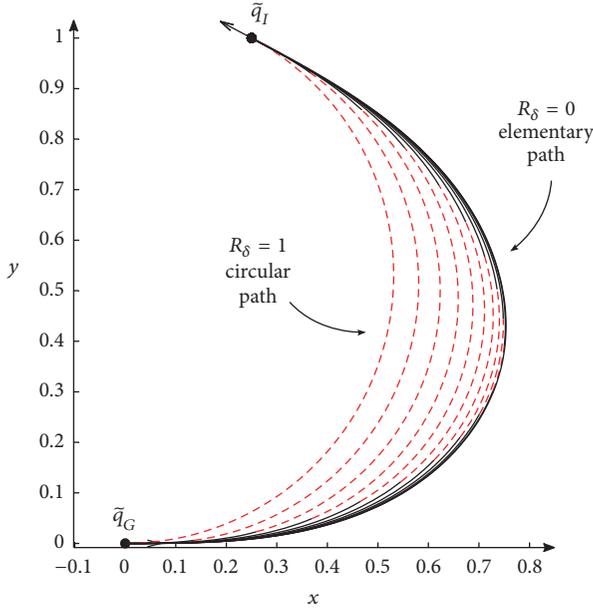


FIGURE 7: Effect of R_δ on the shape of a CC-turn (R_δ is modified in steps of 0.1).

$\tilde{y}_I = 1$), depending on the full deflection and the relative amount of the circular part.

It follows that if we already have obtained an E segment between any two configurations, we can reshape it to a CC-turn having smaller curvature and higher sharpness by tuning the R_δ parameter. With this, we can define a maximal sharpness α_{\max} beyond the already treated κ_{\max} constraint. If the sharpness of the initial E segment is less than α_{\max} , then there exists an R_δ value such that $|\alpha| = \alpha_{\max}$. This is beneficial because the length of the path segment is decreasing, as can be seen in Figure 7.

If we have an EES path, we can do this reshaping for every E segment to obtain a TTS path. However, if any of the E segments violates the α_{\max} constraint, then it is not so straightforward. Fortunately it can be shown that a TTS solution obeying the α_{\max} constraint exists in all cases. If the second segment in the EES solution violates the sharpness constraint, we can write using (59c) and (69b) and (A.3) in Appendix A:

$$\kappa'_{\max} = \sqrt{\alpha_{\max} \tilde{\theta}_I} \quad (70)$$

$$|\tilde{y}_I| \geq \frac{B(\tilde{\theta}_I)}{\kappa'_{\max}}, \quad (71)$$

where κ'_{\max} is the minimal value of the curvature upper bound ensuring that the second elementary path will have $|\alpha| \leq \alpha_{\max}$. If we would like to obtain a TTS path from an EES solution, then by recalling (55b) and (A.6) in Appendix A we can state that κ_1 can be decreased arbitrarily in order to fulfill condition (71) for the second segment and the maximal sharpness constraint for the first segment at the same time.

The only drawback of using TTS paths instead EES solutions is that the sharpness tuning requires an iterative

search of R_δ in the interval $[0, 1]$. This cannot be avoided, because R_δ is in the argument of the $B(\cdot, \cdot)$ function in (69a) and (69b), which does not have an inverse because of the contained Fresnel integrals.

4.5. A Steering Method Based on EES Paths. A TTS path planner based on the conditions in proof of Lemma 2 and their generalization in Section 4.4.2 is not a steering method itself (in the sense of Definition 1), because it delivers not only one solution for a query pair. Instead of this, it delivers a parametrized class of solutions, which contains an infinite number of EES (and TTS) paths from q_I to q_G , fulfilling the maximum curvature (and sharpness) constraint. This property is useful in global planning problems (i.e., in the presence of obstacles), because one can more likely find a collision-free path from a solution class than from the one and only solution of an exact local planner.

This does not mean that in the presence of obstacles the existence of any collision-free (local) solution would be guaranteed. For successful global planning explicit reasoning about free space connectivity is necessary. As mentioned in Section 3, we apply the framework of the Holonomic Path Approximation Algorithm [11] for this purpose. A distinct global planner delivers a preliminary geometric path, which is iteratively subdivided and approximated by an appropriate steering method. To ensure the convergence of this algorithm, the local planner has to generate such paths that get closer to the original path as the local endpoints get closer to each other. This is called the *topological property* [12] which is formulated more precisely in Appendix B.

Therefore, if a collision-free geometric path is given and we have an appropriate steering method that verifies the topological property, then a complete (completeness means that the algorithm does not fail to return a solution if the problem is solvable) approximation algorithm can be constructed (which was a footnote in the manuscript). Hence, in order to design such a complete algorithm based on TTS paths, we need a steering method which returns exactly one from the class of TTS paths and verifies the topological property.

4.5.1. Definition of $e\bar{e}S$ Paths. In the sequel we take only EES paths into account for the purpose of designing the exact steering method. This choice will make possible proving the topological property, because EES paths have closed form expressions. In contrast to that, TTS paths with a given sharpness can only be obtained by iterative deriving from EES solutions, as seen in Section 4.4.2.

More possibilities can be found to restrict the class of EES paths to exactly one specific solution for every query pair (q_I, q_G) . For example, we may choose the sign of κ_2 and the equality in condition (A.4) in Appendix A and fix δ_1 to any chosen value except 0 and $-\theta_I \pm \pi$. In this case the remaining parameter κ_1 is determined unequivocally. This would result in an EES path with $|\kappa_2| = \kappa_{\max}$ and having the intermediate orientation $\tilde{\theta}_I = \theta_I + 2\delta_1$ at the end of the first E segment. This approach has the problem that it is hard to formulate any reasonable direct rule for choosing δ_1 and the sign of κ_2 for a given query. Instead of this we

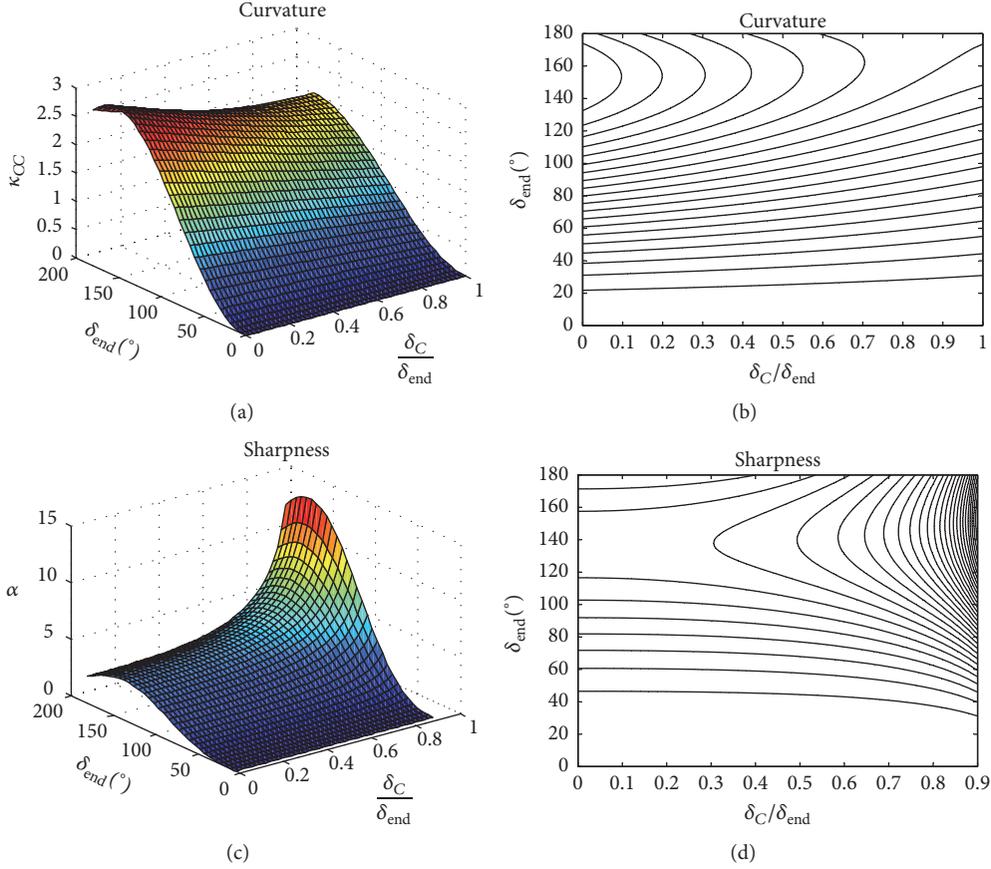


FIGURE 8: Dependence of (a)-(b) maximal curvature and (c)-(d) sharpness of CC-turns on the full detection and relative amount of the circular part (absolute values are depicted).

introduce the following approach. Let us choose $\kappa_1 = -\kappa_2$ such that $|\kappa_1| = |\kappa_2|$ is maximal. In this case we do not have to specify any other parameters. The resulting *EES* path has two *E* segments with same maximal absolute curvature, but having opposite signs (the steering wheel is turned from right to left or reversely). This type of paths is denoted as *eēS* paths in the sequel to emphasize that the resulting turning radii (reciprocal of the curvature) are opposite, equal, and minimal. The *eēS* planning problem is formulated as follows:

$$q_I \xrightarrow[\delta_1, \kappa_1]{E} \tilde{q}_I \xrightarrow[\delta_2, \kappa_2]{E} \tilde{q}_G \xrightarrow[s_3, \gamma_3]{S} q_G \quad (72)$$

such that

$$q_I = (x_I, y_I, \theta_I, 0), \quad (73)$$

$$q_G = (0, 0, 0, 0),$$

$$\kappa := \kappa_1 = -\kappa_2, \quad (74)$$

$$|\kappa| \leq \kappa_{\max}, \quad (75)$$

$$|\kappa| \rightarrow \max. \quad (76)$$

Using (74) we can reformulate (60a), (60b), and (60c) as

$$x_I + \gamma_3 s_3 + \frac{A(2\delta_1 + \theta_I) + C(2\delta_1, \theta_I)}{\kappa} = 0 \quad (77a)$$

$$y_I + \frac{G(2\delta_1, \theta_I)}{B(2\delta_1 + \theta_I) + D(2\delta_1, \theta_I)} = 0 \quad (77b)$$

$$\delta_2 = -\delta_1 - \frac{\theta_I}{2}. \quad (77c)$$

The necessary and sufficient condition for existence of this kind of *EES* paths is obtained by (75) and (77b):

$$|G(2\delta_1, \theta_I)| = |y_I| \cdot |\kappa| \leq |y_I| \cdot \kappa_{\max}. \quad (78)$$

In the sequel we use some properties of the $G(\cdot, \cdot)$ function, which are established in Appendix C.

Lemma 3. For any (y_I, θ_I) there exists δ_1 such that $|G(2\delta_1, \theta_I)| \leq |y_I| \kappa_{\max}$.

Proof. According to Properties C.6 and C.7 of the G function we can state that for all θ_I there exist δ_1 values for both $G(2\delta_1, \theta_I) \neq 0$ and $G(2\delta_1, \theta_I) = 0$. Since G is a continuous function, this means that, for all θ_I , $|G(2\delta_1, \theta_I)|$ can take an arbitrarily small value. \square

4.5.2. The *eēS* Steering Method. Based on the existence condition (78) and the optimization criterion (76) we can construct a steering method which generates a single *eēS* path for any local planning query. We denote it as the *eēS-planner*.

Let us examine the special case $y_I = 0$ first, where the existence condition (78) takes the form

$$|G(2\delta_1, \theta_I)| = 0. \quad (79)$$

According to the Proof of Property C.7 in Appendix, we can state that $|G(2\delta_1, \theta_I)|$ has a zero for all θ_I in the interval $\delta_1 \in [-\theta_I/2, 0]$. Let us denote this δ_1 solution as $\delta_{1,z}$:

$$\delta_{1,z}(\theta_I) : \left\{ \delta_1 \in \left[-\frac{\theta_I}{2}, 0 \right] \mid G(2\delta_1, \theta_I) = 0 \right\} \quad (80)$$

In this case κ could be chosen arbitrarily, but the condition (76) suggests the choice of $\kappa = \pm\kappa_{\max}$.

If $y_I \neq 0$ then it follows that

$$\kappa = -\frac{G(2\delta_1, \theta_I)}{y_I}, \quad (81)$$

and δ_1 which maximizes $|\kappa|$ is defined by

$$\delta_1^*(\theta_I) = \arg \max_{\delta_1 \in I_s(\theta_I)} \{|G(2\delta_1, \theta_I)|\}. \quad (82)$$

Let us define κ^* as

$$\kappa^*(y_I, \theta_I) = -\frac{G(2\delta_1^*(\theta_I), \theta_I)}{y_I}, \quad (83)$$

which can be treated as the unbounded optimum of κ . Since $|G|$ has a maximum at δ_1^* and zero at $\delta_{1,z}$, an arbitrary $|\kappa| \in (0, |\kappa^*|]$ can be achieved by $\delta_1 \in (\delta_{1,z}, \delta_1^*]$. It follows that if $|\kappa^*(y_I, \theta_I)| > \kappa_{\max}$, then a $\delta_{1,\max}(y_I, \theta_I)$ can be found between $\delta_1^*(\theta_I)$ and $\delta_{1,z}(\theta_I)$ such that $|\kappa| = \kappa_{\max}$:

$$\delta_{1,\max}(y_I, \theta_I) : \left\{ \delta_1 \in (\delta_{1,z}, \delta_1^*) \mid \frac{|G(2\delta_1, \theta_I)|}{|y_I|} = \kappa_{\max} \right\} \quad (84)$$

We can summarize the $e\bar{e}S$ steering method as follows. For a given q_I we have to determine δ_1 and $\kappa = \kappa_1$ first:

Condition	δ_1	κ_1
$y_I = 0$	$\delta_{1,z}(\theta_I)$	$\pm\kappa_{\max}$
$y_I \neq 0, \kappa^* \leq \kappa_{\max}$	$\delta_1^*(\theta_I)$	$\kappa^*(y_I, \theta_I)$
$y_I \neq 0, \kappa^* > \kappa_{\max}$	$\delta_{1,\max}(y_I, \theta_I)$	$\text{sgn}(\kappa^*)\kappa_{\max}$

(85)

The remaining path parameters δ_2 , κ_2 , and $\gamma_3 s_3$ can be determined using (77c), (74), and (77a), respectively.

We have proven that the above defined $e\bar{e}S$ steering method verifies the topological property which renders it suitable for application in any approximation-based planning approach. The topological property theoretically guarantees convergence of the approximation process in any cases. This advantageous fact is stated by Theorem B.2 in Appendix B and the proof is given there in detail as well.

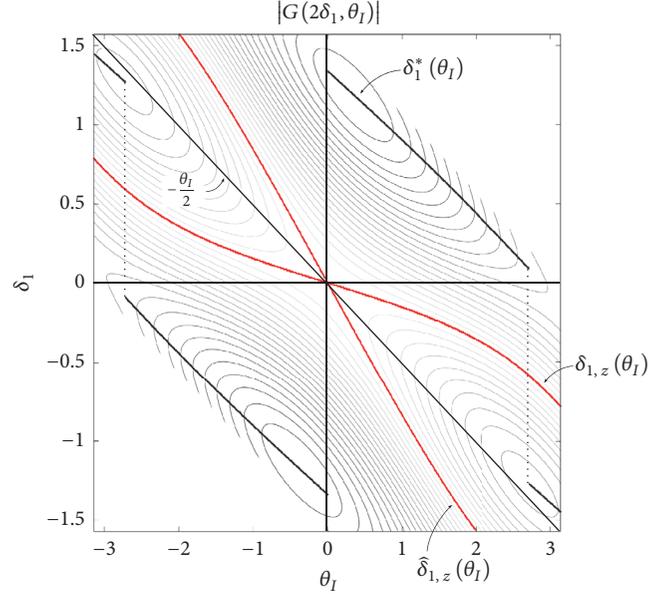


FIGURE 9: Contour plot of the $|G|$ function.

4.5.3. Implementation Issues. It turns out by numerical examination of $G(2\delta_1, \theta_I)$ that for any θ_I there is exactly one $\delta_{1,z}$ and δ_1^* value; hence $\delta_{1,\max}$ must be unique as well. Figure 9 shows a contour plot of the $|G|$ function, where the thick red lines show the zeros and the thick black lines the maxima of $|G|$. It can be seen that for every θ_I there is another zero $\hat{\delta}_{1,z}$ outside the interval $[-\theta_I/2, 0]$, but $\delta_{1,z}$ inside the interval is unique.

Unfortunately there are no closed form expressions for calculating $\delta_{1,z}$, δ_1^* , and $\delta_{1,\max}$ values, because G contains Fresnel integrals with δ_1 in the argument. Thus a numerical evaluation is needed which can be done by approximating the Fresnel integrals, for example, by using precalculated lookup tables for $X(\beta)$ and $Y(\beta)$. It is important to note that no two-dimensional lookup table for G is needed because we can decompose it to one-dimensional factors and terms of $X(\beta)$, $Y(\beta)$ and trigonometric functions.

4.5.4. Remarks on Bounded Sharpness. In the above described local planner we paid attention only to the upper bound of the curvature. One could argue that the sharpness should have an upper bound as well to achieve paths which can be traversed by a given minimal velocity. Indeed, as already seen in (15), when a minimal traveling speed is required, then the sharpness of the path should have a given upper bound. The E segments in the EES or $e\bar{e}S$ have a sharpness of $\alpha = \kappa_{CC}^2 / 2\delta_{CC}$, which has the worst-case value $\alpha = \kappa_{\max}^2 / 2\delta_{CC}$. It follows that the sharpness is unbounded if δ_{CC} approaches zero, and we can only state that it is less than infinity, because $\delta_{CC} = 0$ means no motion. Hence the curvature is continuous and the maximal allowed speed is greater than zero everywhere.

Let us determine the traveling time along an E segment in case of $|\kappa_{CC}| = \kappa_{\max}$. For this purpose we can use the velocity constraint (14) along the path and the following

identities related to the α , γs , δ , and κ parameters of a clothoid [50]:

$$\begin{aligned}\gamma s &= \frac{2\delta}{\kappa}, \\ \alpha &= \frac{2\delta}{s^2}.\end{aligned}\quad (86)$$

Based on these we obtain the length of an E segment (twice as long as a clothoid), the maximal allowed absolute velocity and the minimal travel time along the segment:

$$\begin{aligned}s(\delta_{CC}) &= \frac{4|\delta_{CC}|}{\kappa_{\max}}, \\ v_{\max}(\delta_{CC}) &= \frac{\sigma_{\max}}{|\alpha(\delta_{CC})|}, \\ t_{\min}(\delta_{CC}) &= \frac{s(\delta_{CC})}{v_{\max}(\delta_{CC})} = \frac{4|\delta_{CC}|}{\kappa_{\max}} \cdot \frac{\kappa_{\max}^2}{2|\delta_{CC}|\sigma_{\max}} \\ &= \frac{2\kappa_{\max}}{\sigma_{\max}}.\end{aligned}\quad (87)$$

It turns out that the traveling time along such an E segment is finite and lower bounded by $2\kappa_{\max}/\sigma_{\max}$, independently from δ_{CC} . Thus we can state that the car can go along the planned path in finite time if the number of path segments is finite.

4.6. The TTS Planning Algorithm. At the end of this section, let us summarize how we can use *TTS* paths and the $e\bar{e}S$ steering method in an approximation framework for the purpose of generating collision-free local paths in the presence of obstacles. A mixed algorithm was implemented, which consists of two parts: the first building block is a sampling-based method relying on general *TTS* paths; the second one is the exact $e\bar{e}S$ steering method, as described in Section 4.5.2. We will denote this mixed algorithm as the *TTS-planner* in the sequel. It takes a number of samples from the set of *TTS* paths connecting q_I and q_G and fulfilling the maximal curvature and sharpness constraints. Additionally, it computes the unique $e\bar{e}S$ solution between q_I and q_G as well. From the resulting local path candidates the colliding ones are excluded and the shortest is returned from the remaining set. If every candidate is in collision, then the *TTS-planner* tries to find new candidates by switching the roles of initial and goal configurations. When no collision-free local path candidate could be found, the *TTS-planner* reports failure, and the approximation algorithm proceeds with subdividing the global path. The reason of using this mixed local planner approach is twofold. At one hand the $e\bar{e}S$ -planner part guarantees convergence of the approximation process, according to its topological property. On the other hand, the sampling part gives a good chance of generating good quality paths by reducing the number of necessary subdivision steps during approximation. In fact, the use of $e\bar{e}S$ steering method alone would be sufficient to ensure convergence, but the sampling part gives a number of alternatives which contribute to path quality and are beneficial when the $e\bar{e}S$ solution collides.

```
(1)  $\mathcal{F}.Init(q_{init})$ 
(2) for all  $k = 1$  to  $K$  do
(3)    $q_{rand} \leftarrow RandomConfig()$ 
(4)    $q_{near} \leftarrow \mathcal{F}.NearestNeighbor(q_{rand})$ 
(5)   if  $q_{new} \leftarrow \mathcal{F}.Connect(q_{rand}, q_{near})$  then
(6)      $\mathcal{F}.AddVertex(q_{new})$ 
(7)      $\mathcal{F}.AddEdge(q_{near}, q_{new})$ 
(8)   end if
(9) end for
(10) return  $\mathcal{F}$ 
```

ALGORITHM 1: The basic RRT construction algorithm [44].

5. Global Planning: The RTR-Planner

In Section 4.4 we argued that “human-like” driving prefers moving along straight path segments. We designed the *TTS-planner* with this assumption in mind; indeed it tries to find a local path which arrives at the straight line defined by q_G (or q_I). In order to obtain an effective approximation-based planning solution, we need a global planner which facilitates the local planning phase. Thus a preliminary global path is required which is not necessarily feasible for the car, but can be approximated easily with the *TTS-planner*.

For this purpose we decided not to use any general holonomic planning algorithm; however the topological property of our local planner ensures approximation convergence for any collision-free paths. Instead of this, we looked for a preliminary path which is “almost” feasible for the car. The solution has been found by omitting the minimum turning radius and designing a global planner which uses only straight motion and turning-in-place primitives. The resulting paths are directly applicable for differential drive robots, which have similar motion model as cars except the turning radius constraint. The planning method is based on the Rapidly Exploring Random Trees (RRT) approach [8–10] which is known for its fast convergence properties even in high dimensional configuration spaces and widely used in path planning applications.

Let us first summarize the basic RRT planning process and after that we derive our RTR (rotate-translate-rotate) approach.

5.1. Rapidly Exploring Random Trees. The basic RRT construction process can be seen in Algorithm 1. The building of the tree \mathcal{F} starts from the initial configuration. *RandomConfig()* returns a random configuration q_{rand} from the configuration space \mathcal{C} (sampling step). *NearestNeighbor()* determines the nearest configuration q_{near} in the tree, according to a metric defined on the configuration space (vertex selection step). It depends on the implementation if this function can return only graph vertices or inner configurations of edges as well. *Connect()* tries to connect q_{near} to q_{rand} by simply interpolating between them (tree extension step). It extends q_{near} until it is connected to q_{rand} or a collision is detected. In the latter case q_{new} will be the farthest collision-free configuration towards q_{rand} . In order to reach the goal

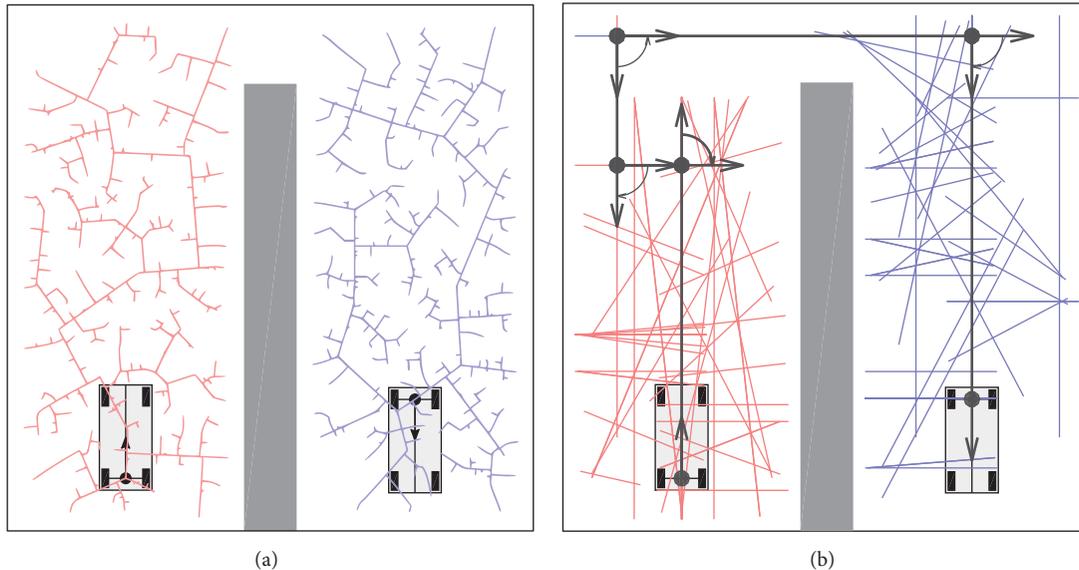


FIGURE 10: Illustration of RRT and RTR global planning. (a) RRT trees after 1000 iterations and (b) RTR trees and solution path after 65 iterations.

configuration, the random sampling can be biased to include q_{goal} sometimes in the random sequence, or a bidirectional search [10] can be performed by growing two trees from both the initial and goal configurations.

The RRT method can be inherently applied to nonholonomic systems. Instead of a simple interpolation towards q_{rand} (which assumes free mobility in any directions) a system-specific local planner should be applied in the *Connect()* step. The original RRT version proposed in [8] recommended choosing an input and applying for a given time quantum Δt to obtain a Δq which brings q_{near} closer to q_{rand} . This was augmented in [9] for such systems where an explicit local planning (steering) method is available to connect two configurations. Actually we build our approach on this *RRT-Connect* version.

For our purpose (i.e., for differential drive) the simplest steering method is the following sequence of straight motion and turning-in-place primitives: (1) turn to head the next position, (2) move straight until it is reached, and (3) turn to the desired orientation. If a collision occurs during this motion sequence, the extension is stopped at the last collision-free configuration. This simple method has the advantage that it delivers an exact solution between two configurations (no sampling of the input set is needed). Furthermore the tree edges will be straight thus making the nearest neighbor search easy, even if inner configurations of edges are involved in the search. This latter property helps to keep the number of tree vertices as low as possible, although we experienced that a naive application of this rotate-translate-rotate steering method together with RRT has poor performance if narrow areas have to be crossed in order to obtain a result. The reason is that in narrow places the collision will occur most likely during the first rotation primitive; hence no translation, that is, no effective extension of the tree will be achieved. An example scenario with a

TABLE 1: Performance measures of RRT and RTR (based on 100 runs).

	RRT	RTR
Success Ratio	25%	100%
Avg. no. of iterations	882.5	65.4

narrow passage is depicted in Figure 10, with illustrative results of RRT and the below described RTR planners. Both algorithms have been run 100 times; in every run maximum 1000 iterations were allowed. We found that RRT had only 25% success rate and 882.5 iterations on average, while RTR was successful in every trial with averagely 65.4 iterations (see Table 1). We describe the operation of RTR-planner in the sequel.

5.2. RT-Trees and Primitives. RTR-planner is similar to a bidirectional RRT-Connect algorithm; however, it has differences in the sampling, the vertex selection, and the extension steps as well. It uses translation (T) and rotation (R) primitives for building the trees. In the extension steps, only RT (rotate-translate) sequences are used instead of the exact rotate-translate-rotate steering method. For this reason, we denote the constructed trees as RT-trees. The vertices of the trees are configurations as usual and the edges are continua of configurations. According to T and R motion primitives, these are called Translational Configuration Intervals (TCIs) and Rotational Configuration Intervals (RCIs). The process of an RT-tree construction can be seen in Algorithm 2 and is detailed in the sequel.

5.2.1. Sampling. The first difference to RRT can be found in the sampling step. *RandomPos()* returns a random position p_G without orientation, denoted as the guiding position in the sequel, instead of a configuration. It can be treated as

```

(1) function RT.CONSTRUCT( $q_{init}$ )
(2)    $\mathcal{T}$ .Init( $q_{init}$ )
(3)   for all  $k = 1$  to  $K$  do
(4)      $p_G \leftarrow$  RandomPos()
(5)      $q_{near} \leftarrow$   $\mathcal{T}$ .NearestNeighbor( $p_G$ )
(6)      $turnDir \leftarrow$  MinTurnDir( $q_{near}, p_G$ )
(7)      $collision \leftarrow$   $\mathcal{T}$ .Extend( $q_{near}, turnDir$ )
(8)     if  $collision$  then
(9)        $\mathcal{T}$ .Extend( $q_{near}, -turnDir$ )
(10)    end if
(11)  end for
(12)  return  $\mathcal{T}$ 
(13) end function
(14)
(15) function  $\mathcal{T}$ .Init( $q$ )
(16)    $\mathcal{T}$ .AddVertex( $q$ )
(17)    $TCI \leftarrow$  TCIEExtend( $q$ , "forward")
(18)    $\mathcal{T}$ .AddVertex( $TCI.q_{end}$ )
(19)    $\mathcal{T}$ .AddEdge( $q, TCI$ )
(20)    $TCI \leftarrow$  TCIEExtend( $q$ , "backward")
(21)    $\mathcal{T}$ .AddVertex( $TCI.q_{end}$ )
(22)    $\mathcal{T}$ .AddEdge( $q, TCI$ )
(23) end function
(24)
(25) function  $\mathcal{T}$ .Extend( $q, turnDir$ )
(26)   [ $RCI, collision$ ]  $\leftarrow$  RCIEExtend( $q, turnDir$ )
(27)    $\mathcal{T}$ .AddVertex( $RCI.q_{end}$ )
(28)    $\mathcal{T}$ .AddEdge( $q, RCI$ )
(29)    $TCI \leftarrow$  TCIEExtend( $RCI.q_{end}$ , "forward")
(30)    $\mathcal{T}$ .AddVertex( $TCI.q_{end}$ )
(31)    $\mathcal{T}$ .AddEdge( $RCI.q_{end}, TCI$ )
(32)    $TCI \leftarrow$  TCIEExtend( $RCI.q_{end}$ , "backward")
(33)    $\mathcal{T}$ .AddVertex( $TCI.q_{end}$ )
(34)    $\mathcal{T}$ .AddEdge( $RCI.q_{end}, TCI$ )
(35) return  $collision$ 
(36) end function

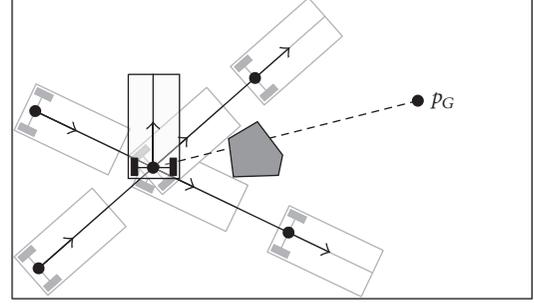
```

ALGORITHM 2: Construction of an RT-tree.

a one-dimensional continuous set of configurations, from which any element can serve as local goal in the tree extension step.

5.2.2. Vertex Selection. \mathcal{T} .NearestNeighbor() returns the configuration in the existing tree which has the smallest *position* distance to p_G . This step uses a simple Euclidean metric; hence no special configuration space metrics are needed.

5.2.3. Extension. The main difference to the RRT method can be found in the tree extension step. On the one hand, translation is always performed in both forward and backward directions. Additionally, the translation is not stopped when p_G is reached, but continued until the first collision in both directions (this is the functionality of $TCIEExtend()$, called by both \mathcal{T} .Init() and \mathcal{T} .Extend() functions). On the other hand, an important difference is the fact that an R-T sequence will always be planned, even if a collision occurs in the rotation phase. The RTR-planner balances between reaching guiding positions and extending the tree. If p_G

FIGURE 11: Illustration of the tree extension procedure if the direction to p_G is blocked.

cannot be reached due to a collision, then two things can be done:

- (1) If the collision occurred during the T primitive, then the iteration is finished (because the tree has been extended).
- (2) If the collision occurred during the R primitive, then $TCIEExtend$ is performed in both forward and backward directions at the colliding orientation (first extension), and the rotation is tried again in the other turning direction as well. After the second rotation, independently from its success or collision, $TCIEExtend$ is called again (second extension). An example of this procedure can be seen in Figure 11.

This extension strategy causes a more aggressive free space exploration than in case of the basic RRT method.

5.3. Connecting RT-Trees. As the RTR-planner builds two RT-trees—from both the initial and goal configurations—in order to obtain a final path, the two trees have to be connected. This is attempted in every iteration. The newly added TCIs are checked against every TCI in the other tree, starting from its root. If an intersection is found and if a collision-free RCI can be put between the intersecting TCIs to connect them, then the two trees can be merged and the path determined easily by tracing the trees back to their roots.

6. Simulation Results

The RTR + TTS planning algorithm was tested extensively in simulations and compared to other similar approaches in terms of effectiveness and path quality. To two other algorithms were investigated:

- (i) *RRT-Connect (CCRS)*. Bidirectional RRT using the CCRS steering method as local planner (direct one-step approach, no approximation needed).
- (ii) *RTR + CCRS*. The RTR-planner is used for generating a preliminary global path, followed by an approximation phase using CCRS paths.

The following scenarios have been used for testing:

- (i) A wide environment with few obstacles (can be treated as an easy planning task, see Figure 12).

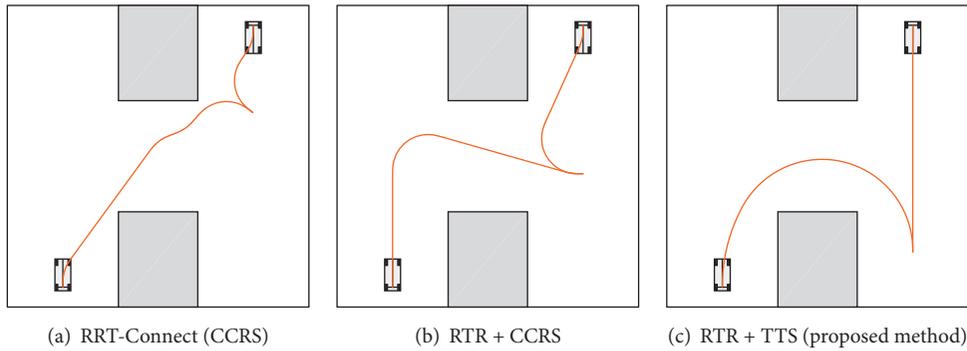


FIGURE 12: Path planning in a wide environment.

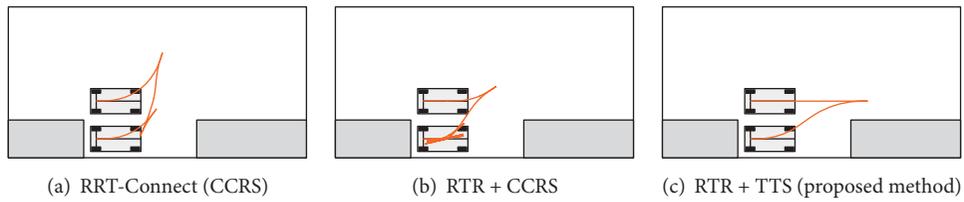


FIGURE 13: Path planning for parallel parking.

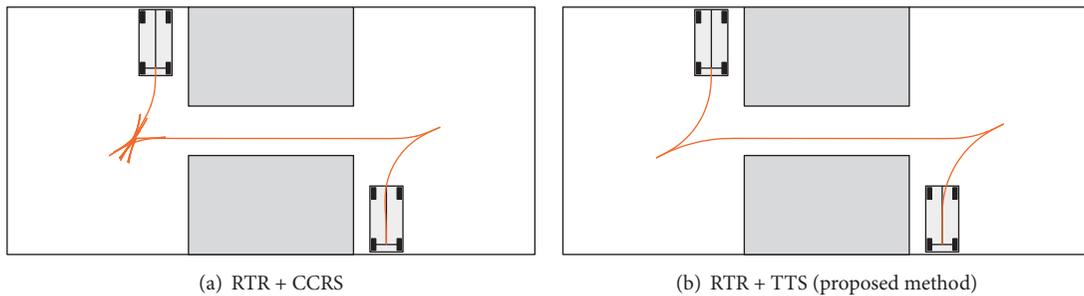


FIGURE 14: Crossing a narrow corridor.

- (ii) A simple parallel parking scenario (a well known problem for path planning algorithms, see Figure 13).
- (iii) Driving through a narrow corridor and parking next to the wall (can be treated as a difficult planning task, see Figure 14).
- (iv) Crossing three perpendicular narrow corridors with bigger free areas at the corners (can be treated as a very difficult planning task, see Figure 15).

The simulated car is 4 m long and 2 m wide with 4.42 m minimal turning radius.

To obtain practical results, we constrained the maximum iteration count of the tree building process to 10000 in every algorithm and applied a lower bound of path subdivision at the approximation phase. Every algorithm has been run 100 times in every scenario, and the average results are shown in Figure 16. The performance and path quality is measured by the following metrics:

- (i) *Success Ratio*. Percentage of successful runs.

- (ii) *Number of Cusps*. The number of reversals along the path. A good path contains as few reversals as possible.
- (iii) *Steering Amount*. The absolute integral of the path curvature. It gives the amount of turning along the path. A path with less sharp turns or with more straight segments is more comfortable than a path requiring much turning along the way.
- (iv) *Travel Time*. The time of driving along the resulting path is estimated as follows. A traveling speed function with $v_{max} = 5 \text{ m/s}$ and $v_{min} = 1 \text{ m/s}$ is assigned to the path, which is inversely proportional to the actual path curvature. The cusps are penalized with 0.5 s “reversing time.” Those paths are better, which require less time to travel along.

Numerical results are listed in Figure 16, while illustrative examples can be examined in Figures 12–15.

As expected, the “easy” path planning task in wide environment can be flawlessly solved by all selected algorithm

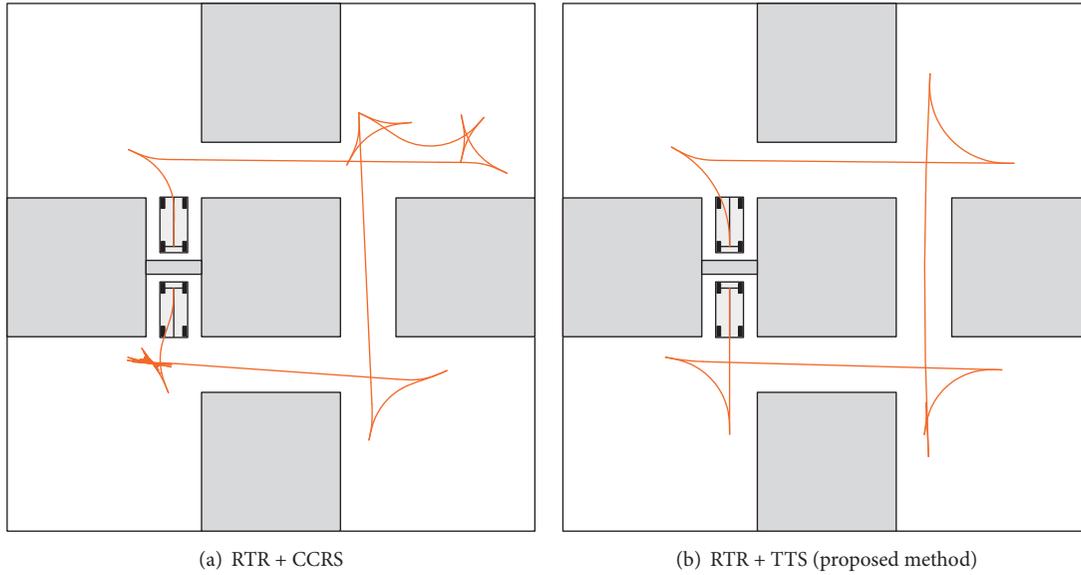


FIGURE 15: Crossing three narrow corridors.

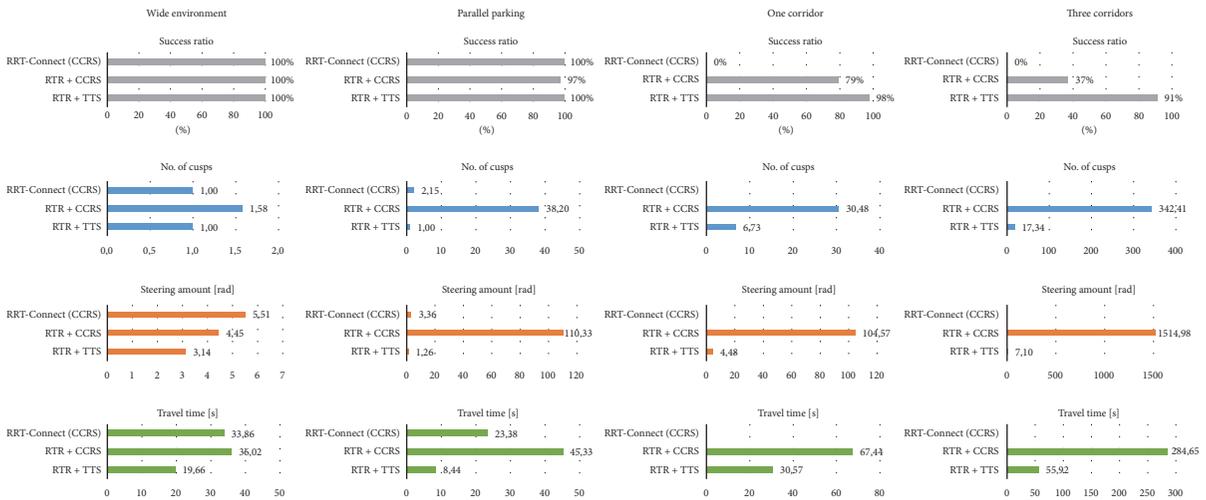


FIGURE 16: Summary of the simulation results.

combinations (Figure 12). The path quality measures are similar (first column in Figure 16); however the proposed RTR + TTS-planner combination results in the most comfortable and human-like paths with respect to steering amount and travel time.

Similarly, the parallel parking task is solvable as well by every tested algorithm (Figure 13). The TTS-based local planner obtains the highest quality paths. This is not surprising because the most “natural” parallel parking maneuver is a TTS path itself. The reason for obtaining more complicated paths with CCRS-based methods is that the CCRS planner does not have the “natural” solution amongst its local path candidates; hence it relies stronger on the global planning phase. During parallel parking we want to achieve a lateral shift to a narrow place which is quite unnatural for the car; this results in many approximation steps for the RTR

+ CCRS planner, producing an enormous number of small movements. The result is much better if the CCRS local planner is applied directly in the RRT-Connect planning phase, because in this case the unnatural approximation is avoided. However, the RTR + TTS-planner outperforms this as well (see the second column in Figure 16).

As we move to more complicated planning challenges in environments containing narrow corridors, sampling-based planners need more iterations to find a solution, resulting in lower success rates (because the number of iterations is limited). Those global planners which tend to use mainly straight movements are more successful than others. In the scenario with one corridor the direct RRT-Connect method completely fails since it uses curved local paths to connect configuration samples (for this reason only RTR + CCRS and RTR + TTS solutions are depicted in

Figure 14). Approximation methods are better suited to such environments and it can be seen in the third column of 14 that the TTS local planner results in higher success ratios and better quality paths. The effectiveness of RTR as global planner originates from the aggressive extension step and from the fact that it contains only straight forward and backward translation movements. Because TTS paths are designed to arrive at straight lines, they are well suited for the approximation of RTR paths, resulting in less subdivision steps (efficiency) and simpler path geometry (quality).

The effectiveness of the RTR global planner can mostly be examined in the last scenario, where three narrow corridors have to be passed (Figure 15). Comparing the approximation result using CCRS and TTS local planners, we can find a much higher efficiency and much better path quality in case of TTS. Furthermore, we can see the different philosophies behind the two local planners; namely, CCRS paths strive to have the maximal allowed curvature, while TTS paths are not restricted to this. In the larger “turning spaces” between corridors, the TTS-planner generates less sharp turns and needs less reversals, which increases the comfort of passengers in the car. The direct RRT-Connect method with CCRS local planner completely fails, similarly to the previous example. As can be seen in the last column of Figure 16, the RTR + TTS-planner significantly outperforms the other algorithms.

7. Conclusions

In this paper we presented a global planning method for car-like vehicles, producing paths with continuous curvature. A comparative analysis with possible alternatives was performed as well and we could see that both components (RTR and TTS) of the two-phase planning approach contribute to the effectiveness and path quality of the resulting algorithm. The (preliminary) RTR path planner is capable of designing paths consisting of straight movements and turning-in-place. We apply the TTS local planner to the RTR path in a second approximation phase, in order to obtain a final path obeying the bounded and continuous curvature conditions. The TTS-planner uses straight segments, CC-turns, and elementary paths in order to generate a feasible and human-like solution even in cluttered and narrow environments. We gave an overview of how to build CC-turns and elementary paths; the class of curvature and sharpness bounded EES and TTS paths were introduced, and existence conditions were given for them. An exact steering method: the *eES*-planner was presented as well, which delivers a unique EES solution for a planning query and verifies the topological property. Simulation studies were presented to illustrate that the proposed RTR + TTS planning algorithm can be applied universally in less and more difficult situations and the obtained paths are quite “natural.”

We continue this work in the future by implementing the proposed algorithm on our physical testbed [52, 53] utilizing small-sized model cars and later on a real vehicle to perform real-life tests and performance measurements. Further research includes the investigation of possible applications in previously unknown and changing environments, and the

interaction possibilities of this planning method with online map building algorithms.

Appendix

A. Existence Conditions for EES Paths

Lemma 2 stated that, in the absence of obstacles, any (q_I, q_G) pair of initial and goal configurations of a continuous steering car can be connected by an EES path. Here we give its proof, where we use some properties of $B(\cdot)$ and $D(\cdot, \cdot)$ functions, which are established in Appendix C.

Proof of Lemma 2. Let us apply the curvature bound to κ_2 first:

$$|\kappa_2| \leq \kappa_{\max}. \quad (\text{A.1})$$

After substituting (59b) it gets the form

$$|\kappa_2| = \left| \frac{B(\tilde{\theta}_I)}{\tilde{y}_I} \right| \leq \kappa_{\max}. \quad (\text{A.2})$$

Using the fact that $B(\cdot)$ is nonnegative in the interval of interest (cf. Property C.3 in Appendix C) we can rearrange this to

$$|\tilde{y}_I| \geq \frac{B(\tilde{\theta}_I)}{\kappa_{\max}} = |\Delta y_{2,\min}(\tilde{\theta}_I)|, \quad (\text{A.3})$$

which means that, for a given $\tilde{\theta}_I = -2\delta_2$, a valid ES subpath exists if and only if \tilde{q}_I is farther from the x -axis than the minimal possible change in the y -coordinate determined by the necessary deflection and the given curvature constraint.

We can obtain the necessary and sufficient conditions for the existence of EES paths by substituting (55a), (55b), (55c), and (55d) into (A.3):

$$\left| y_I + \frac{D(2\delta_1, \theta_I)}{\kappa_1} \right| \geq \frac{B(2\delta_1 + \theta_I)}{\kappa_{\max}}, \quad (\text{A.4})$$

alternatively written as

$$|y_I + \Delta y_1(\theta_I, \delta_1, \kappa_1)| \geq |\Delta y_{2,\min}(\theta_I, \delta_1)|, \quad (\text{A.5})$$

together with the obvious other condition $|\kappa_1| \leq \kappa_{\max}$.

According to Property C.5 in Appendix C, $D(2\delta_1, \theta_I) = 0$ only if $\delta_1 = 0$ or $\delta_1 = -\theta_I \pm \pi$. This means that, for any θ_I and $\delta_1 \notin \{0, -\theta_I \pm \pi\}$, the Δy_1 term is nonzero and a κ_1 can be found such that (A.5) and the curvature bound are fulfilled, because

$$\lim_{\kappa_1 \rightarrow 0} \left| \frac{D(2\delta_1, \theta_I)}{\kappa_1} \right| = \infty. \quad (\text{A.6})$$

We have seen that for any query pair of the form $(q_I, 0)$ a class of EES paths—parametrized by (δ_1, κ_1) —is available. A more general query pair (q_I, q_G) can be transformed to the form $(q_I', 0)$ by the inverse transformation $T^{-1}(q_G)$, and the resulting path can be transformed back by $T(q_G)$. \square

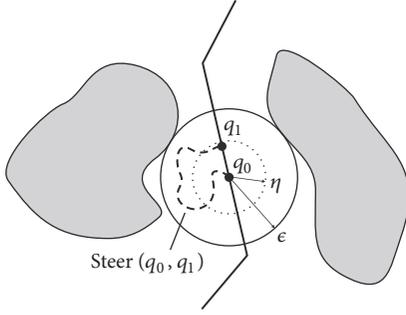


FIGURE 17: Illustration of topological property for $\mathcal{C} = \mathbb{R}^2$.

B. Topological Property of the $e\bar{e}S$ Steering Method

The topological property of steering methods is defined as follows [12].

Definition B.1 (topological property). A steering method $\text{Steer}(q_0, q_1)$ verifies the topological property if

$$\begin{aligned} \forall \epsilon > 0, \\ \exists \eta > 0, \\ \forall q_0, q_1 \in \mathcal{C} \end{aligned} \quad (\text{B.1})$$

$$d_{\mathcal{C}}(q_0, q_1) < \eta \implies d_{\mathcal{C}}(q_0, \text{Steer}(q_0, q_1)(s)) < \epsilon, \quad \forall s \in [0, S_{\lambda}],$$

where $d_{\mathcal{C}}$ is any metric defined on the configuration space \mathcal{C} and S_{λ} is the total length of the of the path returned by $\text{Steer}(q_0, q_1)$.

This means that, for every nonzero ϵ , η can be found such that if q_1 is in an η -neighborhood of q_0 , then the local path generated by Steer does not exit the ϵ -neighborhood of q_0 . When approximating a global path and having q_0 as an intermediate point on it which has a collision-free ϵ -neighborhood, then we can find another intermediate point q_1 along the path in the η -neighborhood of q_0 such that the feasible local path $\text{Steer}(q_0, q_1)$ is collision-free (see Figure 17). Hence if the global path has nonzero clearance, then the topological property ensures that a feasible path can be constructed along it using a sequence of local paths generated by the steering method.

We will show in the sequel that the $e\bar{e}S$ steering method has this advantageous property. But before that, a metric $d_{\mathcal{C}}$ has to be defined over \mathcal{C} . The most important metrics over the n -dimensional Euclidean space \mathbb{R}^n belong to the family of L_p metrics [6]:

$$L_p(r, r') = \left(\sum_{i=1}^n |r_i - r'_i|^p \right)^{1/p}, \quad r, r' \in \mathbb{R}^n. \quad (\text{B.2})$$

A special L_p metric is the L_{∞} metric which can be obtained from (B.2) if $p \rightarrow \infty$:

$$L_{\infty}(r, r') = \max_{1 \leq i \leq n} \{|r_i - r'_i|\}. \quad (\text{B.3})$$

The configuration space of the continuous steering car is not an Euclidean space but a manifold $\mathcal{C} = \mathbb{R}^2 \times \mathbb{S}^1 \times [-\kappa_{\max}, \kappa_{\max}]$, where \mathbb{S}^1 stands for the unit circle which is homeomorphic to $[-\pi, \pi]/\sim$. Because of the identification of $-\pi$ and π , special care must be taken when the “distance” of two angles has to be measured. An appropriate metric for \mathbb{S}^1 can be the following [6]:

$$d_{\mathbb{S}^1}(\theta, \theta') = \min \{|\theta - \theta'|, 2\pi - |\theta - \theta'|\}. \quad (\text{B.4})$$

As stated in [6], metric spaces are extendable by Cartesian product and a new metric can be defined for the extended space using the original metrics. Since our configuration space \mathcal{C} is an extension over \mathbb{R}^2 and \mathbb{S}^1 , we can choose L_{∞} metric for \mathbb{R}^2 , $d_{\mathbb{S}^1}$ for \mathbb{S}^1 , and L_{∞} again over these:

$$\begin{aligned} d_{\mathcal{C}}(q_0, q_1) \\ = L_{\infty}(L_{\infty}((x_0, y_0), (x_1, y_1)), d_{\mathbb{S}^1}(\theta_0, \theta_1)), \end{aligned} \quad (\text{B.5})$$

which leads to

$$\begin{aligned} d_{\mathcal{C}}(q_0, q_1) = \max \{ |x_0 - x_1|, |y_0 - y_1|, \\ \min \{ |\theta_0 - \theta_1|, 2\pi - |\theta_0 - \theta_1| \} \}. \end{aligned} \quad (\text{B.6})$$

Note that we have neglected one dimension of the configuration space (the $[-\kappa_{\max}, \kappa_{\max}]$ part) in the metric. This is valid because the curvatures of q_I and q_G are always 0; thus the “curvature distance” of these is zero as well. Furthermore, note that in this metric Euclidean distances are compared to angles, which is a common problem in motion planning. It is hard to define a metric over \mathcal{C} that can avoid this issue. Nevertheless, the above defined metric is appropriate for proving the following theorem.

Theorem B.2. *The $e\bar{e}S$ steering method verifies the topological property.*

Proof. Since the goal configuration is at the origin by assumption (without loss of generality), the metric (B.6) can be written in the following form:

$$d_{\mathcal{C}}(q(s)) = \max \{ |x(s)|, |y(s)|, |\theta(s)| \}, \quad (\text{B.7})$$

where $q(s)$ is a shorthand for $\text{Steer}(0, q_I)(s)$. The total length of an $e\bar{e}S$ path is

$$S_{\lambda} = s_1 + s_2 + s_3 = \left| \frac{4\delta_1}{\kappa} \right| + \left| \frac{4\delta_2}{\kappa} \right| + s_3, \quad (\text{B.8})$$

where s_1 , s_2 , and s_3 are the corresponding path segment lengths, according to (86). It is obvious that

$$|x(s)| \leq s \leq S_{\lambda}, \quad (\text{B.9})$$

$$|y(s)| \leq s \leq S_{\lambda}, \quad (\text{B.10})$$

$$|\theta(s)| \leq |2\delta_1| + |2\delta_2| \quad (\text{B.11})$$

for all $s \in [0, S_\lambda]$. After multiplying (B.11) with $2/|\kappa|$ we get

$$\frac{2|\theta(s)|}{|\kappa|} \leq s_1 + s_2 \leq S_\lambda, \quad (\text{B.12})$$

which results in

$$|\theta(s)| \leq S_\lambda \frac{|\kappa|}{2} \leq S_\lambda \frac{\kappa_{\max}}{2}. \quad (\text{B.13})$$

Looking at (B.7) and (B.9)–(B.13) we can state that

$$d_{\mathcal{E}}(q(s)) \leq S_\lambda \cdot \max \left\{ 1, \frac{\kappa_{\max}}{2} \right\}. \quad (\text{B.14})$$

With this, the statement to be proven is equivalent to

$$d_{\mathcal{E}}(q_I) < \eta \implies S_\lambda \cdot \max \left\{ 1, \frac{\kappa_{\max}}{2} \right\} < \epsilon, \quad (\text{B.15})$$

which holds when

$$\lim_{q_I \rightarrow 0} S_\lambda = 0. \quad (\text{B.16})$$

According to (83), $q_I \rightarrow 0$ means that $|\kappa^*| \rightarrow \infty$; hence $|\kappa| \rightarrow \kappa_{\max}$. It follows that

$$\lim_{q_I \rightarrow 0} S_\lambda = \frac{|4\delta_1|}{\kappa_{\max}} + \frac{|4\delta_2|}{\kappa_{\max}} + s_3. \quad (\text{B.17})$$

Let us define the function

$$F(y_I, \theta_I, \delta_1) := y_I + \frac{G(2\delta_1, \theta_I)}{\kappa} \quad (\text{B.18})$$

for any fixed κ . According to (77b) we know that for any $e\bar{e}S$ path

$$F(y_I, \theta_I, \delta_1) = 0. \quad (\text{B.19})$$

Let us examine the behavior of $F(\cdot)$ in the vicinity of $(y_I, \theta_I) = (0, 0)$:

$$\begin{aligned} \lim_{(y_I, \theta_I) \rightarrow (0,0)} F(y_I, \theta_I, \delta_1) &= 0 + \frac{G(2\delta_1, 0)}{\pm \kappa_{\max}} \\ &= \pm \frac{2B(2\delta_1)}{\kappa_{\max}}, \end{aligned} \quad (\text{B.20})$$

where we applied Property C.4 in Appendix C. This implies together with (B.19)

$$\pm \frac{2B(2\delta_1)}{\kappa_{\max}} = 0. \quad (\text{B.21})$$

Property C.3 in Appendix C states that

$$B(2\delta_1) = 0 \iff \delta_1 = 0, \quad (\text{B.22})$$

which implies

$$\lim_{(y_I, \theta_I) \rightarrow (0,0)} F(y_I, \theta_I, \delta_1) = 0 \iff \delta_1 \rightarrow 0. \quad (\text{B.23})$$

In other words, in the neighborhood of $(y_I, \theta_I) = (0, 0)$ an $e\bar{e}S$ path can only exist iff $\delta_1 \rightarrow 0$.

To sum it up, we now that if $q_I \rightarrow 0$, then

$$\begin{aligned} |\kappa| &\longrightarrow \kappa_{\max}, \\ \delta_1 &\longrightarrow 0, \\ \delta_2 &= -\delta_1 - \frac{\theta_I}{2} \longrightarrow 0. \end{aligned} \quad (\text{B.24})$$

Furthermore, based on (77a) we get

$$\lim_{q_I \rightarrow 0} s_3 = \lim_{q_I \rightarrow 0} \left| \frac{A(2\delta_1 + \theta_I) + C(2\delta_1, \theta_I)}{\kappa} \right| = 0, \quad (\text{B.25})$$

where we used (B.23) and Property C.1 in Appendix C. Finally, we can conclude that

$$\lim_{q_I \rightarrow 0} S_\lambda = \frac{|4\delta_1|}{\kappa_{\max}} + \frac{|4\delta_2|}{\kappa_{\max}} + s_3 \stackrel{0}{=} 0, \quad (\text{B.26})$$

which proves (B.15); thus the requirement for the topological property is fulfilled. \square

C. Auxiliary Functions

Some auxiliary functions are used during deriving the EES and $e\bar{e}S$ paths:

$$X(\beta) = \text{sgn}(\beta) \sqrt{\pi|\beta|} \cdot C_F \left(\sqrt{\frac{|\beta|}{\pi}} \right) \quad (\text{C.1})$$

$$Y(\beta) = \sqrt{\pi|\beta|} \cdot S_F \left(\sqrt{\frac{|\beta|}{\pi}} \right) \quad (\text{C.2})$$

$$A(\beta) = X(\beta)(1 + \cos \beta) + Y(\beta) \sin \beta \quad (\text{C.3})$$

$$B(\beta) = X(\beta) \sin \beta + Y(\beta)(1 - \cos \beta) \quad (\text{C.4})$$

$$C(\beta, \psi) = A(\beta) \cos \psi - B(\beta) \sin \psi \quad (\text{C.5})$$

$$D(\beta, \psi) = A(\beta) \sin \psi + B(\beta) \cos \psi \quad (\text{C.6})$$

$$G(\beta, \psi) = B(\beta + \psi) + D(\beta, \psi). \quad (\text{C.7})$$

Because all of these contain Fresnel integrals, it is hard to solve equations including these functions if the unknown variable is in the argument. However, some function properties can be specified including parity, zeros, and specific values. These properties are useful when dealing with expressions containing them.

Property C.1. The functions $X(\beta)$, $Y(\beta)$, $A(\beta)$, $B(\beta)$, $C(\beta, \psi)$, and $D(\beta, \psi)$ have zero value at $\beta = 0$.

Proof. For $X(\beta)$ and $Y(\beta)$ the value at $\beta = 0$ can be obtained by substitution: all factors are zero. It follows from the definitions that $X(0)$ and $Y(0)$ make every term zero in the other functions as well. \square

TABLE 2: Signs of the terms of $B(\beta)$.

β	$X(\beta)$	$\sin \beta$	$Y(\beta)$	$1 - \cos \beta$	$B(\beta)$
$-\pi$	-	0	+	+	+
$(-\pi, 0)$	-	-	+	+	+
0	0	0	0	0	0
$(0, \pi)$	+	+	+	+	+
π	+	0	+	+	+

Property C.2. $X(\beta)$ and $A(\beta)$ are odd, while $Y(\beta)$ and $B(\beta)$ are even functions. Additionally, $\text{sgn}(X(\beta)) = \text{sgn}(\beta)$ and $Y(\beta) \geq 0$.

Proof. It follows from the definition of Fresnel sine and cosine integrals (18a) and (18b) that they are odd functions and their signs are the same as the sign of their arguments. In the definitions of $X(\beta)$ and $Y(\beta)$, the arguments of C_F and S_F depend on the absolute value of β , which results in even functions with nonnegative values. The multiplication factor $\sqrt{\pi|\beta|}$ is even as well, while the $\text{sgn}(\cdot)$ function is odd. We know that the product of two functions is even if the parities of factors are the same. Similarly, if the parities differ, then the resulting function is odd. Based on these, it is obvious that $X(\beta)$ is odd with $\text{sgn}(X(\beta)) = \text{sgn}(\beta)$ and $Y(\beta)$ is even and nonnegative.

Given that the sine function is odd and cosine is even and $X(0) = Y(0) = 0$, it can be easily seen that both terms in the definition of $A(\beta)$ are odd; thus $A(\beta)$ itself is odd as well. Similarly, both terms in $B(\beta)$ are even; thus $B(\beta)$ itself is even as well. \square

Property C.3. In the range $\beta \in [-\pi, \pi]$, $B(\beta)$ is positive except at $\beta = 0$, where it has zero value.

Proof. Let us see the sign of terms of $B(\beta)$ in different subsets of range $\beta \in [-\pi, \pi]$, as listed in Table 2. It can be seen that $B(\beta)$ is a sum of positive or zero-valued terms. Only $X(\beta)$ and $\sin \beta$ can be negative, but in all such cases the product of them is positive or zero. The table shows that

$$B(\beta) = 0 \iff \beta = 0, \quad (\text{C.8})$$

$$B(\beta) > 0 \iff \beta \neq 0. \quad (\text{C.9})$$

\square

Property C.4. $D(\beta, \psi)$ has the following specific values:

$$D(\beta, 0) = B(\beta) \quad (\text{C.10})$$

$$D(-\beta, \beta) = D(\beta, -\beta) = -B(\beta). \quad (\text{C.11})$$

Proof. (C.10) can be verified easily by substituting $\psi = 0$ into (C.4). Before looking at the second equation, let us reformulate the definition of $D(\beta)$ as follows:

$$D(\beta, \psi) = X(\beta) (\sin \psi + \sin(\beta + \psi)) + Y(\beta) (\cos \psi - \cos(\beta + \psi)). \quad (\text{C.12})$$

We used here the definitions of $A(\beta)$ and $B(\beta)$ and trigonometric addition rules. In both cases of (C.11) we get

$$D(-\beta, \beta) = D(\beta, -\beta) = -X(\beta) \sin \beta + Y(\beta) (\cos \beta - 1), \quad (\text{C.13})$$

which is obviously equal to $-B(\beta)$. \square

Property C.5. The function $D(\beta, \psi)$ has zeros only at $\beta = 0$ and $\beta = -2\psi + 2\pi k$, $k \in \mathbb{Z}$ in the range $\beta \in [-\pi, \pi]$.

Proof. The first condition for $D(\beta, \psi)$ being zero was already stated by Property C.1. The second condition can be verified easily by substitution to (C.12). Furthermore, we have to check whether $D(\beta, \psi)$ has zeros elsewhere. Let us assume that $\beta \neq 0$ and $\beta \neq -2\psi + 2\pi k$, but $D(\beta, \psi) = 0$. In this case the following equality would be true:

$$X(\beta) (\sin \psi + \sin(\beta + \psi)) = -Y(\beta) (\cos \psi - \cos(\beta + \psi)). \quad (\text{C.14})$$

Let us introduce a new variable

$$\alpha := \beta + 2\psi - 2\pi k \neq 0 \quad (\text{C.15})$$

according to our assumption $\beta \neq -2\psi + 2\pi k$. After substituting $\psi = \alpha/2 - \beta/2 + \pi k$ into (C.14) we get

$$X(\beta) \left[\sin\left(\frac{\alpha}{2} - \frac{\beta}{2} + \pi k\right) + \sin\left(\frac{\alpha}{2} + \frac{\beta}{2} + \pi k\right) \right] = -Y(\beta) \cdot \left[\cos\left(\frac{\alpha}{2} - \frac{\beta}{2} + \pi k\right) - \cos\left(\frac{\alpha}{2} + \frac{\beta}{2} + \pi k\right) \right]. \quad (\text{C.16})$$

By applying trigonometric addition rules we can reformulate it:

$$X(\beta) \cdot \cancel{2 \sin\left(\frac{\alpha}{2} + \pi k\right)} \cdot \cos\frac{\beta}{2} = -Y(\beta) \cdot \cancel{2 \sin\left(\frac{\alpha}{2} + \pi k\right)} \cdot \sin\frac{\beta}{2}. \quad (\text{C.17})$$

The $2 \sin(\alpha/2 + \pi k)$ terms can be canceled because they are nonzero according to (C.15). The equality should hold for the signs as well:

$$\text{sgn}(X(\beta)) \text{sgn}\left(\cos\frac{\beta}{2}\right) = -\text{sgn}(Y(\beta)) \text{sgn}\left(\sin\frac{\beta}{2}\right), \quad (\text{C.18})$$

which evaluates to

$$\begin{aligned} \operatorname{sgn}(\beta) &= -\operatorname{sgn}(\beta), \quad \text{if } \beta \in (-\pi, \pi) \setminus 0, \\ 0 &= 1, \quad \text{if } \beta = -\pi, \\ 0 &= -1, \quad \text{if } \beta = \pi. \end{aligned} \quad (\text{C.19})$$

Because all of these are contradictions, the assumption that $D(\beta, \psi)$ has zeros elsewhere than $\beta = 0$ and $\beta = -2\psi + 2\pi k$ was wrong. \square

Property C.6. In the range $\beta \in [-\pi, \pi]$ and $\psi \in [-\pi, \pi]$, for all ψ there exists β such that $G(\beta, \psi) \neq 0$.

Proof. First let us see the case $\psi = 0$. Using (C.7) and (C.10) we get

$$G(\beta, 0) = 2B(\beta). \quad (\text{C.20})$$

According to (C.9) we can state that $G(\beta \neq 0, \psi = 0) \neq 0$.

We can check this for $\psi \neq 0$ as well. For example, let us choose $\psi = -\beta$. Using (C.7) and (C.11) we arrive at

$$G(\beta, -\beta) = -B(\beta). \quad (\text{C.21})$$

Because $\beta = -\psi \neq 0$, based on (C.9) we can state again that in case of $\psi \neq 0$ a β can be found such that $G(\beta, \psi) \neq 0$. \square

Property C.7. In the range $\beta \in [-\pi, \pi]$ and $\psi \in [-\pi, \pi]$, for all ψ there exists β such that $G(\beta, \psi) = 0$.

Proof. Let us look again at the case $\psi = 0$ first. Based on (C.20) and (C.8) we can state that $G(0, 0) = 0$.

In the case $\psi \neq 0$ let us examine two specific values for β :

(i) If $\beta = 0$, then, using (C.7) and Properties C.1 and C.3, we get

$$G(0, \psi) = B(\psi) > 0. \quad (\text{C.22})$$

(ii) If $\beta = -\psi$, then, using (C.7) and (C.11) and Property C.3, we get

$$G(0, \psi) = -B(\psi) < 0. \quad (\text{C.23})$$

It follows that for all ψ there exists a $\beta \in (-\psi, 0)$ such that $G(\beta, \psi) = 0$. \square

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is connected to the scientific program of the "Development of Quality-Oriented and Harmonized R + D + I Strategy and Functional Model at BME" project. This project is supported by the New Széchenyi Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

References

- [1] J. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, Mass, USA, 1991.
- [2] J.-P. Laumond, *Robot motion planning and control*, vol. 229 of *Lecture Notes in Control and Information Sciences*, Springer, London, UK, 1998.
- [3] S. M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006, <http://planning.cs.uiuc.edu/>.
- [4] J. Rigelsford, "Introduction to Autonomous Mobile Robots20043R. Siegwart and I.R. Nourbakhsh.," *Industrial Robot: An International Journal*, vol. 31, no. 6, pp. 534-535, 2004.
- [5] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135-1145, 2016.
- [6] S. M. LaValle, "Sampling-based motion planning," in *Planning Algorithms*, Cambridge University Press, Cambridge, UK, 2006, <http://planning.cs.uiuc.edu/>.
- [7] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566-580, 1996.
- [8] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Technical Report, Computer Science Dept., Iowa State University, 1998.
- [9] J. J. Kuffner Jr. and S. M. la Valle, "RRT-connect: an efficient approach to single-query path planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 995-1001, April 2000.
- [10] S. M. LaValle and J. J. Kuffner Jr., "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378-400, 2001.
- [11] J.-P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray, "A Motion Planner for Nonholonomic Mobile Robots," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 577-593, 1994.
- [12] J. P. Laumond, S. Sekhavat, and F. Lamiroux, "Guidelines in nonholonomic motion planning for mobile robots," in *Robot motion planning and control*, vol. 229 of *Lect. Notes Control Inf. Sci.*, pp. 1-53, Springer, London, UK, 1998.
- [13] G. Lafferriere and H. Sussmann, "Motion planning for controllable systems without drift," in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 1148-1153, April 1991.
- [14] R. M. Murray and S. S. Sastry, "Nonholonomic motion planning: steering using sinusoids," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 38, no. 5, pp. 700-716, 1993.
- [15] S. Sekhavat and J.-P. Laumond, "Topological property for collision-free nonholonomic motion planning: the case of sinusoidal inputs for chained form systems," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 671-680, 1998.
- [16] D. Tilbury, R. M. Murray, and S. S. Sastry, "Trajectory generation for the n -trailer problem using Goursat normal form," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 40, no. 5, pp. 802-819, 1995.
- [17] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327-1361, 1995.

- [18] P. Rouchon, M. Fliess, J. Lévine, and P. Martin, "Flatness, motion planning and trailer systems," in *Proceedings of the 32nd IEEE Conference on Decision and Control*, pp. 2700–2705, IEEE, San Antonio, Tex, USA, December 1993.
- [19] S. Sekhavat, F. Lamiroux, J. P. Laumond, G. Bauzil, and A. Ferrand, "Motion planning and control for Hilare pulling a trailer: Experimental issues," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation, ICRA. Part 4 (of 4)*, pp. 3306–3311, April 1997.
- [20] F. Lamiroux and J.-P. Laumond, "Flatness and small-time controllability of multibody mobile robots: application to motion planning," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 45, no. 10, pp. 1878–1881, 2000.
- [21] F. Lamiroux and J.-P. Laumond, "Smooth motion planning for car-like vehicles," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 498–502, 2001.
- [22] C. P. Tang, P. T. Miller, V. N. Krovi, J. Ryu, and S. K. Agrawal, "Kinematic control of nonholonomic wheeled mobile manipulator: a differential flatness approach," in *Proceedings of the ASME 2008 Dynamic Systems and Control Conference*, pp. 1117–1124, Ann Arbor, Mich, USA, 2008.
- [23] C. P. Tang, "Differential flatness-based kinematic and dynamic control of a differentially driven wheeled mobile robot," in *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics, ROBIO 2009*, pp. 2267–2272, December 2009.
- [24] C. P. Tang, P. T. Miller, V. N. Krovi, J.-C. Ryu, and S. K. Agrawal, "Differential-flatness-based planning and control of a wheeled mobile manipulator-theory and experiment," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 4, pp. 768–773, 2011.
- [25] C. Fernandes, L. Gurvits, and Z. X. Li, "A variational approach to optimal nonholonomic motion planning," in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 680–685, April 1991.
- [26] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *The American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [27] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.
- [28] H. J. Sussmann and G. Tang, "Shortest paths for the reeds-shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control," Technical Report SYCON-91-10, Department of Mathematics, Rutgers University, September 1991.
- [29] P. Souères and J.-P. Laumond, "Shortest paths synthesis for a car-like robot," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 41, no. 5, pp. 672–688, 1996.
- [30] P. Robuffo Giordano, M. Vendittelli, J.-P. Laumond, and P. Souères, "Nonholonomic distance to polygonal obstacles for a car-like robot of polygonal shape," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 1040–1047, 2006.
- [31] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.
- [32] D. J. Walton and D. S. Meek, "A controlled clothoid spline," *Computers and Graphics (Pergamon)*, vol. 29, no. 3, pp. 353–363, 2005.
- [33] A. Piazzi, C. G. Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic G2-splines for the iterative steering of vision-based autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 27–36, 2002.
- [34] S. Glaser, B. Vanholme, S. Mammarr, D. Gruyer, and L. Nouvelière, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 589–606, 2010.
- [35] J.-W. Lee and B. Litkouhi, "A unified framework of the automated lane centering/changing control for motion smoothness adaptation," in *Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems, ITSC 2012*, pp. 282–287, September 2012.
- [36] L. Z. Wang, K. T. Miura, E. Nakamae, T. Yamamoto, and T. J. Wang, "An approximation approach of the clothoid curve defined in the interval $[0, \pi/2]$ and its offset by free-form curves," *CAD Computer Aided Design*, vol. 33, no. 14, pp. 1049–1058, 2001.
- [37] J. Sánchez-Reyes and J. M. Chacón, "Polynomial approximation to clothoids via s-power series," *CAD Computer Aided Design*, vol. 35, no. 14, pp. 1305–1313, 2003.
- [38] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Söderkvist, "Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 167–172, 2010.
- [39] K. Lee, D. Kim, W. Chung, H. W. Chang, and P. Yoon, "Car parking control using a trajectory tracking controller," in *Proceedings of the 2006 SICE-ICASE International Joint Conference*, pp. 2058–2063, October 2006.
- [40] D. Kim, W. Chung, and S. Park, "Practical motion planning for car-parking control in narrow environment," *IET Control Theory and Applications*, vol. 4, no. 1, pp. 129–139, 2010.
- [41] B. Müller, J. Deutscher, and S. Grodde, "Continuous curvature trajectory design and feedforward control for parking a car," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 541–553, 2007.
- [42] B. Mirtich and J. Canny, "Using skeletons for nonholonomic path planning among obstacles," in *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pp. 2533–2540, May 1992.
- [43] P. Zips, M. Böck, and A. Kugi, "Optimisation based path planning for car parking in narrow environments," *Robotics and Autonomous Systems*, 2015.
- [44] A. Yershova, L. Jaillet, T. Simeon, and S. M. LaValle, "Dynamic-domain RRTs: efficient exploration by controlling the sampling domain," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3856–3861, Barcelona, Spain, April 2005.
- [45] D. Kiss and D. Papp, "Effective navigation in narrow areas: A planning method for autonomous cars," in *Proceedings of the 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*, pp. 000423–000430, Herľany, Slovakia, January 2017.
- [46] D. Kiss and G. Tevesz, "A steering method for the kinematic car using C*CS paths," in *Proceedings of the 15th International Carpathian Control Conference, ICC 2014*, pp. 227–232, May 2014.
- [47] Á. Nagy, G. Csorvási, and D. Kiss, "Path planning and control of differential and car-like robots in narrow environments," in

Proceedings of the 13th IEEE International Symposium on Applied Machine Intelligence and Informatics, SAMI 2015, pp. 103–108, January 2015.

- [48] G. Csorvási, Á. Nagy, and D. Kiss, “RTR+C*CS: An effective geometric planner for car-like robots,” in *Proceedings of the 2015 16th IEEE International Carpathian Control Conference, ICC 2015*, pp. 85–90, May 2015.
- [49] D. Kiss, G. Csorvási, and Á. Nagy, “A planning method to obtain good quality paths for autonomous cars,” in *Proceedings of the 4th Eastern European Regional Conference on the Engineering of Computer-Based Systems, ECBS-EERC 2015*, pp. 104–110, August 2015.
- [50] D. K. Wilde, “Computing clothoid segments for trajectory generation,” in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 2440–2445, October 2009.
- [51] J.-D. Boissonmat, A. Cerezo, and J. Leblond, “A note on shortest paths in the plane subject to a constraint on the derivative of the curvature,” Technical Report RR-2160, Inst. Nat. de Recherche en Informatique en Automatique, 1994.
- [52] S. Pandi, F. H. P. Fitzek, C. Lehmann et al., “Joint design of communication and control for connected cars in 5G communication systems,” in *Proceedings of the 2016 IEEE Globecom Workshops, GC Wkshps 2016*, December 2016.
- [53] H. Charaf, G. Harsányi, A. Poppe et al., “BME VIK annual research report on electrical engineering and computer science 2016,” *Periodica Polytechnica Electrical Engineering and Computer Science*, vol. 61, no. 2, pp. 83–115, 2017.

Research Article

Bifurcation of Lane Change and Control on Highway for Tractor-Semitrailer under Rainy Weather

Tao Peng,¹ Zhiwei Guan,¹ Ronghui Zhang,² Jinsong Dong,³ Kening Li,⁴ and Hongguo Xu⁵

¹College of Automobile and Transportation, Tianjin University of Technology and Education, Tianjin 300222, China

²Guangdong Key Laboratory of Intelligent Transportation System and Research Center of Intelligent Transportation System, School of Engineering, Sun Yat-sen University, Guangzhou 510275, China

³Key Laboratory of Operation Safety Technology on Transport Vehicles, Research Institute of Highway Ministry of Transport, Beijing 100088, China

⁴State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiaotong University, Shanghai 200240, China

⁵College of Transportation, Jilin University, Changchun 130025, China

Correspondence should be addressed to Ronghui Zhang; zrh1981819@126.com

Received 7 July 2017; Accepted 5 September 2017; Published 19 October 2017

Academic Editor: Cheng S. Chin

Copyright © 2017 Tao Peng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new method is proposed for analyzing the nonlinear dynamics and stability in lane changes on highways for tractor-semitrailer under rainy weather. Unlike most of the literature associated with a simulated linear dynamic model for tractor-semitrailers steady steering on dry road, a verified 5DOF mechanical model with nonlinear tire based on vehicle test was used in the lane change simulation on low adhesion coefficient road. According to Jacobian matrix eigenvalues of the vehicle model, bifurcations of steady steering and sinusoidal steering on highways under rainy weather were investigated using a numerical method. Furthermore, based on feedback linearization theory, taking the tractor yaw rate and joint angle as control objects, a feedback linearization controller combined with AFS and DYC was established. The numerical simulation results reveal that Hopf bifurcations are identified in steady and sinusoidal steering conditions, which translate into an oscillatory behavior leading to instability. And simulations of urgent step and single-lane change in high velocity show that the designed controller has good effects on eliminating bifurcations and improving lateral stability of tractor-semitrailer, during lane changing on highway under rainy weather. It is a valuable reference for safety design of tractor-semitrailers to improve the traffic safety with driver-vehicle-road closed-loop system.

1. Introduction

Tractor-semitrailer, as an articulated vehicle system, has complex nonlinear dynamic characteristics. Vehicle dynamics bifurcation mechanisms and driving stability have basically been confirmed based on nonlinear theory and tire mechanics. Early works on this subject have established three degrees of freedom (DOF) dynamic model with simplified nonlinear tire formula, which are essential for a clear understanding of the system's nonlinear stability [1, 2]. Due to the special articulated structure, jackknife and coordination stability have drawn wide attention. Dunn et al. rigorously developed a 15-state model using Lagrange's method and studied in detail the effects on jackknife stability when using brakes

under different road adhesion coefficients [3–5]. To protect the vehicle from spinning and realize improved cornering performance, van de Molengraft-Luijten et al. employed the Routh-Hurwitz stability criterion and bifurcation theory method to get a system equilibrium approximation near the zero value for the forward speed and front wheel steering angle, which revealed the vehicle's instability under high speeds and substantial steering [6, 7]. Similarly, Ding et al. explained some instability phenomena of the vehicle system such as jackknifing, sideslip, and spinning by correlating them with the behavior in the neighborhood of unstable fixed points based on analysis of eigenvectors, phase trajectories, and status of lateral tire force saturation [8]. Sadri and Wu investigated Lyapunov concept exponents and applications to

analyze the stability for the nonlinear vehicle in plane motion with a third-order polynomial tire model [9]. In addition, the existing literatures focus on bifurcation application analysis for the stability assessment of a complex railway vehicle model [10, 11].

Moreover, vehicle/driver and vehicle/driver/road cooperation systems and influence parameters were introduced for tractor-semitrailer nonlinearity investigations [12–15]. To study vehicle Hopf bifurcation and chaos characteristics and variation in pilot model parameters and to discuss the qualitative motion behavior near the critical speed, Rossa et al. proposed a simple 3DOF closed-loop vehicle/driver model [16, 17]. Li et al. presented a nonlinear vehicle-road coupled model and analyzed the dynamic behaviors of a nonlinear system using a numerical integration method [18]. With a closed-loop system of articulated heavy vehicles with driver steering control, Liu et al. employed an integration method to derive an analytical periodic solution of the system in the neighborhood of the critical speed and analyzed supercritical and subcritical Hopf bifurcations [19]. Koglbauer et al. paid close attention to drivers' workload and effort in different road conditions [20]. Bie et al. proposes a weather factor model built by introducing weather factors to free flow speed, capacity, and critical density and plugged it into traffic control [21]. Tang et al. investigated the road condition and driver characteristics on the lane change driving trajectory and traffic safety, such as lane number, driver's time delay, and perception ability effect [22–27]. Related researches are helpful to master the interactions of vehicle, driver, and work environment.

Furthermore, with the rapid development of control theory, a variety of methods are available in the tractor-semitrailer stability control system and product, such as ABS (Antilock Brake System) and ESC (Electronic Stability Control). By an optimal control approach for driver model, Liu studied the stability of a time delayed dynamical tractor-semitrailer, using a numerical method by computing the eigenvalues near the imaginary axis [13]. Lin et al. proposed optimal linear quadratic control algorithm to improve the yaw stability of a tractor semitrailer using active semitrailer steering [28]. Palkovics and El-Gindy applied RLQR/ H_∞ approach to ensure the vehicle's performance in the presence of parametric uncertainties and discussed the influence of different control strategies on the vehicle's directional and roll stability during severe path-follow lane change manoeuvre [29]. Chang et al. designed a H_∞ loop shaping robust controller robust of tractor-semitrailer, which provides a novel mean or method for explore tracking control problem of the tractor-semitrailer [30]. Liang et al. proposed a specified fuzzy sliding mode controller to guarantee the robust control in the presence of system uncertainties [31]. By active wheel braking, Takenaga et al. proposed a novel fuzzy logic based yaw moment controllers to track the reference yaw rate of the tractor and the hitch angle and demonstrated the robust and effects in stabilizing the severe instabilities such as jackknife and trailer oscillation in the chosen simulation scenarios [32–34]. Zong et al. applied a multiobjective stability control algorithm to improve the vehicle stability of a tractor semitrailer by using differential braking [35]. Recently,

more and more people pay attention to intelligent vehicle based on Cooperative Vehicle Infrastructure System [36–43]. Besides, vehicle control and system model was combined with transportation simulation [44–52].

Many of the conclusions of the classical literature play a positive role in vehicle handing stability. However, there are still some limitations, especially in particular research backgrounds. First, it is clear that the vehicle instability in critical conditions is mainly due to the nonlinearity of tires, but most of traditional vehicle dynamics models are deduced by simplified tire formulas. The convenient calculations still deviate greatly from reality. Second, previous research on vehicles' characteristics has concentrated on steady steering [53–56]. However, lane changes and overtaking by heavy trucks on highways, as a frequent driving behavior, could gain more attention in the near future. Furthermore, most of the traditional efforts have focused on vehicle stability on common road [57, 58]. In fact, rainy weather and lower adhesion coefficient should be taken into consideration, especially for heavy trucks.

Tractor-semitrailer, as a very important and highly effective means of freight transportation, is more dangerous than general passenger cars because of their articulated structure, large size, and difficult handling. Accidents involving tractor-semitrailers tend to cause serious casualties and losses. Therefore, active safety and traffic transportation researchers have paid more attention to tractor-semitrailers' driving stability in recent years. In particular, lane changing is a common driving behavior on highways, but high speeds and large-scale continuous steering can easily cause stability losses, such as sideslips and nonlinear oscillatory, for tractor-semitrailers. Last but not least, considering the large-scale rainy season in the south of our country, vehicles driving on lower adhesion efficient road and harsh environment are common, which cause increased instability and traffic accidents [59–63]. Therefore, nonlinear analysis and control of tractor-semitrailers' lane changing on highways under rainy weather have important practical significance in the vehicle dynamics research field and provide a valuable reference for active safety design.

This work deals with the nonlinear dynamics and control of tractor-semitrailers changing lanes on highways under rainy weather, which is a highly relevant real-life situation. In order to master the essence of stability and instability mechanism of tractor-semitrailer changing lanes at high speed, Rough-Hurwitz stability criterion is developed to analyze vehicle transient stability, and the vehicle bifurcation characteristics under steady and transient steering are studied based on nonlinear theory. In view of the vehicle's instability mechanism, feedback linearization control model is deduced and controller combined with AFS (Active Front Steering) and DYC (Direct Yaw Control) was established and presented. It is significant to provide a valuable reference for safety design of tractor-semitrailers, to improve the traffic safety of driver-vehicle-road closed-loop system.

This text is structured as follows: after a brief description of the vehicle model and validation in Section 2, Section 3 analyzes bifurcations with steady steering and sinusoidal steering to simulate the main process of lane changing.

Section 4 deals with an investigation of feedback linearization control for lane changing. Section 5 proposes a conclusion and further work. Conflicts of interest and acknowledgements are dealt with at the end of the article.

2. Vehicle System Model

By defining ground coordinate system XOY , Figure 1 illustrates the plane motion force diagrams of tractor and semitrailer vehicle coordinate systems $x_t o_t y_t$ and $x_s o_s y_s$. The x_t and x_s axes, respectively, point forward and lie both in the ground plane and in the plane normal to the ground of tractor and semitrailer. The y_t and y_s axes point to the left. Positions of o_t and o_s are, respectively, centroid of the vehicles. As motion variables we define the longitudinal velocities v_{xt} and v_{xs} , lateral velocities v_{yt} and v_{ys} of tractor and semitrailer centroids, the slip angles β_t and β_s are respectively introduced by longitudinal and lateral velocities of tractor and semitrailer, the yaw angles φ_t and φ_s are respectively included angles between longitudinal axes and x_s and x_t axes of tractor and semitrailer.

As shown in Figure 1, when front wheels are steering, the vehicle is not only moving along the longitudinal direction, but also introduced the lateral motion around the steering center, and the yaw motion caused by dynamic centrifugal force at centroid. Thus, 5DOF model involves steering system motion, tractor, and semitrailer's longitudinal and lateral motions, and yaw motions could be depicted in Figure 1. During the process, forces acting on the tires provide centripetal force of vehicle motion. In common situation, the centrifugal force is balanced with the centripetal force, and the plane motions of the vehicle will be in a regular way. Otherwise, if the centrifugal force is greater than the centripetal force, the vehicle will sideslip.

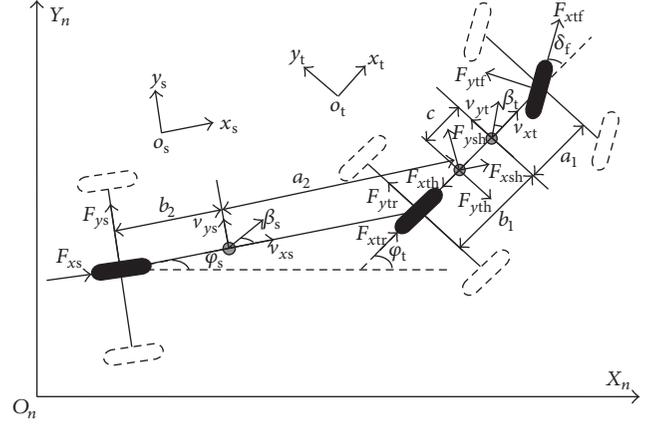
2.1. Vehicle Dynamics Model. Considering the research vehicle is moving on highway when steering without active braking and the road is flat, but wet and slippery under rainy weather, vertical, roll, and pitch motion can be ignored in this situation. Thus, the established vehicle model could be simplified as 5DOF model. The corresponding equations of the vehicle model are as follows:

(1) Tractor,

$$\begin{aligned} m_t (\dot{v}_{xt} + v_{yt} \dot{\varphi}_t) &= F_{xtf} + F_{xtr} - F_{xth}, \\ m_t (\dot{v}_{yt} + v_{xt} \dot{\varphi}_t) &= F_{ytf} + F_{ytr} - F_{yth}, \\ I_{zt} \ddot{\varphi}_t &= a_t F_{ytf} - b_t F_{ytr} + c F_{yth}. \end{aligned} \quad (1)$$

(2) Semitrailer,

$$\begin{aligned} m_s (\dot{v}_{xs} + v_{ys} \dot{\varphi}_s) &= F_{xs} + F_{xsh}, \\ m_s (\dot{v}_{ys} + v_{xs} \dot{\varphi}_s) &= F_{ys} + F_{ysh}, \\ I_{zs} \ddot{\varphi}_s &= -b_s F_{ys} + a_s F_{ysh}. \end{aligned} \quad (2)$$



- F_{xtf} : longitudinal force of tractor front wheels
- F_{ytf} : lateral force of tractor front wheels
- F_{xtr} : longitudinal force of tractor rear wheels
- F_{ytr} : lateral force of tractor rear wheels
- F_{xs} : longitudinal force of semitrailer wheels
- F_{ys} : lateral force of semitrailer wheels
- $F_{xt(s)h}$: longitudinal force of tractor (semitrailer) on saddle
- $F_{yt(s)h}$: lateral force of tractor (semitrailer) on saddle
- δ_f : front wheel steering angle
- $\beta_{t(s)}$: slip angle of tractor (semitrailer) centroid
- $v_{xt(s)}$: longitudinal velocity of tractor (semitrailer) centroid
- $v_{yt(s)}$: lateral velocity of tractor (semitrailer) centroid
- $\varphi_{t(s)}$: yaw angle of tractor (semitrailer)
- a_t : distance between tractor centroid and front axle
- b_t : distance between tractor centroid and rear axle
- c : distance between tractor centroid and saddle point
- a_s : distance between semitrailer centroid and front axle
- b_s : distance between semitrailer centroid and rear axle

FIGURE 1: Schematic diagram of tractor-semitrailer plane motion.

(3) Steering system [64],

$$I_w \ddot{\delta}_f + C_w \dot{\delta}_f = K_w \left(\frac{\delta_{sw}}{i} - \delta_f \right) - \varepsilon F_{ytf}. \quad (3)$$

Meanwhile, conditions of saddle balance approximately satisfy

$$\begin{aligned} v_{xt} &= v_{xs}, \\ v_{ys} &= v_{xt} \theta + v_{yt} - (c + a_s) \dot{\varphi}_t + a_s \dot{\theta}, \\ F_{xth} &= F_{xsh}, \\ F_{yth} &= F_{ysh}. \end{aligned} \quad (4)$$

Taking a domestic tractor-semitrailer as an example, relevant vehicle parameters and symbols are listed in Table 1. Here, the mass and structure values of the vehicle refer to test data, inertia, and stiffness and the other performance data are selected based on database of Trucksim.

2.2. Nonlinear Tire Model. In addition to supporting the vehicle steering dynamics, the tires provide the lateral forces necessary to change the speed and direction of the vehicle.

TABLE 1: Tractor-semitrailer parameters and symbols.

Symbol	Parameter name	Value	Unit
m_t	Tractor mass	5800	kg
m_s	Semitrailer mass	19080	kg
a_t	Distance between tractor centroid and front axle	2.0	m
b_t	Distance between tractor centroid and rear axle	1.4	m
c	Distance between tractor centroid and saddle point	0.8	m
a_s	Distance between semitrailer centroid and saddle point	2.0	m
b_s	Distance between semitrailer centroid and rear axle	1.4	m
I_{zt}	Tractor yaw moment of inertia	9737	kg·m ²
I_{zs}	Semitrailer yaw moment of inertia	138437	kg·m ²
I_w	Moment of inertia of steering wheel around spin	60	kg·m ²
C_w	Damping coefficient of steering wheel around spin	10250	—
K_w	Steering system stiffness	2000	Nm/rad
ε	Tractor front tire pneumatic trail	0.01	m
i	Steering radio	28	—
v_{xt}	Tractor longitudinal velocity	—	m/s
v_{xs}	Semitrailer longitudinal velocity	—	m/s
v_{yt}	Tractor lateral velocity	—	m/s
v_{ys}	Semitrailer lateral velocity	—	m/s
φ_t	Tractor yaw angle	—	rad/s
φ_s	Semitrailer yaw angle	—	rad/s
θ	Articulated angle	—	rad
δ_{sw}	Tractor steering wheel angle	—	rad
δ_f	Tractor front wheel steering angle	—	rad
F_{xtf}	Tractor front wheel longitudinal force	—	N
F_{xtr}	Tractor rear wheel longitudinal force	—	N
F_{ytf}	Tractor front wheel lateral force	—	N
F_{ytr}	Tractor rear wheel lateral force	—	N
F_{xs}	Semitrailer wheel longitudinal force	—	N
F_{ys}	Semitrailer front wheel lateral force	—	N
F_{xth}	Longitudinal force from saddle to tractor	—	N
F_{yth}	Lateral force from saddle to tractor	—	N
F_{xsh}	Longitudinal force from saddle to semitrailer	—	N
F_{ysh}	Lateral force from saddle to semitrailer	—	N

There exist many tire models to describe tire behaviors beyond the linear region. One model commonly used in vehicle dynamics simulations was developed by Pacejka of the Delft University of Technology [64, 65]. The lateral slip angle α generates a lateral force, F_y , at the tire-ground interface, which could be expressed as a magic formula:

$$F_y = D \sin(C \operatorname{atan}\{B\alpha - E[B\alpha - \operatorname{atan}(B\alpha)]\}), \quad (5)$$

where α is the tire slip angle and B , C , D , and E are, respectively, the stiffness factor, shape factor, peak factor, and curvature factor.

As we know, sideslip instability of a tractor-semitrailer is caused by exceeding the lateral tire force saturation. Lower lateral tire force saturation arises from lower adhesion coefficient under rainy weather, which easily causes severe

TABLE 2: Values of B , C , D , and E for a nonlinear tire.

Tire position	Tire number	B	C	D	E
Tractor front wheel	2	3.32	1.2	-12987.6	-1.45
Tractor rear wheel	4	11.35	1.2	-16327.2	-1.45
Semitrailer wheel	8	19.62	1.2	-25792.8	-1.45

nonlinear performance of the vehicle. Normally, on wet and slippery road, the absolute value of peak factor D must be lower and the stiffness and shape factors (B and C) are higher than that on general roads. Relevant parameters valued as all wheels of each axis on wet road ($\mu = 0.3$) for the tires are listed in Table 2 [66].



FIGURE 2: Image of vehicle test environment.

These slip angles may be obtained using the following handing relations:

$$\begin{aligned}\alpha_{tf} &= \arctan\left(\frac{v_{yt} + a_t\omega_t}{v_{xt}}\right) - \delta_f, \\ \alpha_{tr} &= \arctan\left(\frac{v_{yt} - b_t\omega_t}{v_{xt}}\right), \\ \alpha_s &= \arctan\left(\frac{v_{ys} - b_s\omega_s}{v_{xs}}\right),\end{aligned}\quad (6)$$

where α_{tf} , α_{tr} , and α_s are, respectively, the tractor front and rear wheel and the semitrailer wheel's slip angle, v_{yt} and v_{ys} are, respectively, the tractor and semitrailer's lateral velocity, v_{xt} and v_{xs} are, respectively, the tractor and semitrailer's longitudinal velocity, a_t is the distance between tractor centroid and front axle, b_t is distance between tractor centroid and rear axle, b_s is the distance between semitrailer centroid and rear axle, and ω_t and ω_s are, respectively, the tractor and semitrailer's yaw rate.

2.3. Vehicle Dynamics Model Validation. The double-lane change is deemed the general method of overtaking. The whole process can be divided into five steps: straight driving in original lane, approximately sinusoidal steering, straight driving in the other lane, and approximately sinusoidal steering and steer returning in original lane. Thus, the tractor-semitrailer comes under complex and severe dynamics, with changes in lane position, large-scale continuous steering, and high speed.

To validate the vehicle dynamics model, applying a domestic tractor-semitrailer, we conducted a physical double-lane change experiment on an even and wet road, just as a tractor-semitrailer overtaking under rainy weather on highway. Tests were carried out at a professional test site in the Key Laboratory of Operation Safety Technology for Transport Vehicles, at the Research Institute of the Highway Ministry of Transport, Beijing, China. Details of the test environment are shown in Figure 2.

The tested vehicle was driven along the double-change channel on an even and wet road, with its velocity kept to about 60 km/h. The integrated test system mainly involved

a VBOX III for tractor, a VBOX II and an inertial sensor for semitrailer, a vehicle-handling force and angle meter for vehicle steering, a data collection device, and a power supply device, whose function was to measure the steering wheel angle input and the output vehicle motion parameters, including velocity, acceleration, and angular velocity.

Setting the simulation inputs the same as test velocities and steering wheel angles, in addition, the time step is 0.05 s in accord with test sampling frequency 20 Hz. Furthermore, the orientations of simulated output variables keep consistent with test.

Under the same conditions, tractor and semitrailer lateral accelerations, yaw rates, and joint angle were obtained using vehicle testing and Matlab simulation and compared, as shown in Figure 3.

Clearly, the vehicle's lateral and yaw dynamics in Figure 3 exhibit good consistency between the simulation and the vehicle test. The validation of the model basis is given, using limited data. Moreover, according to the test results, roll angles of the tractor and semitrailer shown in Figure 3(g) are very small (less than 1 deg), so vehicle roll motions on wet and slippery road could be ignored reasonably. Therefore, it can be conservatively concluded that the established 5DOF nonlinear dynamics model closely reflects the physical properties.

3. Bifurcation of Steering

Bifurcation theory is the mathematical study of changes in the qualitative or topological structure of a given family, such as the integral curves of a family of vector fields, and the solutions of a family of differential equations. Most commonly applied to the mathematical study of dynamical systems, a bifurcation occurs when a small smooth change made to the parameter values (the bifurcation parameters) of a system causes a sudden "qualitative" or topological change in its behavior. The name "bifurcation" was first introduced by Henri Poincaré in 1885 in the first paper in mathematics showing such a behavior [67, 68].

In view of the lane change handling steps, steering at a high speed on a highway under rainy weather is the primary extensive condition within the whole process. Considering the specific features of tractor-semitrailer, the vehicle nonlinear dynamics with large-scale steering on wet road are more complex than single vehicle under the normal situation and are analyzed in the following subsections.

3.1. Vehicle Model Transformation. In order to efficiently analyze tractor-semitrailer's bifurcation characteristics in typical work conditions and simplify calculation process, steering wheel is not applied as the input of steering, front wheel instead. Based on the idea, the vehicle system is transformed as

$$\dot{\mathbf{X}} = f(\mathbf{X}, \delta_f), \quad (7)$$

where $\mathbf{X} = (v_{xt}, v_{yt}, \dot{\varphi}_t, \dot{\theta}, \theta)^T$.

In the case of the vehicle steering motion, suppose the vector $\mathbf{X}^e = (v_{xt}^e, v_{yt}^e, \dot{\varphi}_t^e, \dot{\theta}^e, \theta^e)^T$ is an equilibrium point for the system described in (7). To assess the system behavior

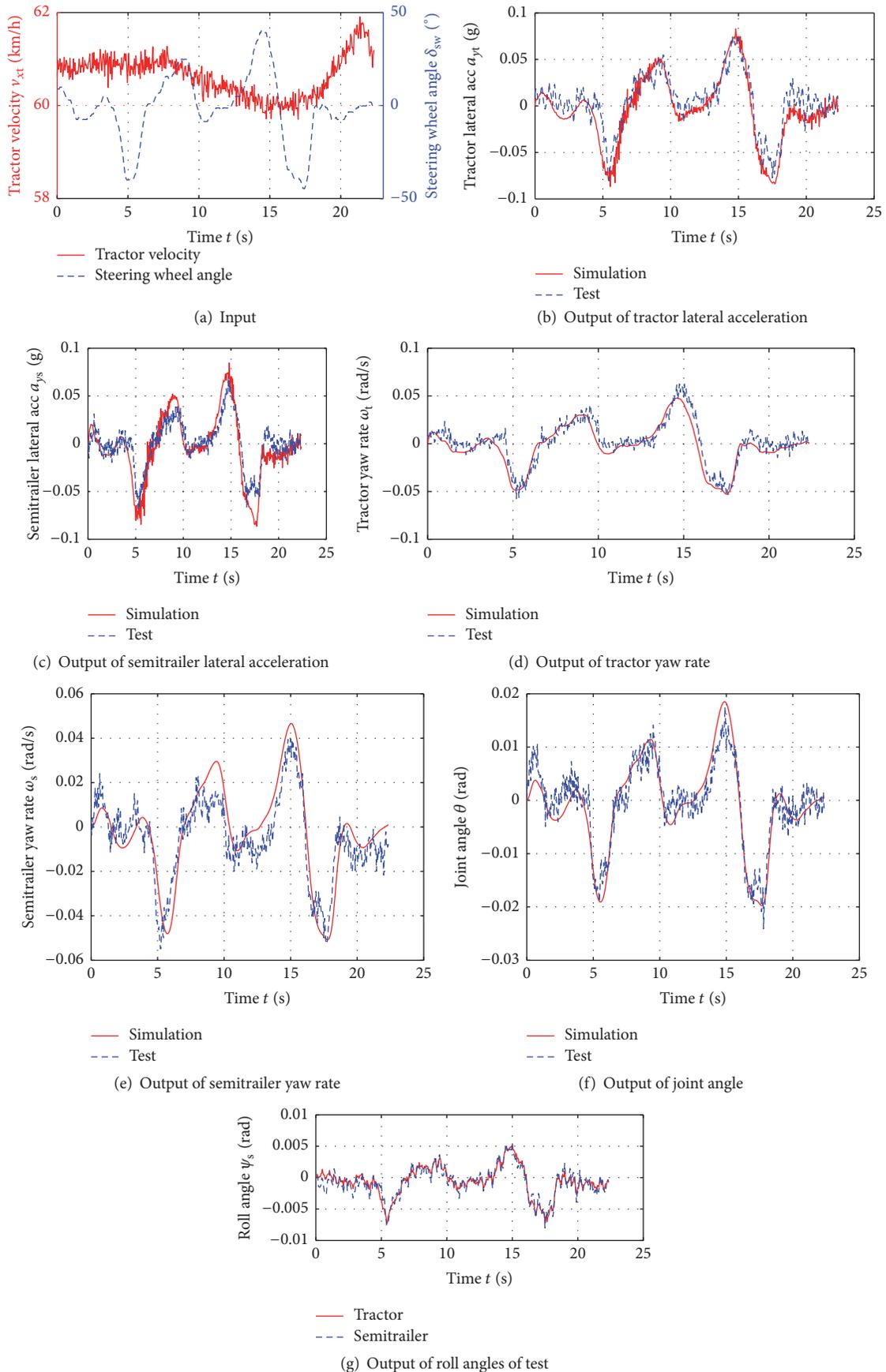


FIGURE 3: Comparison of simulation and vehicle test.

TABLE 3: Calculated eigenvalues of Jacobian matrix A under various front wheel steering angles.

δ_f/rad	Eig. 1		Eig. 2		Eig. 3		Eig. 4		Eig. 5	
	Re	Im	Re	Im	Re	Im	Re	Im	Re	Im
0.01	-1.8135	3.3965	-1.8135	-3.3965	-1.3298	2.1452	-1.3298	-2.1452	-0.0001	0
0.02	-1.7838	3.3517	-1.7838	-3.3517	-1.3154	2.1192	-1.3154	-2.1192	-0.0006	0
0.03	-1.7011	3.2258	-1.7011	-3.2258	-1.2801	2.0400	-1.2801	-2.0400	-0.0014	0
0.04	-1.4962	2.9357	-1.4962	-2.9357	-1.2203	1.8070	-1.2203	-1.8070	-0.0032	0
0.049	-0.8385	2.2125	-0.8385	-2.2125	0	0.1102	0	-0.1102	-2.1906	0

after stability loss, (7) is expanded into a multidimensional Taylor series in x_i around \mathbf{X}^e up to order 3, yielding

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{F}(\mathbf{X}, \delta_f) + \mathbf{O}(|\mathbf{X}|^4), \quad (8)$$

where A is the Jacobian matrix of the nonlinear function $f(\mathbf{X}, \delta_f)$, computed at the equilibrium point \mathbf{X}^e and $F(\mathbf{X}, \delta_f)$ contains nonlinear terms of order 3 at \mathbf{X}^e :

$$A = D_{\mathbf{x}}f(\mathbf{X}, \delta_f)|_{(v_{xt}^e, v_{yt}^e, \phi_f^e, \theta^e, \theta^e)^T}. \quad (9)$$

The fifth-order characteristic equation of Jacobian matrix A is as follows:

$$\lambda^5 + c_1\lambda^4 + c_2\lambda^3 + c_3\lambda^2 + c_4\lambda + c_5 = 0. \quad (10)$$

According to Hurwitz stability criterion, if and only if all the roots of Jacobian matrix A have strictly negative real part, the system is stable. The criterion identifies the conditions when the poles of a polynomial cross into the right hand half plane and hence would be considered as unstable in control engineering [67–69].

3.2. Bifurcation of Steady Steering. Steady steering not only is a typical working condition, for exhibiting vehicle steady characteristics, but also is the basic of transient. Good steady characteristics predict better transient performance to a certain extent.

With initial longitudinal velocity 20 m/s, the calculated five eigenvalues (Re represents the real part and Im represents the imaginary part) of Jacobian matrix A with front wheel steering angle varying in the range [0.01, 0.049] rad are listed in Table 3.

As shown in Table 3, when the steering angle is less than 0.049 rad, there are five eigenvalues with negative real parts. According to the nonlinear theory, in these cases, the vehicle system is stable and the equilibrium points are called nodal points. Once the steering angle reaches 0.049 rad, two pairs of imaginary eigenvalues with zero real parts appear, which means that a Hopf bifurcation may exist [69].

According to [68], there is a pair of pure imaginary conjugate eigenvalues and the other eigenvalues have nonzero real part; thus, the corresponding equilibrium point is non-hyperbolic. In addition, it is clearly seen that the real part of

the eigenvalues varies as the steering angle increases, and the transverse condition is satisfied as follows:

$$\begin{aligned} \left. \frac{d \operatorname{Re}(\lambda)}{d\delta_f} \right|_{\delta_f=0.049} &= \operatorname{Re} \left(\left. \frac{d(\lambda)}{d\delta_f} \right|_{\delta_f=0.049} \right) \\ &= \frac{0 - (-1.2203)}{0.049 - 0.04} = 135.59 \neq 0. \end{aligned} \quad (11)$$

Therefore, it is proved that the Hopf bifurcation is taking place at the equilibrium point when $\delta_f = 0.049$ rad.

The equilibrium point could be calculated with the vehicle model by Matlab. Input parameters are $\delta_f = 0.049$ rad and initial $v_x = 20$ m/s; the system will be stable within a limited period of time; then the equilibrium point could be obtained according to the stable variables; that is, $(\mathbf{X}^e, \delta_f) = (-1.3162, 0.0761, 0, 0.0576, 19.86, 0.049)$.

Once Hopf bifurcation is taking place, under reasonably generic assumptions about the dynamical system, as steering angle slightly rises, a periodic solution called limit cycle branches from the fixed point [67, 68]. When the steering angle is 0.05 rad, limit circles of tractor's velocities with time are shown in Figure 4.

It can be seen from Figure 4, as time increases, a large-scale periodic vibration of the tractor's velocities, which represents a periodic solution of vehicle system. So there is limit circle when the steering angle is 0.05 rad, after Hopf bifurcation has taken place, which shows large range conversion of vehicle longitudinal and lateral kinetic energy. Obviously, the energy conversion could not be realized in reality because of frictional loss and motion constraint between tractor and semitrailer. However, the articulated vehicle's conservation of energy and dynamic trends of stability loss is clearly revealed.

For the 4DOF model with longitudinal velocity changes, the state variable of quasiperiodic vibration shows energy conservation. Moreover, the eigenvalues change from imaginary with negative real part to pure imaginary in accordance with the manifolds, predicting the Hopf bifurcation phenomenon. It is verified that a pair of conjugate roots, first crossing on the imaginary axis, translates into oscillatory behavior leading to instability.

3.3. Bifurcation of Sinusoidal Steering. Sinusoidal steering is introduced to approximately represent the steering process for lane changing. Unlike steady steering, the front wheel steering angle is represented as

$$\delta_f = A_s \sin(2\pi ft), \quad (12)$$

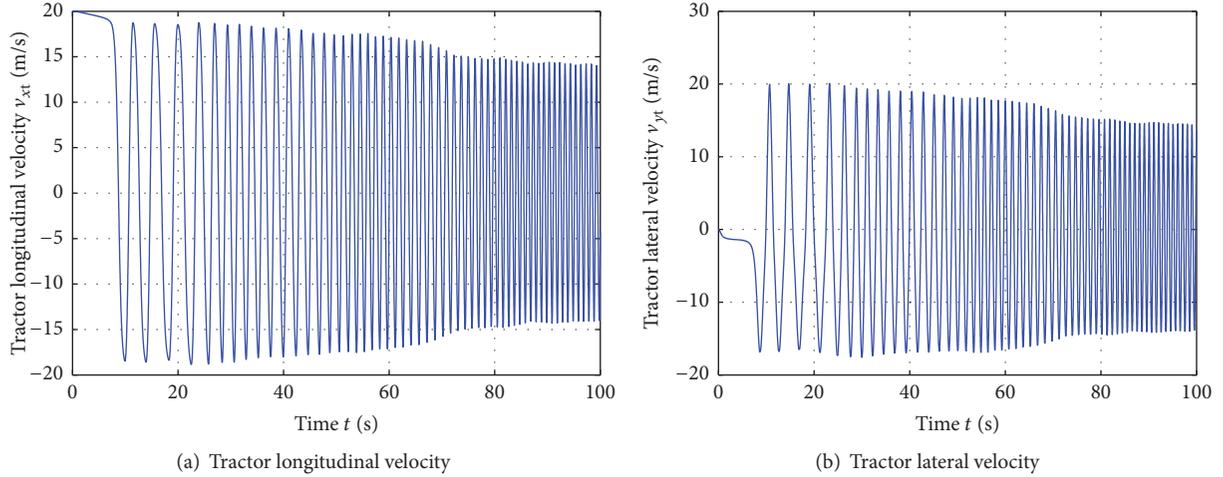


FIGURE 4: Limit circle of tractor's velocities with $\delta_f = 0.05$ rad.

where A_s is the steering amplitude and f is steering frequency. Compared with steady steering, the vehicle with sinusoidal steering is considered a time-changing system. The equilibrium points and corresponding eigenvalues vary with time and the steering angle.

Figure 5 shows the changes in the steering angle, lateral velocity, and calculated eigenvalues when the initial longitudinal velocity is 20 m/s, and the front wheel steering amplitude and frequency are, respectively, 0.06 rad and 0.2 Hz. With $A_s = 0.06$ rad, $f = 0.2$ Hz, the calculated eigenvalues present a double wave shape with sinusoidal steering. According to the tractor lateral velocity, the vehicle system is still stable, although transit values of eigenvalue 1 and eigenvalue 3 are positive or have positive real parts during steering. In addition, when $t > 6.3$ s, all of the eigenvalues are negative as steer returning takes place and eigenvalue 5 is approximately equal to 0. Therefore, it can be concluded that the negative eigenvalue is sufficient, but not necessary, for judging stability. Furthermore, comparing sinusoidal and steady steering, when the front wheel angle is greater than 0.05 rad, the vehicle system with steady steering loses stability, but that with sinusoidal steering remains stable. Specifically, vehicle stability is influenced by the frequency and the transient state.

Figure 6 shows, as the amplitude is increased from 0.06 rad to 0.07 rad, the changes in the steering angle, lateral velocity, and calculated eigenvalues.

It can be seen from Figure 6 that, as the steering amplitude increases to 0.07 rad, not only transit values of eigenvalue 1 and eigenvalue 3 are positive or have positive real parts during steering, just as $A_s = 0.06$ rad, but also transit values of eigenvalue 5 are positive eigenvalues after steer returning. Meanwhile, there is a pair of conjugate eigenvalues (eigenvalues 1 and 2), approximately lying on the imaginary axis, which indicates the existence of Hopf bifurcation. Figure 7 shows the unstable limit circle of the vehicle typical parameters with continuous sinusoidal steering when $A_s = 0.07$ rad, $f = 0.2$ Hz.

Just as in Figure 4 for the case of steady steering, Figure 7 shows an unstable limit circle of vehicle dynamic motions with critical sinusoidal steering. It clearly shows there are obvious differences in fluctuation range of vehicle's lateral velocity between steady and sinusoidal steering. Besides this, the vehicle's dynamic bifurcation and oscillatory behavior are identical.

4. Bifurcation Control with Feedback Linearization

It is obvious that, with the Jacobian matrix A induced by vehicle's nonlinear model, the eigenvalues could reveal the bifurcation characteristics and stability. Thus, basic ideal is proposed to ensure the eigenvalues with negative real parts, by proper control method.

Feedback Linearization is an approach to nonlinear control design which has attracted a great deal of research interest in recent years. Algebraically transform a nonlinear system into a (fully or partly) linear one, so that linear control techniques can be applied. *This differs from the conventional linearization in that feedback linearization is achieved by exact state transformations and feedback, rather than by linear approximations of the dynamics* [63, 69–72]. Considering the nonlinearity of tractor-semitrailer, based on feedback linearization theory, an integrated controller is introduced to realize vehicle system asymptotically stable and tracking performance in this section.

4.1. Vehicle Control Model. Supposing the tractor longitudinal velocity constant, the vehicle model introduced in Section 2 is deduced as follows:

$$\dot{\mathbf{X}} = f(\mathbf{X}, \delta_{sw}), \quad (13)$$

where $\mathbf{X} = [x_1, x_2, x_3, x_4, x_5, x_6]^T = [v_{yt}, \dot{\phi}_t, \dot{\theta}, \theta, \delta_f, \delta_f]^T$.

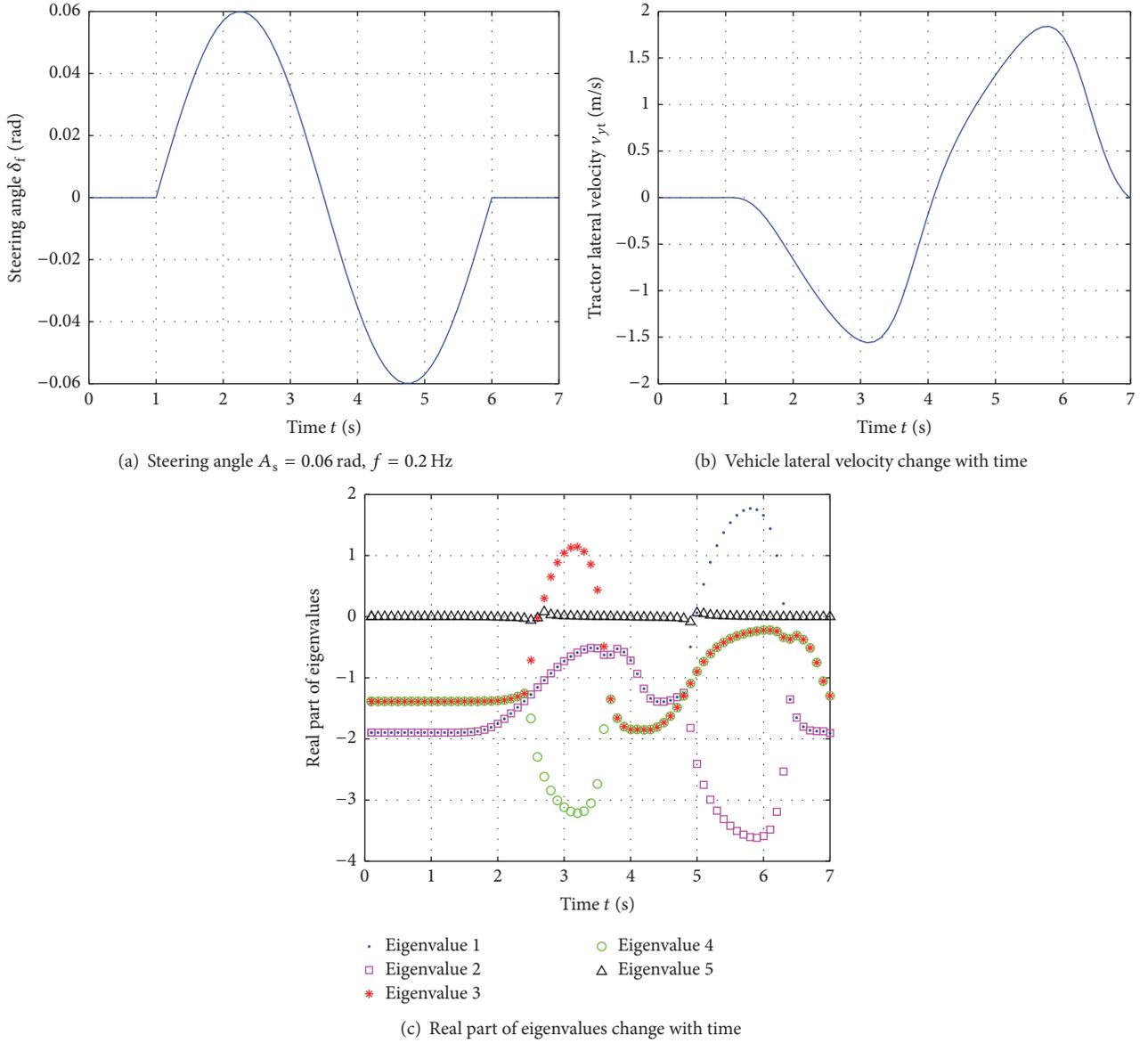


FIGURE 5: Changes in steering angle, lateral velocity, and calculated eigenvalues with $A_s = 0.06$ rad, $f = 0.2$ Hz.

In order to linearize nonlinear systems by feedback linearization method, feedback parameters, as control outputs, not only are controllable, but also ensure linearization. Therefore, setting tractor yaw rate $\dot{\varphi}_t$ and the difference of joint angle and steering angle $\theta - \delta_f$ as control output variables, applying AFS and DYC combined control method, the feedback linearization control model could be described as

$$\begin{aligned} \dot{\mathbf{X}} &= f(\mathbf{X}, \delta_{sw}) + g_1 u_1 + g_2 u_2, \\ y_1 &= h_1(x) = \dot{\varphi}_t, \\ y_2 &= h_2(x) = \theta - \delta_f, \end{aligned} \quad (14)$$

where

$$f(\mathbf{X}, \delta_{sw}) = \begin{bmatrix} a_{11}F_{yf} + a_{12}F_{yr} + a_{13}F_{ys} + a_{14}v_{xt}x_2 \\ a_{21}F_{yf} + a_{22}F_{yr} + a_{23}F_{ys} \\ a_{31}F_{yf} + a_{32}F_{yr} + a_{33}F_{ys} \\ x_3 \\ -\frac{C_w x_5}{I_w} - \frac{K_w x_6}{I_w} - \frac{\varepsilon F_{ytf}}{I_w} + \frac{K_w}{iI_w} \delta_{sw} \end{bmatrix}, \quad (15)$$

$$g_1 = \left[0, 0, 0, 0, -\frac{K_w}{I_w}, 0 \right]^T, \quad (16)$$

$$g_2 = [g_{21}, g_{22}, g_{23}, 0, 0, 0]^T,$$

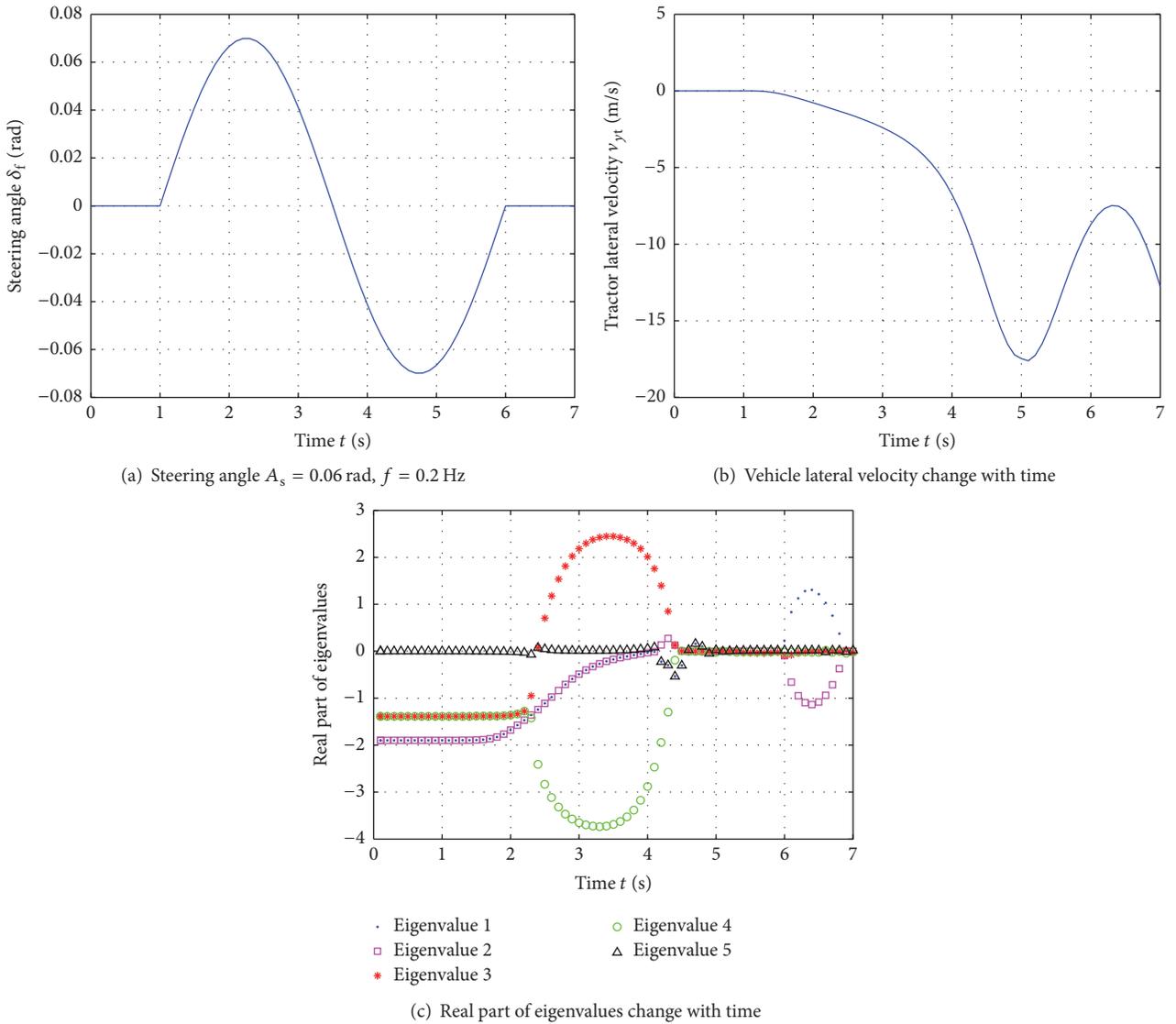


FIGURE 6: Changes in steering angle, lateral velocity, and real parts of calculated eigenvalues with $A_s = 0.07$ rad, $f = 0.2$ Hz.

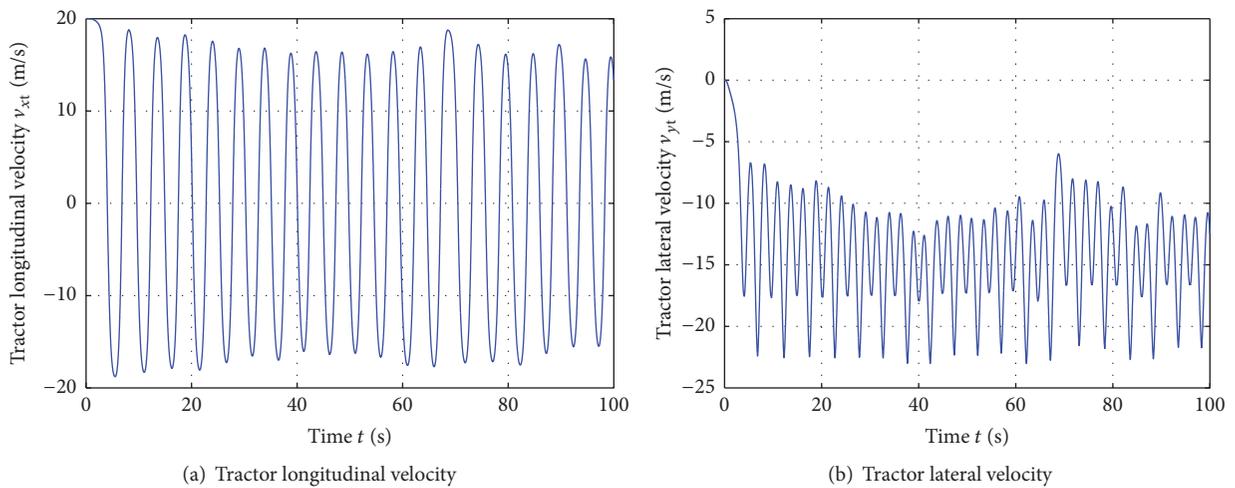


FIGURE 7: Limit circle of vehicle's velocity with $A_s = 0.07$ rad, $f = 0.2$ Hz.

$$\begin{aligned} u_1 &= \delta_{swc}, \\ u_2 &= M_c. \end{aligned} \quad (17)$$

Variables a_{11} , a_{12} , a_{13} , and so on in (15) and g_{21} , g_{22} , and g_{23} in (16) could be expressed by known variables, as follows:

$$\begin{aligned} a_{11} &= \frac{(a_1 + c) cm_2 I_{z2} + I_{z1} I_{z2} + a_2^2 m_2 I_{z1}}{S}, \\ a_{12} &= \frac{(c - b_1) cm_2 I_{z2} + I_{z1} I_{z2} + a_2^2 m_2 I_{z1}}{S}, \\ a_{13} &= \frac{(I_{z2} - a_2 b_2 m_2) I_{z1}}{S}, \\ a_{14} &= \frac{-(a_2^2 I_{z1} + c^2 I_{z2}) m_1 m_2 - (m_1 + m_2) I_{z1} I_{z2}}{S}, \\ a_{21} &= \frac{a_1 a_2^2 m_1 m_2 + (a_1 m_1 + a_1 m_2 + cm_2) I_{z2} - a_2 m_2 I_{z1}}{S}, \\ a_{22} &= \frac{(cm_2 - b_1 m_1 - b_1 m_2) I_{z2} - a_2^2 b_1 m_1 m_2}{S}, \\ a_{23} &= \frac{cm_1 (a_2 b_2 m_2 - I_{z2})}{S}, \\ a_{31} &= \frac{a_1 a_2 (c + a_2) m_1 m_2 + (a_1 m_1 + a_1 m_2 + cm_2) I_{z2} - a_2 m_2 I_{z1}}{S}, \\ a_{32} &= \frac{-a_2 b_1 (c + a_2) m_1 m_2 + (cm_2 - b_1 m_2 - b_1 m_1) I_{z2} - a_2 m_2 I_{z1}}{S}, \\ a_{33} &= \frac{(a_2 I_{z1} - c I_{z2}) m_1 + b_2 [(m_1 + m_2) I_{z1} + c (c + a_2) m_1 m_2]}{S}, \\ g_{21} &= \frac{(a_2 I_{z1} - c I_{z2}) m_2}{S}, \\ g_{22} &= \frac{-(a_2 + c) a_2 m_1 m_2 - (m_1 + m_2) I_{z2}}{S}, \\ g_{23} &= \frac{-(a_2 + c)^2 m_1 m_2 - (m_1 + m_2) (I_{z1} + I_{z2})}{S}, \\ S &= (a_2^2 I_{z1} + c^2 I_{z2}) m_1 m_2 + (m_1 + m_2) I_{z1} I_{z2}. \end{aligned} \quad (18)$$

According to conditions of feedback linearization theory, relative degree of vehicle system should be confirmed as

$$\begin{aligned} L_{g_1} h_1(x) &= \frac{\partial(x_2)}{\partial(\mathbf{X})} \cdot g_1 = 0; \\ L_{g_2} h_2(x) &= \frac{\partial(x_2)}{\partial(\mathbf{X})} \cdot g_2 = g_{22} \neq 0; \end{aligned}$$

$$L_{g_1} h_2(x) = \frac{\partial(x_4 - x_6)}{\partial(\mathbf{X})} \cdot g_1 = 0;$$

$$L_{g_2} h_2(x) = \frac{\partial(x_4 - x_6)}{\partial(\mathbf{X})} \cdot g_2 = 0;$$

$$L_f h_1(x) = \frac{\partial(x_2)}{\partial(\mathbf{X})} \cdot f(\mathbf{X})$$

$$= a_{21} F_{ytf} + a_{22} F_{ytr} + a_{23} F_{ys};$$

$$L_f h_2(x) = \frac{\partial(x_4 - x_6)}{\partial(\mathbf{X})} \cdot f(\mathbf{X}) = x_3 - x_5;$$

$$L_{g_1} L_f h_2(x) = \frac{\partial(x_3 - x_5)}{\partial(\mathbf{X})} \cdot g_1 = \frac{k_w}{I_w} \neq 0;$$

$$L_{g_2} L_f h_2(x) = \frac{\partial(x_3 - x_5)}{\partial(\mathbf{X})} \cdot g_2 = g_{23} \neq 0;$$

$$L_f^2 h_2(x) = \frac{\partial(x_3 - x_5)}{\partial(\mathbf{X})} \cdot f(\mathbf{X})$$

$$= \left(a_{31} + \frac{\varepsilon}{I_w} \right) F_{ytf} + a_{32} F_{ytr} + a_{33} F_{ys}$$

$$+ \frac{C_w}{I_w} x_5 + \frac{k_w}{I_w} x_6 - \frac{k_w}{I_w} \delta_{sw}.$$

(19)

It is concluded that relative degree of vehicle system is 3 less than 6, which satisfies partial feedback linearization.

Coordinate transformation is necessary for the model, as

$$\begin{aligned} \mathbf{Z} = [z_1, z_2, z_3, z_4, z_5, z_6]^T &= \left[x_2, x_4 - x_6, x_3 - x_5, \right. \\ &\quad \left. - \frac{g_{22} x_1}{g_{21}} + x_2, - \frac{g_{23} x_1}{g_{21}} + x_3, x_6 \right], \end{aligned}$$

$$\begin{aligned} \mathbf{X} = [x_1, x_2, x_3, x_4, x_5, x_6]^T &= \left[\frac{g_{21} (z_1 - z_4)}{g_{22}}, z_1, z_5 \right. \\ &\quad \left. + \frac{g_{21} (z_1 - z_4)}{g_{22}}, z_2 + z_6, z_5 - z_3 \right. \\ &\quad \left. + \frac{g_{23} (z_1 - z_4)}{g_{22}}, z_6 \right]^T. \end{aligned} \quad (20)$$

Then, partial feedback linearization control model is as follows:

$$\dot{\xi} = A_c \xi + B_c \gamma(\mathbf{X}) [\mathbf{u} - \alpha(\mathbf{X})],$$

$$\dot{\eta} = q_0(\eta, \xi), \quad (21)$$

$$\mathbf{y} = C_c \xi,$$

where $\xi = [z_1, z_2, z_3]^T$; $\eta = [z_4, z_5, z_6]^T$.

That is,

$$\begin{aligned} \dot{\xi} &= \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} L_f h_1(x) + L_{g_1} h_1(x) u_1 + L_{g_2} h_2(x) u_2 \\ 0 \\ L_f^2 h_2(x) + L_{g_1} L_f h_2(x) u_1 + L_{g_2} L_f h_2(x) u_2 \end{bmatrix}, \\ \dot{\eta} &= \begin{bmatrix} \dot{z}_4 \\ \dot{z}_5 \\ \dot{z}_6 \end{bmatrix} = \begin{bmatrix} \left(a_{21} - \frac{g_{22} a_{11}}{g_{21}}\right) F_{ytf} + \left(a_{22} - \frac{g_{22} a_{12}}{g_{21}}\right) F_{ytr} + \left(a_{23} - \frac{g_{22} a_{13}}{g_{21}}\right) F_{yts} - \frac{g_{22} a_{14} x_2}{g_{21}} \\ \left(a_{31} - \frac{g_{23} a_{11}}{g_{21}}\right) F_{ytf} + \left(a_{32} - \frac{g_{23} a_{12}}{g_{21}}\right) F_{ytr} + \left(a_{33} - \frac{g_{23} a_{13}}{g_{21}}\right) F_{yts} - \frac{g_{23} a_{14} x_2}{g_{21}} \\ x_5 \end{bmatrix}, \quad (22) \\ \mathbf{y} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}. \end{aligned}$$

According to (14) and (22), it is satisfied that $x_2 = x_3 = x_4 = x_5 = x_6 = 0$ if $\xi = [z_1, z_2, z_3]^T \equiv 0$. Thus, the zero dynamic equation $\dot{\eta} = [\dot{z}_4, \dot{z}_5, \dot{z}_6]^T$ is asymptotic stable at the origin. So it is proved that the vehicle feedback linearization model could be effectively controlled by $\xi = [z_1, z_2, z_3]^T$.

Defining control input variable \mathbf{v} , as

$$\begin{aligned} \mathbf{v} &= \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \\ &= \begin{bmatrix} L_f h_1(x) + L_{g_1} h_1(x) u_1 + L_{g_2} h_2(x) u_2 \\ L_f^2 h_2(x) + L_{g_1} L_f h_2(x) u_1 + L_{g_2} L_f h_2(x) u_2 \end{bmatrix}, \quad (23) \end{aligned}$$

then initial system could be transformed as a linear system $\dot{\xi} = A_c \xi + B_c \mathbf{v}$.

4.2. Feedback Linearization Controller Design. A linear 4DOF model of tractor-semitrailer is introduced to obtain a

reference for control output. Just as the controlled model (13), the reference 4DOF vehicle model also considers tractor and semitrailer's lateral motions and yaw motions and steering wheel rotation. The only difference is that it is applied linear tire model, as

$$F_y = k\alpha, \quad (24)$$

where k is tire slip stiffness and α is tire slip angle.

Based on first-order Taylor expansion of magic formula (5), k of each tractor-semitrailer's tire could be calculated by $k_i = B_i C_i D_i$, whose values are shown in Table 2.

The reference 4DOF vehicle model is describes as

$$\dot{\mathbf{X}}_r = A_r \mathbf{X}_r + B_r \delta_{sw}, \quad (25)$$

where the reference state variables $\mathbf{X}_r = [x_{1r}, x_{2r}, x_{3r}, x_{4r}, x_{5r}, x_{6r}]^T = [v_{ytr}, \dot{\phi}_{tr}, \dot{\theta}_r, \theta_r, \dot{\delta}_{fr}, \delta_{fr}]^T$ and the reference outputs are $y_{1r} = x_{1r}$ and $y_{2r} = x_{4r} - x_{6r}$. A_r and B_r are as follows:

$$A_r = \begin{bmatrix} m_t + m_s & -(c + a_s) m_s & a_s m_s & 0 & 0 & 0 \\ cm_t & I_{zt} & 0 & 0 & 0 & 0 \\ a_s m_t & I_{zs} & -I_{zs} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^{-1}$$

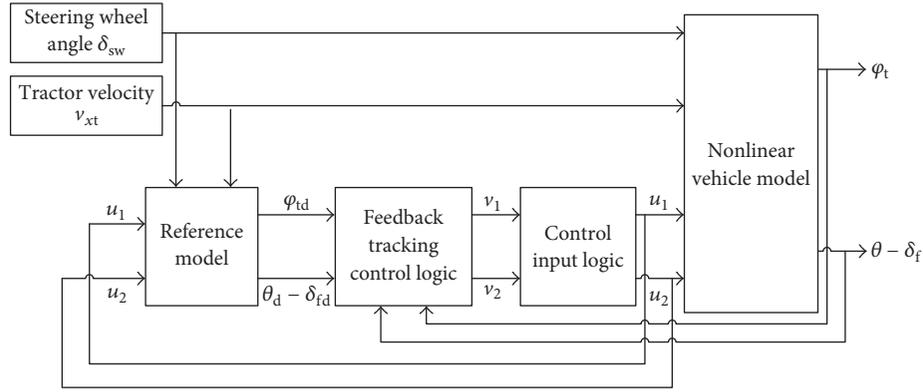


FIGURE 8: Structure of feedback tracking controller.

$$\times \begin{bmatrix} k_{tf} + k_{tr} + k_s & - \left[(m_s + m_t) v_{xt} - \frac{a_t k_{tf} - b_t k_{tr} - (c + a_s + b_s) k_s}{v_{xt}} \right] & \frac{(a_s + b_s) k_s}{v_{xt}} & k_s & 0 & -k_{tf} \\ (a_t + c) k_{tf} + (c - b_t) k_{tr} & -c m_t v_{xt} + \frac{a_t (a_t + c) k_{tf} + b_t (b_t - c) k_{tr}}{v_{xt}} & 0 & 0 & 0 & -(a_t + c) k_{tf} \\ a_s (k_{tf} + k_{tr}) - b_s k_s & -a_s m_t v_{xt} + \frac{a_s (a_t k_{tf} - b_t k_{tr}) + b_s (c + a_s + b_s) k_s}{v_{xt}} & -\frac{b_s (a_s + b_s) k_s}{v_{xt}} & -b_s k_s & 0 & -a_s k_{tf} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -\frac{\varepsilon k_{tf}}{v_{xt}} & -\frac{a_t \varepsilon k_{tf}}{v_{xt}} & 0 & 0 & -C_w & \varepsilon k_{tf} - K_w \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$B_r = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{K_w}{i} & 0 \end{bmatrix}.$$

(26)

Figure 8 shows a designed feedback tracking controller. The control input variable \mathbf{v} is defined as

$$\begin{aligned}
 v_1 &= -\lambda_1 e_1 + \dot{y}_{1r}, \\
 v_2 &= -\lambda_2 e_2 - \lambda_3 \dot{e}_2 + \ddot{y}_{2r},
 \end{aligned} \quad (27)$$

where e_1 and e_2 are tracking errors, $e_1 = y_1 - y_{1r}$ and $e_2 = y_2 - y_{2r}$.

Equation (27) is satisfied as

$$\begin{aligned}
 \dot{e}_1 + \lambda_1 e_1 &= 0, \\
 \ddot{e}_2 + \lambda_3 \dot{e}_2 + \lambda_2 e_2 &= 0.
 \end{aligned} \quad (28)$$

Therefore, the closed-loop system errors e_1 and e_2 are exponentially stable by reasonable choice of λ_1 , λ_2 , and λ_3 .

4.3. Control Performance Analysis. The specific control effects are mainly considered by the control stability, rapidity, and accuracy. Typical effect parameters contain peak time t_p , overshoot $\sigma\%$, and regulation time t_s .

Applying the proposed vehicle dynamic model, driving circles of step steering and single-lane changing with 20 m/s

are conducted to simulate tractor-semitrailer lane changing on highway under rainy weather.

Step steering is used to select control parameters λ_1 , λ_2 , and λ_3 by contrasting control effects. Choosing 4 sets of data of λ_1 , λ_2 , and λ_3 control effects of each set and are shown in Figure 9.

It can be seen from Figure 9 that there are slight difference among 4 sets of data. Then, maximum values of e_1 , e_2 , typical effect parameters t_p , $\sigma\%$, and t_s and DYC torque input are, respectively, calculated, as listed in Table 4.

Table 4 indicates the influence of control parameters for vehicle system control response and accuracy. It clearly showed some diametrically opposed change trend of control effects, such as e_1 and e_2 and $\sigma\%$ and t_s . In consideration of response time, control input size, and system stability, when $\lambda_1 = 300$, $\lambda_2 = 5000$, and $\lambda_3 = 6$, values of the peak time $t_p = 1.08$ s and regulation time $t_s = 2.08$ s are all smaller, and the DYC torque input $T_{\max} = 779.4590$ N is also lower correspondingly. Because of the balance of control effects, Group 2 is determined to be as final control parameters.

Based on the selected control parameters, critical scenarios of steady steering ($\delta_f = 0.05$) and single-lane changing

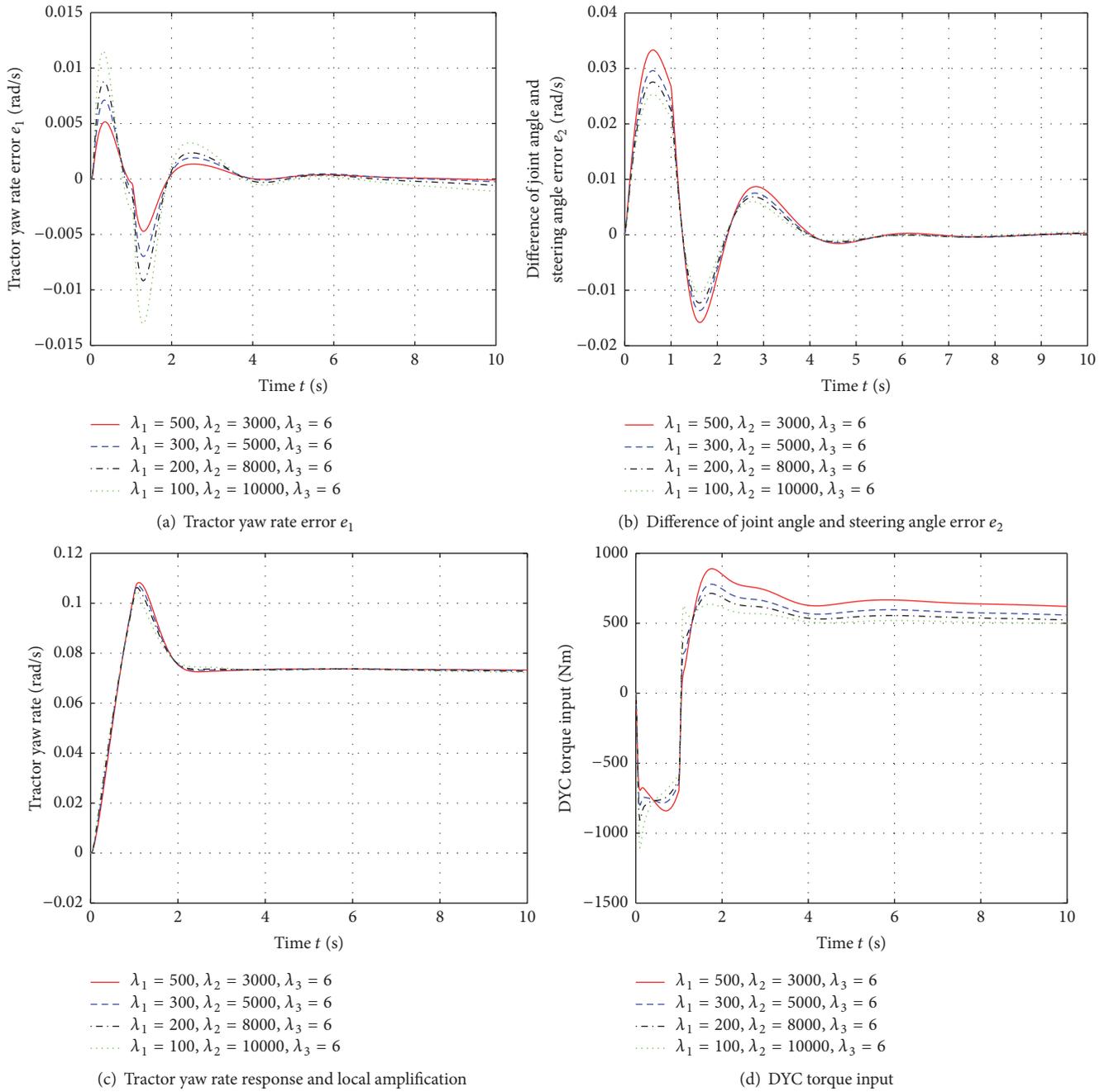


FIGURE 9: Control effects with 4 sets of data.

TABLE 4: Calculated controlled effect parameters and DYC input torques with 4 sets of data.

No.	λ_1	λ_2	λ_3	$e_{1\max}/\text{rad/s}$	$e_{2\max}/\text{rad}$	Tractor yaw rate response			DYC T_{\max}/N
						t_p/s	$\sigma\%$	t_s/s	
(1)	500	3000	6	0.0052	0.0303	1.11	47.77	2.04	809.0810
(2)	300	5000	6	0.0071	0.0296	1.08	46.59	2.08	779.4590
(3)	200	8000	6	0.0088	0.0290	1.07	46.02	2.16	751.6507
(4)	100	10000	6	0.0115	0.0281	1.06	45.26	3.34	1.2294e + 003

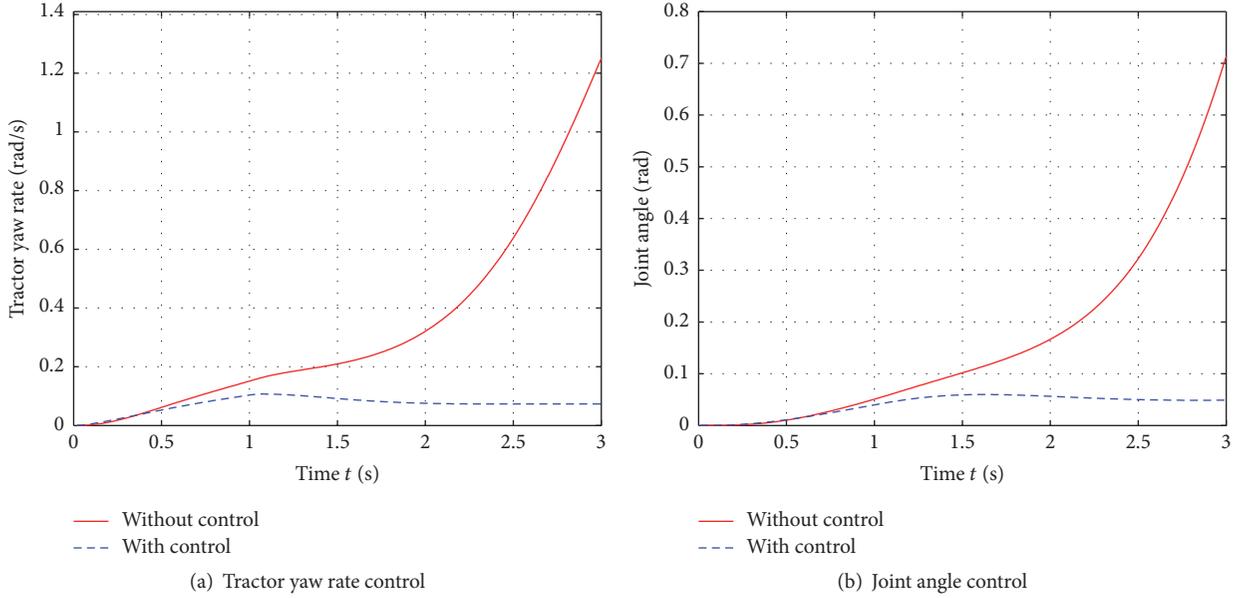


FIGURE 10: Typical contract of vehicle dynamic variables in critical steady steering.

($A_s = 0.07$ rad, $f = 0.2$ Hz), just as introduced in Section 3, are simulated with the established feedback linearization controller. Typical contract of vehicle dynamic variables is shown in Figures 10 and 11.

Figures 10 and 11 clearly show the control effects in critical steady steering and single-lane changing conditions. It can be seen from Figure 10 that if it is without control, tractor and semitrailer yaw motions increase sharply and joint angle will reach to 0.7 rad (about 40.1 deg) in 3 seconds, which indicates that there is jackknife instability in critical steady steering. But vehicle will reach to an ideal steady state value with steady steering if applying the designed controller. Similarly, in Figure 11, vehicle dynamics is stable and the yaw motions change regularly with control in critical single-lane changing, but if it is without control, it is also obvious that the vehicles yaw sharply as yaw rate and joint angle increase rapidly (tractor yaw rate reaches about 2 rad/s in 5 seconds and joint angle reaches about 150 deg), which indicates serious sideslip and jackknife instability.

Looking back at Hopf bifurcation phenomenon and instability in critical steady steering or single-lane changing, it can be seen from Figures 10 and 11 that, by feedback linearization controller, combined with AFS and DYC strategy, vehicle's dynamic characteristics could be effectively controlled and ensure stability. In Figure 12, taking the critical single-lane changing as an example, real parts of calculated eigenvalues with control are all lying in the negative half plane with $A_s = 0.07$ rad, $f = 0.2$ Hz, which indicates the control effects in eliminating bifurcation and ensure stability.

5. Discussion and Conclusion

In this paper, considering the high degree of danger and complex maneuvers for tractor-semitrailer on low adhesion coefficient road, the nonlinear lane changing behaviors on

highways under rainy weather have been studied with a verified 5DOF mechanical model based on a test experiment, and a feedback linearization controller by DYC and AFS strategy is designed to ensure stability in critical scenarios. Points of discussion and conclusions can be summarized as follows:

- (1) Nonlinear theory is used to analyze the relationship between the eigenvalues of the Jacobian matrix and vehicle stability. Considering the transient response, the eigenvalues of Jacobian matrix may be positive or have positive real parts as steering wheel angle approaches peak, but the vehicle system is still stable with the angle returns. Therefore, unlike steady steering, transient stability is satisfied if there are transient positive real parts eigenvalues, so it is definitely improved that eigenvalues lying in the negative half plane is sufficient but not necessary for judging stability. It is also clearly concluded that Hopf bifurcation occurs with critical steady and sinusoidal steering, which translates into an oscillatory behavior leading to instability.
- (2) It is realized for application feedback linearization theory in tractor-semitrailer stability control. Design process of an integrated controller is detailed presented with vehicle mechanical model, which contains model building, feedback linearization, and control parameters selection. The consequences of performance analysis introduce new insight into the vehicle nonlinear stability control. The eigenvalues analysis results also provide a substantive understanding on control principle.
- (3) Through analyzing the vehicle nonlinear stability in special driving conditions, the dynamic rule and stability mechanism for tractor-semitrailers changing

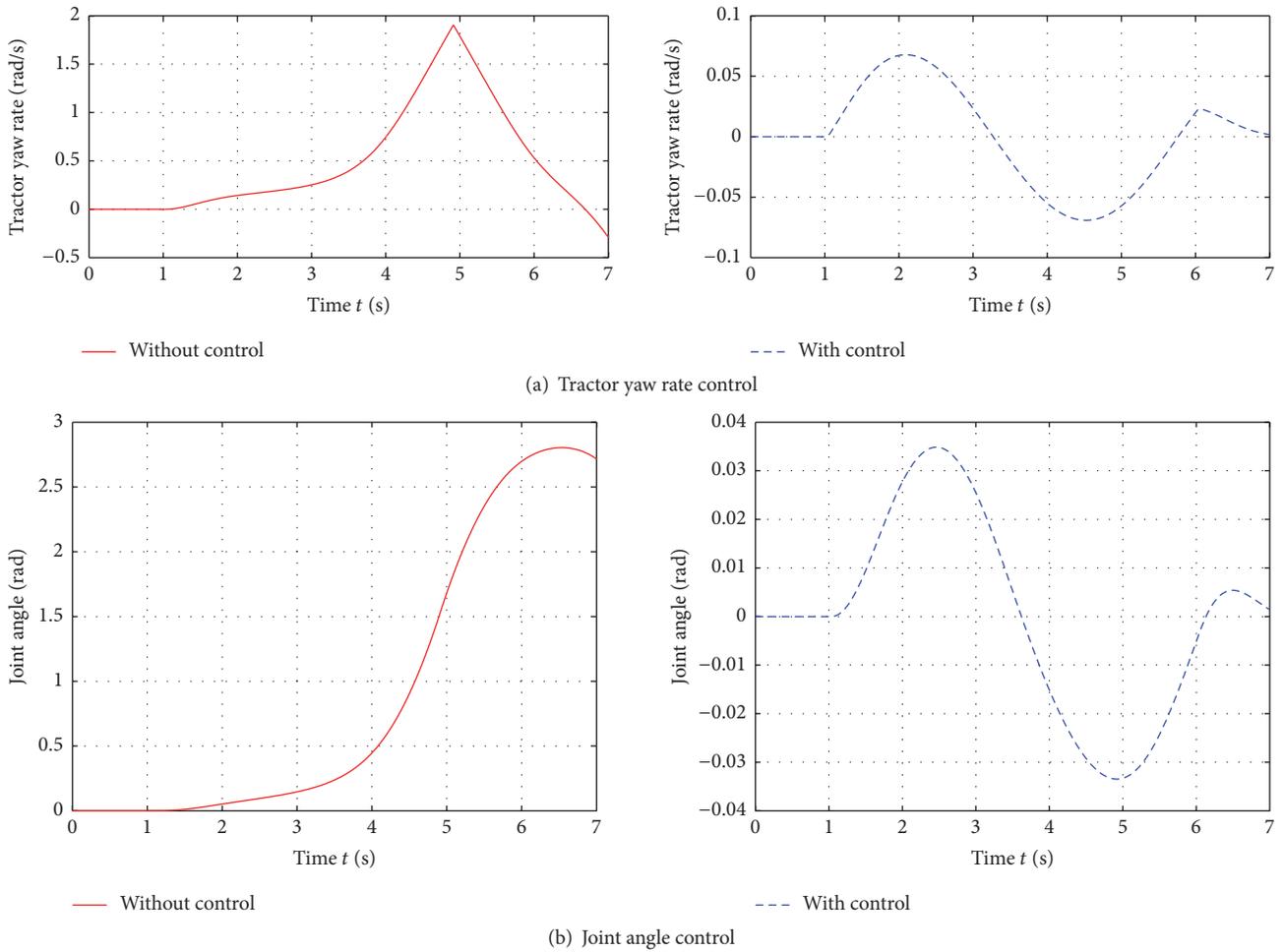


FIGURE 11: Typical contract of vehicle dynamic variables in critical single-lane changing.

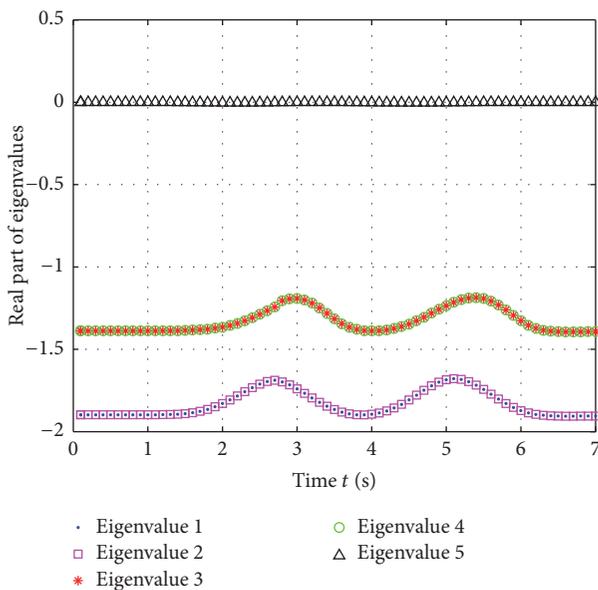


FIGURE 12: Changes in real parts of calculated eigenvalues with control in critical single-lane changing.

lanes at high speeds under rainy weather are well mastered. The bifurcation analysis and nonlinear control provided could be informative for research into vehicle dynamics and active safety control design.

Furthermore, tractor-semitrailers have complex nonlinear behaviors, especially considering driver and environment characteristics. The following could be considered in the future:

- (1) The driver is always an early decision-maker for vehicle movement; it is essential to study the lane-changing nonlinear behaviors of tractor-semitrailers associated with driver characteristics.
- (2) As the study on adhesion coefficient caused by rainy weather, the influence of the other meteorological effects, such as rain for driver sight distance and high winds for vehicle dynamic, should be taken into account to comprehensively analyze the driving stability of tractor-semitrailers.
- (3) Synergetic thought should be introduced creatively to tractor-semitrailers, not only for single tractor and

semitrailer internal relationship, but also for cooperative operation management of a dynamic motorcade, under a driver/vehicle/road integrated system.

As we know, traffic safety is the interaction results of human, vehicle, and environment. First, based on the essence of stability mechanism and nonlinear dynamic characteristics of tractor-semitrailer, advanced active control devices similar to ABS and ESC, should be developed and equipped, especially on slippery pavement. Second, large vehicle drivers' physiological and psychological characteristics should be paid more attention and it is necessary to research and apply ADAS (Advanced Driver Assistance Systems), such as fatigue warning, auxiliary brake, and Lane keeping assist system. Last but not least, with the development of Internet technology, ITS (Intelligent Traffic System) should be widely applied to build a harmonious running state.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Grants nos. 51775565 and 51208500), the Tianjin Science and Technology Innovation Platform Plan (Grant no. 16PTGCCX00150), the National Natural Science Foundation of Tianjin (Key Program, Grant no. 16JCZDJC38200), the University Foundation of Tianjin University of Technology and Education (Grant no. KYQD 1710), the Opening Fund of Guangdong Key Laboratory of Intelligent Transportation System (Grant no. 201701002), and the project named as China New Energy Automobile Products Testing Conditions Research and Development with Guangzhou traffic condition data collection. The first author would like to thank Dr. Z. W. Guan and Dr. R. H. Zhang for valuable discussions to improve the quality and presentation of the paper.

References

- [1] H. Troger and K. Zeman, "Application of bifurcation theory to tractor-semitrailer dynamics," *Vehicle System Dynamics*, vol. 10, no. 2-3, pp. 156-161, 1981.
- [2] H. Troger and K. Zeman, "A nonlinear analysis of the generic types of loss of stability of the steady state motion of a tractor-semitrailer," *Vehicle System Dynamics*, vol. 13, no. 4, pp. 161-172, 1984.
- [3] A. Dunn, *Jackknife stability of articulated tractor semitrailer vehicles with high-output brakes and jackknife detection on low coefficient surfaces [MSc. thesis]*, Ohio State University, Columbus, Ohio, USA, 2003.
- [4] M.-W. Suh, Y.-K. Park, S.-J. Kwon, S.-H. Yang, and B.-C. Park, "A simulation program for the braking characteristics of tractor-semitrailer vehicle," *SAE Technical Papers*, vol. 9, no. 2, pp. 152-167, 2000.
- [5] A. L. Dunn, C. B. Tanner, D. R. Morr et al., "The influence of disablement of various brakes on the dry stopping performance and stability of a tractor-semitrailer," *SAE International Journal of Commercial Vehicles*, vol. 2, no. 1, pp. 1-16, 2009.
- [6] M. F. J. van de Molengraft-Luijten, I. J. M. Besselink, R. M. A. F. Verschuren, and H. Nijmeijer, "Analysis of the lateral dynamic behaviour of articulated commercial vehicles," *Vehicle System Dynamics*, vol. 50, supplement 1, pp. 169-189, 2012.
- [7] X. Yang and J. Xiong, "Nonlinear yaw dynamics analysis and control for the tractor-semitrailer vehicle," *International Journal of Heavy Vehicle Systems*, vol. 20, no. 3, pp. 253-288, 2013.
- [8] N. Ding, X. Shi, Y. Zhang, and W. Chen, "Analysis of bifurcation and stability for a tractor semi-trailer in planar motion," *Vehicle System Dynamics*, vol. 52, no. 12, pp. 1729-1751, 2014.
- [9] S. Sadri and C. Wu, "Stability analysis of a nonlinear vehicle model in plane motion using the concept of Lyapunov exponents," *Vehicle System Dynamics*, vol. 51, no. 6, pp. 906-924, 2013.
- [10] O. Polach and I. Kaiser, "Comparison of methods analyzing bifurcation and hunting of complex rail vehicle models," *Journal of Computational and Nonlinear Dynamics*, vol. 7, no. 4, Article ID 041005, 2012.
- [11] X. W. Wu and M. R. Chi, "Parameters study of hopf bifurcation in railway vehicle system," *Journal of Computational Nonlinear Dynamics*, vol. 10, no. 3, pp. 43-53, 2014.
- [12] S. Tousi, A. K. Bajaj, and W. Soedel, "Closed-loop directional stability of car-trailer combinations in straight-line motion," *Vehicle System Dynamics*, vol. 51, no. 6, pp. 906-924, 2013.
- [13] Z. Liu, "Characterisation of optimal human driver model and stability of a tractor-semitrailer vehicle system with time delay," *Mechanical Systems and Signal Processing*, vol. 21, no. 5, pp. 2080-2098, 2007.
- [14] M. Sarvi, "Heavy commercial vehicles-following behavior and interactions with different vehicle classes," *Journal of Advanced Transportation*, vol. 47, no. 6, pp. 572-580, 2013.
- [15] G. Yu, P. Wang, X. Wu, and Y. Wang, "Linear and nonlinear stability analysis of a car-following model considering velocity difference of two adjacent lanes," *Nonlinear Dynamics*, vol. 84, no. 1, pp. 387-397, 2016.
- [16] F. D. Rossa, M. Gobbi, G. Mastinu, C. Piccardi, and G. Previati, "Bifurcation analysis of a car and driver model," *Vehicle System Dynamics*, vol. 52, supplement 1, pp. 142-156, 2014.
- [17] Z. Liu and G. Payre, "Global bifurcation analysis of a nonlinear road vehicle system," *Journal of Computational and Nonlinear Dynamics*, vol. 2, no. 4, pp. 308-315, 2007.
- [18] S. Li, S. Yang, and L. Chen, "A nonlinear vehicle-road coupled model for dynamics research," *Journal of Computational and Nonlinear Dynamics*, vol. 8, no. 2, Article ID 21001, pp. 1-15, 2013.
- [19] Z. Liu, K. Hu, and K.-W. Chung, "Nonlinear analysis of a closed-loop tractor-semitrailer vehicle system with time delay," *Mechanical Systems and Signal Processing*, vol. 76-77, pp. 696-711, 2016.
- [20] I. Koglbauer, J. Holzinger, A. Eichberger, and C. Lex, "Drivers' interaction with adaptive cruise control on dry and snowy roads with various tire-road grip potentials," *Journal of Advanced Transportation*, vol. 2017, Article ID 5496837, 10 pages, 2017.
- [21] Y. Bie, T. Z. Qiu, C. Zhang, and C. Zhang, "Introducing weather factor modelling into macro traffic state prediction," *Journal of Advanced Transportation*, vol. 2017, Article ID 4879170, 15 pages, 2017.
- [22] T. Tang, C. Li, H. Huang, and H. Shang, "A new fundamental diagram theory with the individual difference of the driver's

- perception ability," *Nonlinear Dynamics*, vol. 67, no. 3, pp. 2255–2265, 2012.
- [23] T. Tang, Y. Wang, X. Yang, and Y. Wu, "A new car-following model accounting for varying road condition," *Nonlinear Dynamics*, vol. 70, no. 2, pp. 1397–1405, 2012.
- [24] T.-Q. Tang, L. Caccetta, Y.-H. Wu, H.-J. Huang, and X.-B. Yang, "A macro model for traffic flow on road networks with varying road conditions," *Journal of Advanced Transportation*, vol. 48, no. 4, pp. 304–317, 2014.
- [25] D. Ma, X. Luo, W. Li, S. Jin, W. Guo, and D. Wang, "Traffic demand estimation for lane groups at signal-controlled intersections using travel times from video-imaging detectors," *IET Intelligent Transport Systems*, vol. 11, no. 4, pp. 222–229, 2017.
- [26] D. Ma, F. Fu, S. Jin et al., "Gating control for a single bottleneck link based on traffic load equilibrium," *International Journal of Civil Engineering*, vol. 14, no. 5, pp. 281–293, 2016.
- [27] D. Ma, X. Luo, S. Jin, D. Wang, W. Guo, and F. Wang, "Lane-based saturation degree estimation for signalized intersections using travel time data," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 136–148, 2017.
- [28] X. Lin, N. Ding, G. Xu, and F. Gao, "High speed optimal yaw stability of tractor-semitrailers with active trailer steering," *SAE Technical Papers*, vol. 1, 2014.
- [29] L. Palkovics and M. El-Gindy, "Examination of different control strategies of heavy-vehicle performance," *Journal of Dynamic Systems, Measurement, and Control*, vol. 118, no. 3, pp. 489–497, 1996.
- [30] S. Chang, H. G. Xu, and H. F. Liu, " H_{∞} loop shaping robust controller design for tractor-semitrailer," *Journal of Jilin University*, vol. 41, no. 6, pp. 1571–1576, 2011.
- [31] J. Liang, X. Yuan, Y. Yuan, Z. Chen, and Y. Li, "Nonlinear dynamic analysis and robust controller design for Francis hydraulic turbine regulating system with a straight-tube surge tank," *Mechanical Systems and Signal Processing*, vol. 85, pp. 927–946, 2017.
- [32] H. Takenaga, M. Konishi, and J. Imai, "Route tracking control of tractor-trailer vehicles based on fuzzy controller," in *Proceedings of the 5th International Workshop on Computational Intelligence and Applications*, pp. 105–110, Higashihiroshima, Japan, November 2009.
- [33] C. S. Chin, X. Ji, W. L. Woo, T. J. Kwee, and W. Yang, "Modified multiple generalized regression neural network models using fuzzy C-means with principal component analysis for noise prediction of offshore platform," *Neural Computing & Applications*, no. 6, pp. 1–16, 2017.
- [34] X. Yang, J. Song, and J. Gao, "Fuzzy logic based control of the lateral stability of tractor semitrailer vehicle," *Mathematical Problems in Engineering*, vol. 2015, Article ID 692912, 16 pages, 2015.
- [35] C. Zong, T. Zhu, C. Wang, and H. Liu, "Multi-objective stability control algorithm of heavy tractor semi-trailer based on differential braking," *Chinese Journal of Mechanical Engineering*, vol. 25, no. 1, pp. 88–97, 2012.
- [36] Z.-Z. He, H. Qi, M.-J. He, and L.-M. Ruan, "Experimental research on the photobiological hydrogen production kinetics of *Chlamydomonas reinhardtii* GY-D55," *International Journal of Hydrogen Energy*, vol. 41, no. 35, pp. 15651–15660, 2016.
- [37] R. Zhang, K. Li, Z. He, H. Wang, and F. You, "Advanced emergency braking control based on a nonlinear model predictive algorithm for intelligent vehicles," *Applied Sciences*, vol. 7, no. 5, pp. 504–522, 2017.
- [38] H. Xiao, R. Cui, and D. Xu, "A sampling-based bayesian approach for cooperative multiagent online search with resource constraints," *IEEE Transactions on Cybernetics*, 2017.
- [39] F. You, R. Zhang, G. Lie, H. Wang, H. Wen, and J. Xu, "Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system," *Expert Systems with Applications*, vol. 42, no. 14, pp. 5932–5946, 2015.
- [40] Z. Liu, Y. Zhang, S. Wang, and Z. Li, "A trial-and-error method with autonomous vehicle-to-infrastructure traffic counts for cordon-based congestion pricing," *Journal of Advanced Transportation*, vol. 2017, Article ID 9243039, 8 pages, 2017.
- [41] J. Wang, C. Wang, J. Lv, Z. Zhang, and C. Li, "Modeling travel time reliability of road network considering connected vehicle guidance characteristics indexes," *Journal of Advanced Transportation*, vol. 2017, Article ID 2415312, 9 pages, 2017.
- [42] C. S. Chin, W. Atmodihardjo, L. W. Woo, and E. Mesbahi, "Remote temperature monitoring device using a multiple patients-coordinator set design approach," *ROBOMECH Journal*, vol. 2, no. 1, pp. 1–9, 2015.
- [43] R. Zhang, J. Wu, L. Huang, and F. You, "Study of bicycle movements in conflicts at mixed traffic unsignalized intersections," *IEEE Access*, vol. 5, pp. 10108–10117, 2017.
- [44] P. Chen, W. Zeng, G. Yu, and Y. Wang, "Surrogate safety analysis of pedestrian-vehicle conflict at intersections using unmanned aerial vehicle videos," *Journal of Advanced Transportation*, vol. 2017, Article ID 5202150, 12 pages, 2017.
- [45] C. S. Chin, W. P. Lin, and J. Y. Lin, "Experimental validation of open-frame ROV model for virtual reality simulation and control," *Journal of Marine Science & Technology*, no. 2, pp. 1–21, 2017.
- [46] F.-C. Wang and W.-H. Fang, "The development of a PEMFC hybrid power electric vehicle with automatic sodium borohydride hydrogen generation," *International Journal of Hydrogen Energy*, vol. 42, no. 15, pp. 10376–10389, 2017.
- [47] W. Lin and C. S. Chin, "Block diagonal dominant remotely operated vehicle model simulation using decentralized model predictive control," *Advances in Mechanical Engineering*, vol. 9, no. 4, pp. 1–24, 2017.
- [48] D. K. Kim, H. E. Min, I. M. Kong et al., "Parametric study on interaction of blower and back pressure control valve for a 80-kW class PEM fuel cell vehicle," *International Journal of Hydrogen Energy*, vol. 41, no. 39, pp. 17595–17615, 2016.
- [49] F. Tazerart, Z. Mokrani, D. Rekioua, and T. Rekioua, "Direct torque control implementation with losses minimization of induction motor for electric vehicle applications with high operating life of the battery," *International Journal of Hydrogen Energy*, vol. 40, no. 39, pp. 13827–13838, 2015.
- [50] R. Zhang, Z. He, H. Wang, F. You, and K. Li, "Study on self-tuning tyre friction control for developing main-servo loop integrated chassis control system," *IEEE Access*, vol. 5, pp. 6649–6660, 2017.
- [51] P. Huynh, T.-H. Do, and M. Yoo, "A probability-based algorithm using image sensors to track the LED in a vehicle visible light communication system," *Sensors*, vol. 17, no. 2, pp. 347–357, 2017.
- [52] R. Huang, H. Liang, P. Zhao, B. Yu, and X. Geng, "Intent-estimation- and motion-model-based collision avoidance method for autonomous vehicles in urban environments," *Applied Sciences*, vol. 7, no. 5, p. 457, 2017.
- [53] B. Gao, R. Zhang, and X. Lou, "Modeling day-to-day flow dynamics on degradable transport network," *PLoS ONE*, vol. 11, no. 12, Article ID e0168241, 2016.

- [54] V. V. Dixit, S. Chand, and D. J. Nair, "Autonomous vehicles: disengagements, accidents and reaction times," *PLoS ONE*, vol. 11, no. 12, Article ID 0168054, 2016.
- [55] D. Ahn, H. Park, S. Hwang, and T. Park, "Reliable identification of vehicle-boarding actions based on fuzzy inference system," *Sensors*, vol. 17, no. 2, pp. 333–342, 2017.
- [56] R. Dai, Y. Lu, C. Ding, and G. Lu, "The effect of connected vehicle environment on global travel efficiency and its optimal penetration rate," *Journal of Advanced Transportation*, vol. 2017, Article ID 2697678, 10 pages, 2017.
- [57] W. Sun, X. Wu, Y. Wang, and G. Yu, "A continuous-flow-intersection-lite design and traffic control for oversaturated bottleneck intersections," *Transportation Research Part C: Emerging Technologies*, vol. 56, pp. 18–33, 2015.
- [58] X. Wu, X. He, G. Yu, A. Harmandayan, and Y. Wang, "Energy-optimal speed control for electric vehicles on signalized arterials," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2786–2796, 2015.
- [59] H. Li, C. Li, H. Li, Y. Li, and Z. Xing, "An integrated altitude control design for a tail-sitter UAV equipped with turbine engines," *IEEE Access*, vol. 5, pp. 10941–10952, 2017.
- [60] X. Bao, H. Li, D. Xu, L. Jia, B. Ran, and J. Rong, "Traffic vehicle counting in jam flow conditions using low-cost and energy-efficient wireless magnetic sensors," *Sensors*, vol. 16, no. 11, pp. 1868–1883, 2016.
- [61] M. Ibarra-Arenado, T. Tjahjadi, J. Pérez-Oria, S. Robla-Gómez, and A. Jiménez-Avello, "Shadow-based vehicle detection in Urban traffic," *Sensors*, vol. 17, no. 5, pp. 975–993, 2017.
- [62] M. Da Lio, F. Biral, E. Bertolazzi et al., "Artificial co-drivers as a universal enabling technology for future intelligent vehicles and transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 244–263, 2015.
- [63] T. Yin, D. Huang, and S. Fu, "Adaptive decision and robust control under parallel precedence constraints in human strategy modeling," *IEEE Access*, vol. 5, pp. 9944–9956, 2017.
- [64] Z. S. Yu, Z. F. Wang, and W. Y. Wang, *Handbook of Mechanical Engineering*, China Communications Press, Beijing, China, 2003.
- [65] H. B. Pacejka, *Tire and Vehicle Dynamics*, Butterworth-Heinemann, London, UK, 2012.
- [66] E. Bakker, L. Nyborg, and H. B. Pacejka, "Tyre modelling for use in vehicle dynamics studies," *SAE Technical Papers*, Article ID 870421, pp. 1–15, 1987.
- [67] H. Tögl and A. Steindl, *Nonlinear Dynamic Introduction*, Springer-Verlag Press, Berlin, Germany, 2012.
- [68] Q. C. Zhang, H. L. Wang, and Z. W. Zhu, *Bifurcation and Chaos Theory and Application*, Tianjin University Press, Tianjin, China, 2005.
- [69] D. P. Li, *Nonlinear Control System*, Northwestern Polytechnical University Press, Xi'an, China, 2009.
- [70] S. Tippaya, S. Sitjongsataporn, T. Tan, M. M. Khan, and K. Chamnongthai, "Multi-modal visual features-based video shot boundary detection," *IEEE Access*, vol. 5, pp. 12563–12575, 2017.
- [71] O. Ur-Rehman and N. Zivic, "Two-phase method for image authentication and enhanced decoding," *IEEE Access*, vol. 5, pp. 12158–12167, 2017.
- [72] R. Zhang, Y. Ma, F. You, T. Peng, Z. He, and K. Li, "Exploring to direct the reaction pathway for hydrogenation of levulinic acid into γ -valerolactone for future Clean-Energy Vehicles over a magnetic Cu-Ni catalyst," *International Journal of Hydrogen Energy*, vol. 42, no. 40, pp. 25185–25194, 2017.

Research Article

Liveness-Based RRT Algorithm for Autonomous Underwater Vehicles Motion Planning

Yang Li,¹ Fubin Zhang,¹ Demin Xu,¹ and Jiguo Dai²

¹The School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

²Department of Mechanical Engineering, Texas Tech University, Lubbock, TX 79409, USA

Correspondence should be addressed to Fubin Zhang; zhangfb@nwpu.edu.cn

Received 26 June 2017; Accepted 13 September 2017; Published 19 October 2017

Academic Editor: Cheng S. Chin

Copyright © 2017 Yang Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Motion planning is a crucial, basic issue in robotics, which aims at driving vehicles or robots towards to a given destination with various constraints, such as obstacles and limited resource. This paper presents a new version of rapidly exploring random trees (RRT), that is, liveness-based RRT (Li-RRT), to address autonomous underwater vehicles (AUVs) motion problem. Different from typical RRT, we define an index of each node in the random searching tree, called “liveness” in this paper, to describe the potential effectiveness during the expanding process. We show that Li-RRT is provably probabilistic completeness as original RRT. In addition, the expected time of returning a valid path with Li-RRT is obviously reduced. To verify the efficiency of our algorithm, numerical experiments are carried out in this paper.

1. Introduction

In the last decade, motion planning for single robot or multiple robot system has gained a lot of research interest as one of the primary problems in robot field. In most application cases, it mainly involves the structured mobility to drive the robots to the final destination given any initial states [1]. In general, the motion planning algorithms can be generally categorized into two types [2]. The first ones return the positions along the path from start to the goal, which are called path planning algorithms. In addition, the second ones, that is, trajectory planning algorithms (also called motion planning algorithms), return a sequence set of input space that drives the robot towards the destination.

In recent years, various algorithms have been proposed for both path and trajectory planning problems. Artificial potential fields (APF) can be viewed as a powerful method for path planning, which was firstly proposed in [3]. Although APF suffers from the presence of local minimums, it is adaptable in such cases with online intractable changes of environment. Meantime, various of methods are proposed to avoid sticking into local minimums, such as potential functions without local minimums [4] and random search methods [5–7].

Another remarkable probabilistic method was proposed as Probabilistic Roadmap (PRM) [8–11]. The key idea of PRM is to generate a connective random graph that comprises valid nodes selected randomly in the free configuration space (C-space). Meanwhile, RRT was presented as a single query incremental algorithm [12]. PRM and RRT are shown with more effectiveness, especially facing high-dimensional problems. Besides, they are insensitive to the complexity of the problems. RRT has provably probabilistic completeness, which means that the probability that RRT cannot find a valid path from the initial to the goal position converges to zero, as the number of samples tends to infinity. Also, the concept of probabilistic completeness is officially defined, which is a crucial property, although it is weaker than “completeness” in the field of sample based algorithms. Consequently, RRT has been applied in many applications ranging from industrial production to military battle systems.

Several RRT based algorithms are designed in order to deal with different constraints or to provide solutions more effectively. For example, in [13–16], additional dynamic differential constraints are considered when extending the random tree. A kind of new variation space, called trajectory parameter-space (TP-space), was presented in [17].

Particularly each edge is kinematically feasible once it is generated by two points in TP-space. In [18], the expanding process utilizes implicit flood-fill-like mechanism to avoid the local minimum. Thus, the expected time to find the solution is reduced. Because of the insensitivity to the dimension of problems, it is able to be used to search time invariant parameters [19]. It is a fact that AUVs have been widely applied in the marine resources exploration and utilization [20–23]. In order to map a scalar field, mutual information based multidimensional RRT algorithm was presented for multiple AUVs in [15].

However, RRT is proven to converge almost to a nonoptimal path [24], in which RRT* was detailed to be asymptotically optimal. That means RRT* will “almost surely” find the optimal path as the number of samples tends to infinity. RRT* applies the reconstruction of the searching tree, which always contains the optimal subtree at each iteration. Since then several RRT* based optimal planning algorithms were proposed [25–28] and applied in many fields [29, 30]. LBT-RRT was proposed in [25], which is a combination algorithm with RRT and RRT*. It defines a variable ϵ to describe the “distance” away from the optimal path. In other words, ϵ can be viewed as the comparison between RRT and RRT*. If ϵ equals zero, it becomes RRT* essentially. In [26], RRT* is refined to be capable of replanning with dynamic environments. In [27], the limitation, that is, requirements for an asymptotical optimal steer function during the tree’s expansion, is provably addressed based on sparse data structure.

From [31], we know that the sampling strategies have much effect on the efficiency of sampling-based algorithms. In early researches, typical choice is sampling uniformly in C-space [8, 12, 24]. Every node in C-space is sampled with exactly the same probability, by which the prior information actually is not exploited. For instance, a new node intuitively prefers to be sampled away from the region where a large number of samples already exist. And it certainly reduces the probability that samples are generated in the case of narrow passages [31]. For this reason, various heuristic methodologies are designed. Goal bias is generally utilized to enhance extending towards the goal due to its effectiveness. Medial axis is combined with path planning algorithms to search the C-space with much more efficiency [8]. On the contrary, Gaussian sampling strategy prefers the region nearby obstacles [10, 32]. In this view, geometric shapes of obstacles are able to be quickly constructed. Some other hybrid sampling strategies are proposed to combine two or more methods. These methods uniformly consider the information of the environment to refine the sampling probability; however, the information of nodes is not totally utilized. Specifically, regions where a number of nodes are already sampled can be mostly considered with less worth. In other words, it is preferred to sample in “new” regions.

In this paper, we demonstrate the expanding mechanism of the random tree and make an estimation of the expanding ability of each node. We present several new indicators in the expanding procedure, which are utilized in our new algorithm. Also, we detail the indicators quantitative descriptions of the nodes. Meanwhile, we provide theoretical analysis and property of our algorithm with mathematical proof (i.e.,

probabilistic completeness and outperforming compared to RRT). The proof is based on a technical notation attraction sequence and probability theory.

The reminder of the paper is organized as follows. Section 2 introduces the definition and notations in this paper. The indicators which reflect the growing of the tree and expanding ability are analyzed in Section 3. Then our new algorithm, that is, Li-RRT, is described in detail. In Section 4, the analysis of Li-RRT is provided. Section 5 presents an application example for a planar mobile robot. In final, concluding remarks are drawn in Section 6.

2. Preliminaries and Problem Formation

2.1. Problem Formulation. Path planning problem can be generally viewed as a search problem in the C-space $C \subseteq \mathbb{R}^n$. For instance, the C-space of a planar two-wheel mobile robot can be defined as $C \subseteq \mathbb{R}^3$, that is, position (x, y) , heading angle θ . In this way, any $x \in C \subseteq \mathbb{R}^3$ can describe the state of the mobile robot. Additionally, we define $C_{\text{obs}} \subset C$ and $C_{\text{free}} = C \setminus C_{\text{obs}}$ as the obstacle C-space and free C-space separately. The path planning problems require a finite sequence of states in C_{free} from an initial state to a given goal state. Here we present the basic path planning problem as defined in [24].

Problem 1 (path planning problem [24]). The bounded C-space C , obstacle space C_{obs} , initial state $x_{\text{start}} \in C_{\text{free}}$, goal region $X_{\text{goal}} \in C_{\text{free}}$, denoted as tuple $\{x_{\text{start}}, C_{\text{free}}, X_{\text{goal}}\}$, return a path $\xi : [0, 1] \rightarrow C_{\text{free}}$, such that $\xi(0) = x_{\text{start}}$, $\xi(1) \in X_{\text{goal}}$. If there exist no feasible paths, return failure.

Facing path planning problems, each element of dimension is considered independently with no other constraints. However, it is unrealistic in most applications. Specifically, the kinodynamic of AUVs is restricted in case of uncontrollability. It means that there exists corresponding available input sequence $u(s) : [0, 1] \rightarrow U$ that drives the AUV from x_{start} to X_{goal} . Subsequently the path can also be viewed as a function of input sequence $\xi(u(\cdot)) : u(\cdot) \rightarrow C_{\text{free}}$. Formally, we give the definition of trajectory planning problem for AUVs.

Problem 2 (trajectory planning problem for AUVs). Given $\{x_{\text{start}}, C_{\text{free}}, X_{\text{goal}}\}$ and kinodynamic constraint of AUVs $\dot{x} = f(x, u)$, return an available control sequence $u : [0, 1] \rightarrow U$, such that the path $\xi(u(\cdot)) = \{x \in C_{\text{free}} \mid \xi(u(0)) = x_{\text{init}}, \xi(u(1)) \in C_{\text{free}}, \xi(u(s)) = \int_0^1 [x_{\text{init}} + f(x, u)] ds\}$.

In this paper, we focus on the trajectory planning problem for AUVs. Since the initial force has major impact on the motion of AUVs compared to ground vehicles, the kinodynamic constraints are more complex.

3. Algorithms

RRT was firstly introduced by Lavelle as a sampling-based path planning algorithm, which is shown in Algorithm 1. It has the ability of quickly searching or exploring the C-space by a random tree. At each iteration, the tree expands

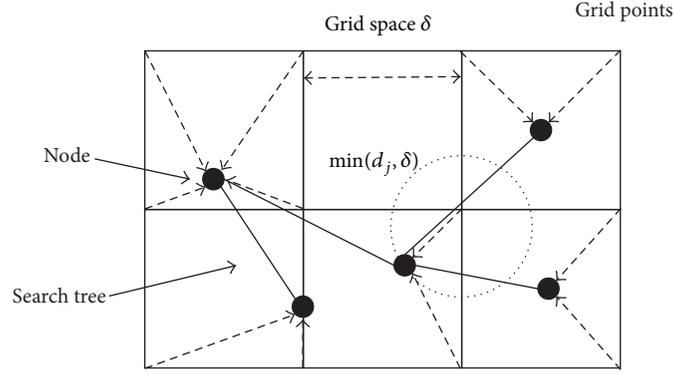


FIGURE 1: Grid point distance from the nearest node in the tree. δ is an adjustable grid spacing, and the dashed arrow indicates the distance, d_j , from the grid points to their closet nodes in tree.

Require: C-space C , Free C-space C_{free} , Initial state x_{start} , Goal region X_{goal} , max searching iteration K ;
 Ensure: Feasible path τ
 (1) $V \leftarrow (x_{\text{start}})$; $E \leftarrow \emptyset$; $\mathcal{E} \leftarrow (V, E)$; // Initialization
 (2) for $k = 1, 2, \dots, K$ do
 (3) $x_{\text{rand}} \leftarrow \text{Sample}(k)$; // Randomized sample in C_{free}
 (4) $\mathcal{E}(V, E) \leftarrow \text{Extend}(\mathcal{E}, x_{\text{rand}})$; // Extend random tree
 (5) end for
 (6) $\tau \leftarrow \text{Export}(\mathcal{E})$; // Return the feasible path
 (7) return τ ;

ALGORITHM 1: Body of basic RRT algorithm [33].

sequentially towards the goal. We refer readers to more details about RRT in [12].

The main branches of RRT are firstly constructed as it rapidly reaches the far corners of the square. As samples are incrementally added in the searching tree, C-spaces are mostly covered by smaller branches gradually. In other words, given any interval $\epsilon > 0$, C-space will be filled with nodes as the number of samples tends to infinity. It means the probability that there exists at least one node of the random tree located in the goal set equals one, as the number of samples tends to infinity, if there exists one available path in C_{free} : that is, $\lim_{n \rightarrow \infty} \mathbb{P}\{\forall x \in \mathcal{E}, \exists x' \in x_{\text{RRT}}, |x - x'| < \epsilon, \epsilon > 0\} = 1$.

3.1. Coverage of a Graph. A kind of coverage measure of the tree in C-space was proposed in [34]. We uniformly overlay grids of n_g points and spacing δ on C . Define the minimum distance from each grid point j to the nearest node in the tree as d_j . Thus, $\min(d_j, \delta)$ can describe the radius of the largest ball centered at the grid point which contains no nodes of the tree and adjacent grid points as shown in Figure 1. Given a graph \mathcal{E} , we define its coverage measure as in [34].

$$c(\mathcal{E}) = \frac{1}{\delta} \sum_{j=1}^{n_g} \frac{\min(d_j, \delta)}{n_g}, \quad (1)$$

where $1/\delta$ removes the impact of different spacing distance. The coverage of a tree can be viewed as the average of all nodes distances, normalized by the grid space δ .

The key idea of this measure is to describe the dispersion of all nodes, which is a conception proposed from the Monte Carlo method that indicates the unevenness of sample points. Primarily, the cover performance is significantly better with smaller coverage measure. In other words, expanding more widely and uniformly may lead to smaller coverage measure. In an extreme case, that is, $c(\mathcal{E}) = 0$ which is impracticable, if there exists at least one node of the tree in every grid point, the tree can be viewed well distributed in C_{free} with given solution of the grids. Intuitively, we may try to optimize $c(\mathcal{E})$ with an acceptable computing complexity, which is addressed in Section 3.4.

3.2. Growth Speed of a Tree. The growth of a tree is naturally defined as the derivative of the coverage measure. We denote the number of nodes in the tree as n_t and then define the growth of the tree as [34]

$$g(\mathcal{E}) = \frac{-\Delta c(\mathcal{E})}{\Delta n_t}. \quad (2)$$

Because the differential form $dc(\mathcal{E})/dn_t$ is intractably formed and computed, we utilize the difference form to indicate the growing speed of the tree. Obviously, $g(\mathcal{E})$ is able to be used to test the nodes of tree appropriately. Let \bar{g} denote the threshold, if $g(\mathcal{E})$ drops below \bar{g} with new node v_i , it means that v_i is almost worthless.

3.3. Liveliness of Nodes. The factors which illustrate expanding ability of nodes in a tree can be roughly categorized into two parts: the external circumstance factors and the internal circumstance factors.

3.3.1. External Circumstance of Nodes. External environment has significant influence on the expanding ability of each node. If some node locates in a blind alley, it has absolutely no contribution on searching. As shown in Figure 2, it describes the scope of a parent node that could be extended to within

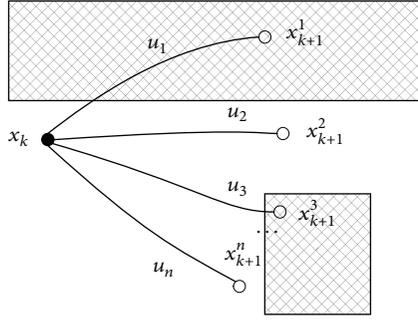


FIGURE 2: Latent expansion with obstacles.

time interval Δt . Without loss of generality, we assume that an appropriate discrete control set U is defined and the cardinality of U is m : that is, $U = u_1, u_2, \dots, u_m$. Let x_k denote the parent and let $x_{k+1}^{u_i}$, $i = 1, \dots, m$ denote the successor of x_k . The superscript u_i means the control vector applied on x_k , where the gray areas represent the obstacles. If $x_{k+1}^{u_i}$ locates in C_{obs} for all $k = 1, 2, \dots, m$, it certainly means that x_k is dead for the concurrent tree. Also, this expansion by x_k should be considered totally useless. By reducing useless nodes the searching process can be obtained more efficiently.

Based on that, the available potential successor set of x_k is achieved by applying U ; that is, $\tilde{X}_{k+1}^U = \{x_{k+1}^{u_i} \mid \text{Collision_Free}(x_k, x_{k+1}^{u_i}) = \text{TRUE}, i = 1, 2, \dots, m\}$. Meanwhile, we denote the corresponding available control vector as $\tilde{U} = \{u_i \mid \text{Collision_Free}(x_k, x_{k+1}^{u_i}) = \text{TRUE}, i = 1, 2, \dots, m\}$. We denote the cardinality of \tilde{U} as m_a and define collision detective index as $\text{Se}(x_k) = m_a/m$. Note that if $\text{Se}(x_k) = 0$, the point x_k is a dead point. If so, x_k is necessarily removed from the tree. Meanwhile, we could define the collision detective index of a tree as

$$\text{Se}(\mathcal{E}) = \frac{\sum_{k=1}^{n_t} \text{Se}(x_k)}{n_t}, \quad (3)$$

where n_t denotes the number of nodes in a tree, which is utilized to normalize the feature of the nodes. The collision detective index of a tree is the average value of all component nodes.

It is a fact that the prior information about the environment is unavailable in many applications. In such cases, the performance of searching C-space is more important. $\text{Se}(\mathcal{E})$, defined in (3), may give a reasonable solution on the process of expansion of searching tree. Generally, we prefer the nodes that make $\text{Se}(\mathcal{E})$ as large as possible or larger than a threshold $\overline{\text{Se}}$, because it means that the probability of reaching the target will stay high.

Unfortunately, it is commonly difficult to tell whether a node is stuck in a blind alley or just in a narrow passage. Note that it is still necessary to expand x_k if m_a is not equal to zero. In other words, only $\{x_k \mid m_a = 0\}$ could be removed in case of unexpected failure. In this way, the set of state $\{x_k \mid m_a = 0\}$ can be viewed as the inevitable obstacles. Unlike that, the dead nodes in this paper are naturally cheap to compute.

3.3.2. Internal Structure of Trees. Internal structure of a tree also influences the expanding ability of nodes. Due to the stochastic sampling strategy, it may lead to crowd in some regions by random expanding. In Figure 3(a), the gray circles with dotted lines are considered as regression nodes, which are avoided by RRT-blossom method proposed in [18]. Although it may be necessary to explore the regression states since the complex environment, the other unexplored regions have reasonable high priority.

In Figure 3(a), the hollow white dots are possible newly added nodes of an identical parent. The hollow white dots embraced in the ellipse together with a solid black dot are closer to the neighbor nodes than their parents. Figure 3(b) shows all expansions in the tree which avoid being closer to the existing structure. This strategy ensures that the tree will have a strong tendency to explore the whole C-space with inflated branches.

We define the degree of a node as the number of its successors, which is denoted by $d(x_k)$. All offspring of a node is defined as

$$ds(x_k) = \sum_{x_i \in \text{Us}(x_k)} d(x_i), \quad (4)$$

where the set $\text{Us}(x_k)$ is the union of all offspring of the node x_k . It is obviously that a node with more successors and offspring has less expanding ability. Figure 4 illustrates a tree's distribution of degrees and offspring.

The neighbors of a node are viewed as obstacles, which are shown in Figure 5. The neighbor nodes are viewed as circular or spherical obstacles with a fixed radius. Define the number of neighbor collisions as $m_a^{\text{nei}}(x_k) = \sum_{i=1}^m 1\{\text{Collision_Free}(x_k, x_{k+1}^{u_i}) = \text{TRUE}\}$, where $1(\cdot)$ equals 1 if equation (\cdot) holds.

Considering all these factors, we design a hybrid indicator to illustrate the quantity of liveness of each node. Synthesizing two external and internal factors, the liveness index of each node in a tree is defined in the form of

$$\begin{aligned} \text{Li}(x_k) &= \text{Se}(x_k) \exp(-m_h) \exp[-d(x_k)] \exp\left[-\frac{ds(x_k)}{n_t}\right], \end{aligned} \quad (5)$$

where n_t is the number of total nodes at current iteration. Indeed, (5) reflects the expanding ability of a specific node. We translate $\text{Li}(x_k)$ in logs denoted as LnLi .

$$\begin{aligned} \text{LnLi}(x_k) &= \ln \text{Li}(x_k) \\ &= \ln \text{Se}(x_k) - \left[m_h + d(x_k) + \frac{ds_k}{n_t} \right]. \end{aligned} \quad (6)$$

Since log function has the same monotonicity, $\text{LnLi}(x_k)$ can describe the liveness of each node. Particularly, if x_k is a dead point, that is, $\text{Se}(x_k) = 0$, we set $\text{LnLi}(x_k) = -\text{inf}$.

Obviously, the liveness LnLi defined in (6) will be attached to each node once added to the random tree. The value of LnLi is calculated and updated when the expanding process will be called. It can be viewed as a guidance that describes a better expanding direction for the searching tree.

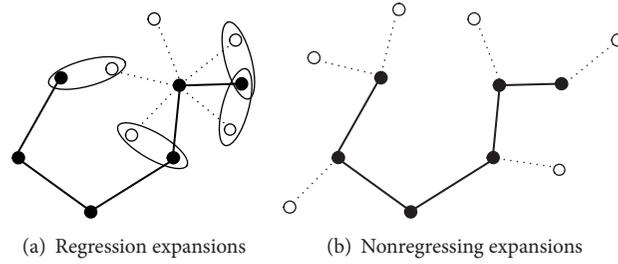


FIGURE 3: RRT-blossom regression. New point tends to stay away from the existing structure, like the hollow circles.

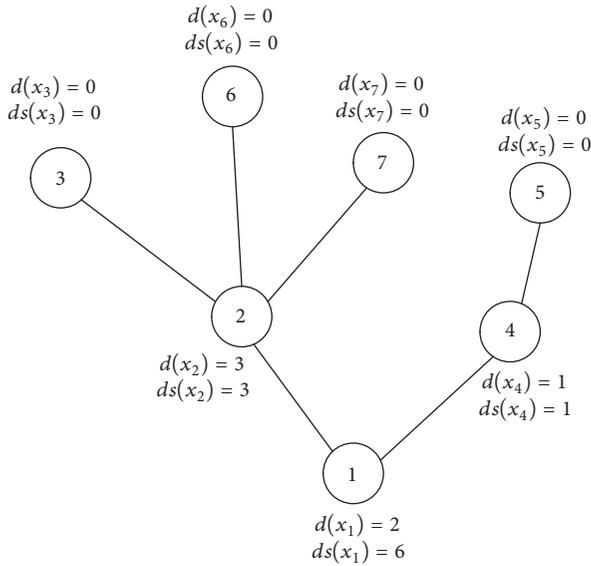


FIGURE 4: The degree of nodes in a tree.

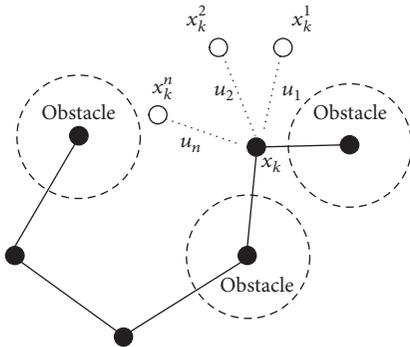


FIGURE 5: Neighbor obstacles. The radius of neighbor obstacle is generally obtained from experience.

3.4. Li-RRT. In this section, we present the hybrid RRT algorithm, that is, Li-RRT, as shown in Algorithm 2. It utilizes the liveness of each node to guide the expanding process of the random searching tree. More efficient or useful nodes will be popped out to enhance the property of exploration. The main methodology of Li-RRT is based on RRT. Different from the typical RRT, we define a liveness set $L = \{\text{LnLi}(v) \mid v \in V\}$ initially (line (1)).

At each iteration, node $\max \text{Li}$ pops with \max liveness from L in line (3). Then an input vector is randomly selected from function RandU (line (5)), according to which a random state in C_{free} is generated. Meantime, x_{rand} is chosen from X_{goal} with probability \mathcal{W} (line (13)). It is proposed as a goal bias sampling method, which is simple but effective. Similarly, the nearest node is calculated given the cost function $c(\cdot)$. New node x_{new} then is achieved based on forward processing drive (line (20)). CollisionFree is called to verify the validity of x_{new} and $(x_{\text{nearest}}, x_{\text{new}})$ (line (21)). Expanding process is implemented by adding x_{new} and $(x_{\text{nearest}}, x_{\text{new}})$ (line (22)).

Liveness of each node is updated by $\text{UpdateLi}(v)$ in Algorithm 3. From the definition of $\text{LnLi}(x_k)$ (see (5)), we only need to care about three nodes sets, that is, $\{V_{\text{ch}}^1, V_{\text{ch}}^2, V_{\text{ch}}^3\}$. V_{ch}^1 contains only x_{new} , since $\text{Se}(x_k)$ is constant and is determined based on the state and the environments. Here we define that “all parent nodes” of x_k are the nodes set $\{v \mid x_k \text{ can be traced from } v\}$ different from the parent node $x_{\text{par}}(x_k)$. V_{ch}^2 includes exactly all parent nodes of x_{new} . Because x_{new} increases the successors of $v \in V_{\text{ch}}^2$, we set $\text{LnLi}(v \in V_{\text{ch}}^2) = \text{LnLi}(v) + (ds(v)/n_t - (ds(v) + 1)/(n_t + 1))$. Specifically, we set the liveness of $x_{\text{parx}}(x_{\text{new}})$ as $\text{LnLi}(x_{\text{parx}}) = \text{LnLi}(x_{\text{parx}}) - 1$. V_{ch}^3 is the near set of x_{new} : that is, $V_{\text{ch}}^3 = \{v \mid c(v, x_{\text{new}}) \leq \mathcal{H}\}$. m_h of $v \in V_{\text{ch}}^3$ will increase because of x_{new} . Thus, we define $\text{LnLi}(v \in V_{\text{ch}}^3) = \text{LnLi}(v) - 1$.

4. Analysis of the Algorithms

To prove the following theorems, we remind the definition of attraction sequence [35]. Let $\mathbb{A} = \{A_0, A_1, \dots, A_k\}$ be a finite sequence of sets as follows. (i) For each A_i , there exists a set $A_i \subseteq B_i \subseteq C_{\text{free}}$ called the basin of A_i , and $\forall x \in A_{i-1}$ and $y \in A_i$ and $z \in C_{\text{free}} \setminus B_i$, $\|x - y\| \leq \|x - z\|$ holds. (ii) $\forall x \in B_i$, there exists an m such that the sequence of action $\{u_1, u_2, \dots, u_m\}$ selected by LOCAL_PLANNER algorithm will bring the point into $A_i \subseteq B_i$. (iii) $A_0 = \{x_{\text{start}}\}$ and $A_k = \{X_{\text{goal}}\}$.

An attractor region A_i like a funnel as a metaphor for motions converges into a small region in the space. As shown in Figure 6, a basin B_i can be viewed as a safety zone which ensures that a point of B_i will be selected by the NEAREST_NEIGHBOR and potential well which attracts the point into A_i . Given an attraction sequence \mathbb{A} with length k and letting $p_i = \mu(A_i)/\mu(C_{\text{free}})$ and $p_m = \min p_i$, we have the following lemma.

```

Require: C-space  $C$ , Obstacle space  $C_{\text{obs}}$ , Initial point
 $x_{\text{start}}$ , Goal region  $X_{\text{goal}}$ , Discrete input set  $U = [u_1, u_2, \dots, u_f]$ ,
Maximum search steps  $K$ ;
Ensure: Feasible trajectory  $\xi$ 
(1)  $V \leftarrow x_{\text{start}}$ ;  $E \leftarrow \emptyset$ ;  $\mathcal{G} \leftarrow (V, E)$ ; Liveness set  $L = \{\text{Li}(x_{\text{start}})\}$ 
// Initialization
(2) for  $k = 1, \dots, K$  do
(3)    $\max \text{Li} \leftarrow \text{argmax}_{v \in V} \text{LnLi}(v)$ 
(4)   if  $\text{Rand}() < \mathcal{B}$  then
(5)      $u_{\text{rand}} \leftarrow \text{RandU}(u_1, \dots, u_f)$ 
(6)      $x_{\text{rand}} \leftarrow \text{drive}(\max \text{Li}, u_{\text{rand}})$ ;  $x_{\text{nearest}} \leftarrow \max \text{Li}$ 
(7)     for each  $x_{\text{near}} \in X_{\text{near}}$  do
(8)       if  $c(x_{\text{near}}, x_{\text{rand}}) < c(\max \text{Li}, x_{\text{rand}})$  then
(9)          $x_{\text{nearest}} \leftarrow x_{\text{near}}$ 
(10)      end if
(11)    end for
(12)  else
(13)    if  $\text{Rand}() < \mathcal{W}$  then
(14)       $x_{\text{rand}} \leftarrow x_{\text{goal}} \in X_{\text{goal}}$ 
(15)    else
(16)       $x_{\text{rand}} \leftarrow \text{RandProc}(C_{\text{free}})$ 
(17)       $x_{\text{nearest}} \leftarrow \text{Nearest}(x_{\text{rand}})$ 
(18)    end if
(19)  end if
(20)  $x_{\text{new}} \leftarrow \text{drive}(x_{\text{nearest}}, x_{\text{rand}})$ 
(21) if  $\text{CollisionFree}(x_{\text{nearest}}, x_{\text{rand}})$  then
(22)    $V \leftarrow V \cup x_{\text{new}}$ ;  $E \leftarrow E \cup (x_{\text{nearest}}, x_{\text{new}})$ 
(23) end if
(24) if  $x_{\text{new}} \in X_{\text{goal}}$  then
(25)   objective = TRUE
(26) end if
(27) for all  $v \in V$  do
(28)   UpdateLi( $v$ )
(29) end for
(30) end for
(31) return  $\xi$  if objective = TRUE; else failure

```

ALGORITHM 2: Liveness-based RRT algorithm.

```

Require: Near set threshold  $\mathcal{N}$ ; Three nodes set
 $\{V_{\text{ch}}^1, V_{\text{ch}}^2, V_{\text{ch}}^3\}$ 
(1)  $\text{LnLi}(x_{\text{new}}) \leftarrow \text{In Se}(x_{\text{new}}) + [-m_h(x_{\text{new}})]$ 
(2) for  $v \in V_{\text{ch}}^2$  do
(3)    $\text{LnLi}(v) \leftarrow \text{LnLi}(v) + (ds(v)/n_t - (ds(v) + 1)/(n_t + 1))$ 
(4) end for
(5)  $x_{\text{par}} \leftarrow \text{parent}(x_{\text{new}})$ 
(6)  $\text{LnLi}(x_{\text{par}}) \leftarrow \text{LnLi}(x_{\text{par}}) - 1$ 
(7) for  $v \in V_{\text{ch}}^3 = \text{Near}(x_{\text{new}})$  do
(8)    $\text{LnLi}(v) \leftarrow \text{LnLi}(v) - 1$ 
(9) end for

```

ALGORITHM 3: Update liveness algorithm.

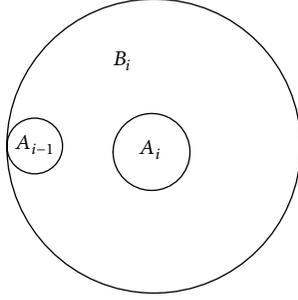


FIGURE 6: Attraction sequence.

Lemma 3 (see [36]). Let C_1, C_2, \dots, C_n be independent Poisson trials such that, for $1 \leq i \leq n$, $\Pr[C_i = 1] = p_i$, where $0 < p_i < 1$. Then, for $\mathcal{C} = \sum_{i=1}^n C_i$, $\mu = E[\mathcal{C}] = \sum_{i=1}^n p_i$ and $0 < \delta \leq 1$,

$$\Pr[\mathcal{C} < (1 - \delta)\mu] < \exp\left(-\frac{\mu\delta^2}{2}\right). \quad (7)$$

Lemma 4 (see [35]). If an attraction sequence of length k exists, for a constant δ , the probability of basic RRT algorithm fails to find a trajectory after n iterations are at most $\exp[-(1/2)(np_m - 2k)]$.

Let sequence $\{p'_1, p'_2, \dots, p'_n\}$ be an ascending order that $p'_1 \leq p'_2 \leq \dots \leq p'_n$, where $p'_i \in \{p_1, p_2, \dots, p_n\}$, $p'_1 = \min_i p_i$ and $p'_n = \max_i p_i$; the following theorem can be achieved.

While C_{free} is partitioned into m connected regions, motion planning problem for multiple AUVs could be considered as an m goals version of Problem 2. $\mathbb{A}^{(i)} = \{A_1^{(i)}, A_2^{(i)}, \dots, A_{k_i}^{(i)}\}$ is an attraction sequence which connects x_{start} and X_{goal} , where k_i is the length of attraction sequence. Let $p_j^{(i)} = \mu(A_j^{(i)})/\mu(C_{\text{free}})$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, k_i$, $p_m = \min_{ij} p_j^{(i)}$; we have the following theorem.

Theorem 5. If m attraction sequences of length k_i , $i = 1, 2, \dots, m$ exist, the probability of basic RRT algorithm fails to find m trajectories after n iterations are at most $\exp[-(1/2)(np_m - 2\sum_{i=1}^m k_i)]$.

Theorem 6. If m attraction sequences of length k_i , $i = 1, 2, \dots, m$ exist, the probability of liveliness-based RRT algorithm fails to find m trajectories after n iterations are at most $\exp[(1/n)\sum_{i=1}^m \sum_{j=1}^{n_i} ((p_j^{(i)}/p_m)(-(1/2)np_m + \sum_{i=1}^m k_i))]$.

Proof. Let sequence $(p_1^{(i)}, p_2^{(i)}, \dots, p_{n_i}^{(i)})$ be m ascending orders satisfying $(p_1^{(i)} \leq p_2^{(i)} \leq \dots \leq p_{n_i}^{(i)})$, where $p_1^{(i)} = \min_j p_j^{(i)}$, and $p_{n_i}^{(i)} = \max_j p_j^{(i)}$, $j = 1, 2, \dots, n_i$, $i = 1, 2, \dots, m$. While using the strategy of eliminating the “useless” nodes, the random variable \mathcal{C} is viewed as Poisson trails with probability distribution $(p_j^{(i)})$, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n_i$. \mathcal{C} is viewed as Poisson trails with probability distribution $p_j^{(i)}$; then $\mu = E[\mathcal{C}] = \sum_{i=1}^m \sum_{j=1}^{n_i} p_j^{(i)}$, and $0 < \delta \leq 1$.

According to Lemma 3, $\Pr[\mathcal{C} < (1 - \delta)\mu] < \exp(-\mu\delta^2/2)$, where $\delta = 1 - \sum_{i=1}^m k_i/(np_m)$. In addition,

$$\begin{aligned} \frac{-\mu\delta^2}{2} &= -\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^{n_i} \frac{p_j^{(i)}}{p_m} \left(1 - \frac{\sum_{i=1}^m k_i}{np_m}\right)^2 \\ &= \frac{1}{n} \sum_{i=1}^m \sum_{j=1}^{n_i} \frac{p_j^{(i)}}{p_m} \left(-\frac{1}{2}np_m + \sum_{i=1}^m k_i - \frac{(\sum_{i=1}^m k_i)^2}{2np_m}\right) \\ &< \frac{1}{n} \sum_{i=1}^m \sum_{j=1}^{n_i} \frac{p_j^{(i)}}{p_m} \left(-\frac{1}{2}np_m + \sum_{i=1}^m k_i\right). \end{aligned} \quad (8)$$

This completes the proof. \square

Proposition 7. Given specific C , C_{free} , x_{start} , and a set of goal regions $\{X_{1\text{goal}}, X_{2\text{goal}}, \dots, X_{m\text{goal}}\}$, while solving Problem 2, the liveliness-based RRT algorithm always has smaller failing probability than the basic RRT algorithm.

Proof. Note that $(1/n)\sum_{i=1}^m \sum_{j=1}^{n_i} (p_j^{(i)}/p_m) > 1$, according to Theorems 5 and 6,

$$\begin{aligned} \Pr[\text{LiRRT}] &= \exp\left[\frac{1}{n} \sum_{i=1}^m \sum_{j=1}^{n_i} \frac{p_j^{(i)}}{p_m} \left(-\frac{1}{2}np_m + \sum_{i=1}^m k_i\right)\right] \\ &< \exp\left[-\frac{1}{2} \left(np_m - 2\sum_{i=1}^m k_i\right)\right] \\ &= \Pr[\text{bRRT}]. \end{aligned} \quad (9)$$

\square

In summary, we can see that Li-RRT enhances the expanding efficiency from Proposition 7.

5. Applications of Li-RRT

In this section, we present a numerical simulation for planar AUV with data from NOAA. A region of Hawaii, that is, a rectangle from $157^\circ 58' 48''\text{W}$, $21^\circ 20' 24''\text{N}$ to $157^\circ 57' 36''\text{W}$, $21^\circ 21' 36''\text{N}$, is utilized to testify the effectiveness of Li-RRT; see Figure 7(a). Without loss of generality, we assume that AUV only voyages at depth of 5 meters at least. And the dynamics of AUV is $\dot{x} = v \cos(\theta)$, $\dot{y} = v \sin(\theta)$, $\dot{\theta} = \omega$. Meantime, we set $0 \leq |v| \leq 2$ m/s and $0 \leq |\omega| \leq 0.15$ rad/s.

We can see that a valid path is found by Li-RRT shown as in Figure 7(b). Although solutions returned by Li-RRT are mostly not optimal, it can supply candidate paths effectively in less planning time. A comparison between RRT and Li-RRT is described in Figure 8. Li-RRT obviously costs less searching iterations by utilizing existing information between current random tree and the environment.

Here we take multiple simulations by RRT with same conditions and present two better solutions. We can see that RRT needs more searching iterations than Li-RRT averagely. Sampling strategy in typical RRT utilizes only goal bias method which tries to make the random searching

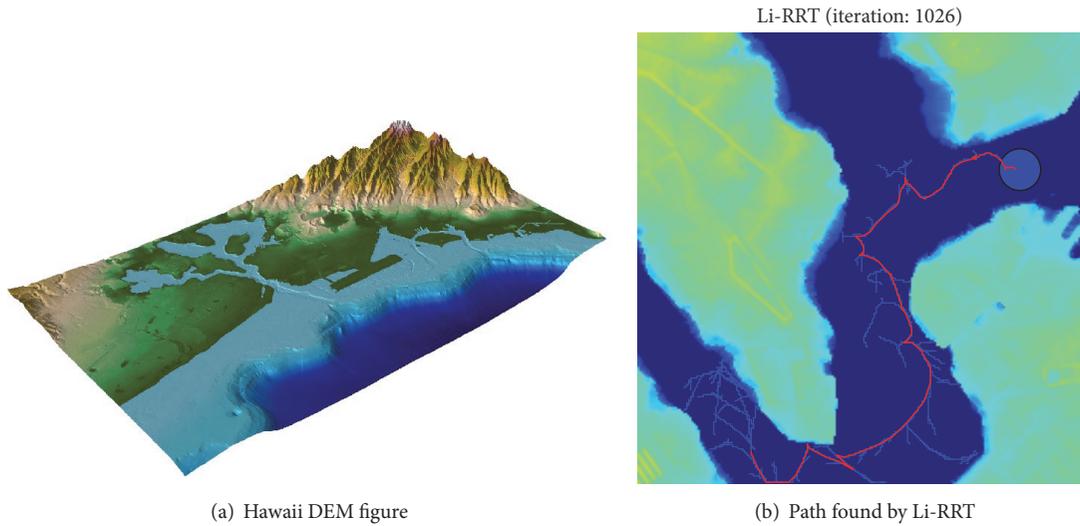


FIGURE 7: Start position locates at $157^{\circ}58'30''\text{W}$, $21^{\circ}20'29''\text{N}$; goal locates at $157^{\circ}57'47''\text{W}$, $21^{\circ}21'14''\text{N}$, shown as a solid blue circle. Red line describes a valid path from start to goal.

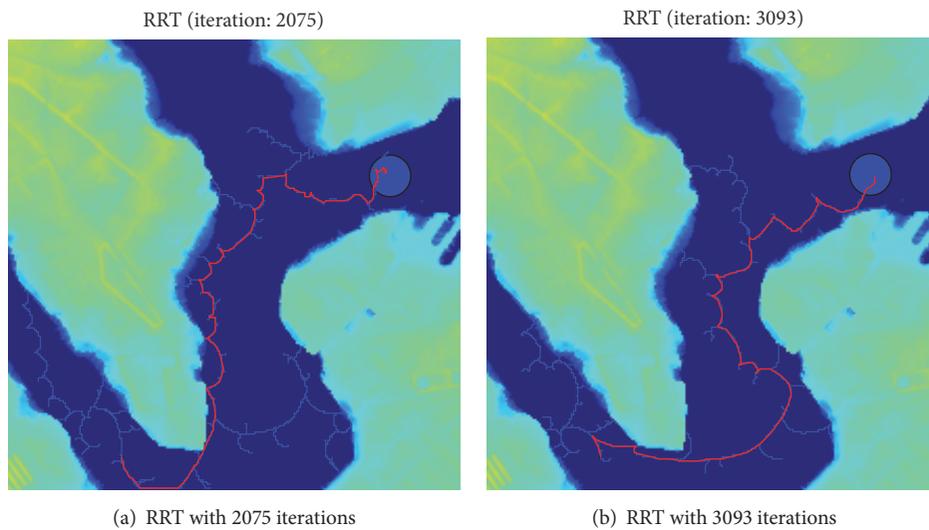


FIGURE 8: Planning with the same conditions by RRT. Start position locates at $157^{\circ}58'30''\text{W}$, $21^{\circ}20'29''\text{N}$; Goal locates at $157^{\circ}57'47''\text{W}$, $21^{\circ}21'14''\text{N}$, shown as a solid blue circle. Red line describes a valid path from start to goal.

tree towards the goal region. It certainly may generate useless sampling nodes in every searching step. For example, sampling that avoids an oversampled areas seems more efficient; however, typical RRT has no such ability. Thus, RRT needs more searching time or iterations to reach the goal.

6. Conclusions

In this paper, we have presented an analysis of rapidly exploring random trees and defined the several indicators of a tree, including the coverage and the growth speed describing the growing behavior of a tree. Considering the external and internal factors which influence the expanding ability,

we also have defined indicators collision detective index, degree, offspring, and neighbor collisions. On this basis, we have proposed Li-RRT for AUVs motion planning. By using the tools attraction sequence, theoretical analysis has indicated that liveliness-based RRT enhances the expanding speed. Simulations are also provided to show the effectiveness of Li-RRT. It has been shown that Li-RRT performs well in finite planning time. Moreover, the growth direction of a tree also reflects some properties of a tree's expanding ability which could be utilized by RRT to conduct the expanding procedure. One of the possible research directions will focus on the planning without any prior information in the uncertain environment considering such indicators proposed in this work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61472325 and 61273333.

References

- [1] Y. J. Heo and W. K. Chung, "RRT-based path planning with kinematic constraints of AUV in underwater structured environment," in *Proceedings of the 10th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI '13*, pp. 523–525, November 2013.
- [2] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: a general overview," *Mechanisms and Machine Science*, vol. 29, pp. 3–27, 2015.
- [3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '85)*, vol. 2, pp. 500–505, 1985.
- [4] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.
- [5] C. M. Saaj, V. Lappas, and V. Gazi, "Spacecraft swarm navigation and control using artificial potential field and sliding mode control," in *Proceedings of the 2006 IEEE International Conference on Industrial Technology, ICIT '06*, pp. 2646–2651, 2007.
- [6] P. F. J. Lermusiaux, T. Lolla, P. J. H. et al., *Science of Autonomy: Time-Optimal Path Planning and Adaptive Sampling for Swarms of Ocean Vehicles*, Springer International Publishing, Gewerbestrasse, Switzerland, 2016.
- [7] S. Carpin and G. Pillonetto, "Merging the adaptive random walks planner with the randomized potential field planner," in *Proceedings of the in Proceedings of the Fifth International Workshop on Robot Motion and Control, 2005*, pp. 151–156, 2005.
- [8] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "MAPRM: a probabilistic roadmap planner with sampling on the medial axis of the space," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1024–1031, May 1999.
- [9] L. Han and N. M. Amato, "A kinematics-based probabilistic roadmap method for high DOF closed chain systems," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 473–478, 2004.
- [10] V. Boor, M. H. Overmars, and A. F. V. D. Stappen, "Gaussian sampling strategy for probabilistic roadmap planners," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation, ICRA '99*, pp. 1018–1023, May 1999.
- [11] R. Cui, B. Gao, and J. Guo, "Pareto-optimal coordination of multiple robots with safety guarantees," *Autonomous Robots*, vol. 32, no. 3, pp. 189–205, 2012.
- [12] S. M. LaValle, "Rapidly-exploring random trees: a new tool for path planning," Tech. Rep. 98-11, Computer Science Dept., Iowa State University, Iowa, Iowa, USA, 1998.
- [13] J. Kim and J. P. Ostrowski, "Motion planning a aerial robot using rapidly-exploring random trees with dynamic constraints," in *Proceedings of the in IEEE International Conference on Robotics and Automation, 2003*, vol. 2, pp. 2200–2205, 2003.
- [14] D. Shim, H. Chung, H. J. Kim, and S. Sastry, "Autonomous exploration in unknown urban environments for unmanned aerial vehicles," in *Proceedings of the in AIAA GNC Conference*, pp. 1–8, 2005.
- [15] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-auv path planning for scalar field sampling using multidimensional RRT," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2016.
- [16] T. Van Huynh, M. Dunbabin, and R. N. Smith, "Convergence-guaranteed time-varying RRT path planning for profiling floats in 4-dimensional flow," in *Proceedings of the Australasian Conference on Robotics and Automation, ACRA '14*, December 2014.
- [17] J. L. Blanco, M. Bellone, and A. Gimenez-Fernandez, "TP-space RRT - Kinematic path planning of non-holonomic any-shape vehicles," *International Journal of Advanced Robotic Systems*, vol. 12, article no. A55, 2015.
- [18] M. Kalisiak and M. van de Panne, "RRT-blossom: RRT with a local flood-fill behavior," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 1237–1242, May 2006.
- [19] J. M. Esposito, J. Kim, and V. Kumar, *Adaptive RRTs for Validating Hybrid Robotic Control Systems*, Springer, Berlin, Germany, 2005.
- [20] J. Si and C. Chin, "An adaptable walking-skid for seabed ROV under strong current disturbance," *Journal of Marine Science and Application*, vol. 13, no. 3, pp. 305–314, 2014.
- [21] R. Cui, L. Chen, C. Yang, and M. Chen, "Extended state observer-based integral sliding mode control for an underwater robot with unknown disturbances and uncertain nonlinearities," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6785–6795, 2017.
- [22] M. D. Thekkedan, C. S. Chin, and W. L. Woo, "Virtual reality simulation of fuzzy-logic control during underwater dynamic positioning," *Journal of Marine Science and Application*, vol. 14, no. 1, pp. 14–24, 2015.
- [23] C. S. Chin, W. P. Lin, and J. Y. Lin, "Experimental validation of open-frame rov model for virtual reality simulation and control," *Journal of Marine Science Technology*, vol. no. 2, p. 21, 2017.
- [24] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [25] O. Salzman and D. Halperin, "Asymptotically near-optimal RRT for fast, high-quality motion planning," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 473–483, 2013.
- [26] M. Otte and E. Frazzoli, "RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning," *International Journal of Robotics Research*, vol. 35, no. 7, pp. 797–822, 2016.
- [27] Y. Li, Z. Littlefield, and K. E. Bekris, "Sparse methods for efficient asymptotically optimal kinodynamic planning," *Springer Tracts in Advanced Robotics*, vol. 107, pp. 263–282, 2015.
- [28] D. J. Webb and J. Van Den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," Computer Science, 2012.
- [29] P. Pharpatara, B. Herisse, and Y. Bestaoui, "3-D trajectory planning of aerial vehicles using RRT*," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1116–1123, 2017.

- [30] D. Lee, H. Song, and D. H. Shim, "Optimal path planning based on spline-RRT* for fixed-wing UAVs operating in three-dimensional environments," in *Proceedings of the 14th International Conference on Control, Automation and Systems, ICCAS '14*, pp. 835–839, October 2014.
- [31] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: a review," *IEEE Access*, vol. 2, pp. 56–77, 2014.
- [32] V. Boor, M. H. Overmars, and A. F. V. D. Stappen, *Gaussian Sampling for Probabilistic Roadmap Planners*, Utrecht University The Netherlands, Utrecht, Netherlands, 2001.
- [33] S. Karaman and E. Frazzoli, "Sampling-based motion planning with deterministic μ -calculus specifications," in *Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 2222–2229, December 2009.
- [34] J. M. Esposito, J. Kim, and V. Kumar, "Adaptive RRTs for Validating Hybrid Robotic Control Systems," in *Algorithmic Foundations of Robotics VI*, vol. 17 of *Springer Tracts in Advanced Robotics*, pp. 107–121, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [35] S. M. LaValle and J. J. Kuffner Jr., "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
- [36] R. Motwani, *Randomized Algorithms*, Cambridge University Press, Cambridge, UK, 1995.

Research Article

Reactive Path Planning Approach for Docking Robots in Unknown Environment

Peng Cui, Weisheng Yan, and Yintao Wang

School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

Correspondence should be addressed to Weisheng Yan; wsysan@nwpu.edu.cn

Received 13 May 2017; Accepted 6 July 2017; Published 12 September 2017

Academic Editor: Cheng S. Chin

Copyright © 2017 Peng Cui et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Autonomous robots need to be recharged and exchange information with the host through docking in the long-distance tasks. Therefore, feasible path is required in the docking process to guide the robot and adjust its pose. However, when there are unknown obstacles in the work area, it becomes difficult to determine the feasible path for docking. This paper presents a reactive path planning approach named Dubins-APF (DAPF) to solve the path planning problem for docking in unknown environment with obstacles. In this proposed approach the Dubins curves are combined with the designed obstacle avoidance potential field to plan the feasible path. Firstly, an initial path is planned and followed according to the configurations of the robot and the docking station. Then when the followed path is evaluated to be infeasible, the intermediate configuration is calculated as well as the replanned path based on the obstacle avoidance potential field. The robot will be navigated to the docking station with proper pose eventually via the DAPF approach. The proposed DAPF approach is efficient and does not require the prior knowledge about the environment. Simulation results are given to validate the effectiveness and feasibility of the proposed approach.

1. Introduction

Nowadays autonomous mobile robots such as autonomous drones, autonomous underwater vehicles, and automated vehicles are widely used in the complex environment and undertake the dangerous and heavy tasks [1–8]. However, their durations are constrained by their limited battery capacities and data storage spaces [9]. To solve this problem docking stations are designed and deployed in the work areas of the robots to maintain them in practical applications [10, 11]. To ensure the safety and the success in the docking process, a feasible path is required to be planned firstly [12]. This is because, for one thing, the area for docking may be unknown in advance or dynamically changing and there may be static and moving obstacles that threaten the safety of the robots [12, 13]. For another, the final configuration (pose and velocity) of the robot should be adjusted properly to avoid the impact with the docking station. The kinematic characters of the robot should be considered as well for path tracking and energy saving [14–16].

In recent years, a variety of approaches have been proposed to solve the path planning problem in unknown

environment. Among them a biologically inspired neural network approach is presented in [17] to plan path for the autonomous underwater vehicle in unknown two-dimension (2D) environment, which is achieved via updating the environment maps according to Dempster-Shafer theory in steps. In [18] the online path planning problem with prescribed target in environment with unknown obstacles is considered and the neural networks trained by the reinforcement learning approach are adopted to solve this problem. In [19] the rapidly exploring random trees star (RRT*) algorithm is employed to plan the path for autonomous underwater vehicles, where the mutual information between the scalar field model and observations is used to improve the path planning result. In [20] the path planning problem for household robot in unknown environment is considered and solved by the modified artificial potential field (APF) method based on the motion characteristics of household animals. In [21] the collision-free path planning for autonomous container truck is achieved via utilizing the improved ant colony optimization (ACO) algorithm. In this algorithm the local path is generated according to the selected local target which is determined and updated by the rolling window

approach. In [22] a dynamic planning algorithm is presented to determine the collision-free path for the mobile terrestrial robot in unknown environment. In this algorithm the local objectives are determined by the genetic algorithm (GA) and the optimum routes are generated dynamically towards the global object. It can be concluded from the efforts above that the efficiency of the path planning approach is considered emphatically in the unknown environment. Meanwhile the reactive frameworks are adaptive in solving the path planning problem in the unknown environment in spite of the fact that different various local path planning approaches such as modified bioinspired method, RRT algorithm, and APF method are adopted. However, the smoothness of the planned path is rarely considered in these efforts as well as the pose and velocity of the robot, which is critical in the docking tasks [23].

In this paper, a reactive path planning approach named Dubins-APF is proposed to solve the path planning problem for docking in unknown environment based on combination of the Dubins curves and the APF approach. Dubins curves have been proved to be the optimal paths with minimal turning radius that connect two points with prescribed poses in 2D space [24, 25]. However, if there are obstacles in the environment for docking, it is difficult to determine the feasible path based on Dubins curves [26, 27]. Since the APF approach is efficient in obstacle avoidance, in this paper it is combined with the Dubins curves to determine the feasible path for docking [28–31]. The proposed path planning approach works in a reactive mode which is described as follows. Once the planned path is infeasible, feasible intermediate configurations are determined based on the obstacle avoidance potential field according to the configurations of the detected obstacles. Then feasible Dubins curves are generated as the replanned path based on the intermediate configuration. Through implementing this path planning-replanning strategy continually, the DAPF algorithm will solve the path planning problem for docking.

The main contributions of this paper are as follows:

- (1) The geometrical approach to determine the 3D Dubins curves is proposed in this paper. It can be utilized to generate the docking path and evaluate the feasibility of the planned path as well.
- (2) The conception of combining the advantages of the Dubins curves and the artificial potential field is proposed and implemented in this paper to improve the quality of the docking path while avoiding obstacles.

The structure of this paper is presented as follows. The path planning problem for docking and the notion of the DAPF approach are introduced in Section 1. The problem statement and the path generation approach with 3D Dubins curves are described in Section 2. In Section 3 the Dubins-APF path planning approach is proposed and illustrated in detail. The simulation result and discussion about the DAPF approach are presented in Section 4. Some conclusions and future works are provided in Section 5.

2. Path Planning with 3D Dubins Curves

2.1. Problem Statement. The path planning problem for docking is to determine a collision-free path to connect the initial position of the autonomous robot and the docking station with prescribed poses under certain constraints of the robot [28]. In this paper the docking station with unidirectional entrance is considered and it is assumed to be static. The position of the docking station is written as P_d and the direction of its entrance is expressed as \vec{V}_d . Meanwhile the velocity of the autonomous robot is assumed to be constant and its position and velocity are written as P_r and \vec{V}_r , where $\|\vec{V}_r\| = v$. The turning ability of the robot is assumed to be limited and the minimal turning radius is written as R_t . Hence to avoid collision with the docking station, the final position and velocity of the robot should be close to P_d and \vec{V}_d . Additionally the work area for docking is assumed to be unknown in advance, which means the autonomous robot can only acquire the environment information within its sensor range R_s .

2.2. 3D Dubins Curves. The Dubins curve only consists of two kinds of segments which are the circle (C) segment and the straight-line (S) segment, where the radius of the C segment is equal to the minimal turning radius of the robot and this curve is smooth at the intersections of the adjacent segments. In 2D environment the optimality of the two Dubins curves has been proved and the CCC curve or the CSC curve is reckoned as the shortest path [24]. However, the determination of the Dubins curve becomes complex in 3D environment due to the increase of the dimensionality, which means not all the segments of the Dubins curves are coplanar [32, 33]. Therefore considering the efficiency of its application in path planning, a 3D Dubins curve determination approach is presented in this paper based on the geometric characters of the typical CSC curve.

The typical CSC curve is shown in Figure 1. The initial configuration of position and pose (green vector) and the final configuration (red vector) are presented as $[P_1, \vec{V}_1]$ and $[P_2, \vec{V}_2]$ and the feasible path that connects them is a 3D Dubins curve which consists of one S segment and two C segments. The C segments C_1 (red arc) and C_2 (green arc) are two circular arcs with centers O_1 and O_2 and radii R_1 and R_2 , respectively. They are connected by the S segment (blue straight line) with intersections P_5 and P_6 separately. To indicate the coplanarity of the Dubins curve, two auxiliary lines L_1 and L_2 are drawn as Figure 1 shows. L_1 has P_1 on it and parallels with \vec{V}_1 and L_2 has P_2 on it and parallels with \vec{V}_2 . They intersect with the elongation of the S segment at P_3 and P_4 , respectively. Based on the spatial relations of L_1 , L_2 , and the S segment, the segments of the CSC curve can be divided into two planes determined by L_1 and the S segment and L_2 and the S segment separately. The intersection line of these two planes is the S segment.

Inspired by the geometric relations of these two planes, the determination process of the 3D Dubins curve is implemented as follows. The intersections of L_1 , L_2 , and the elongation of S are presented as

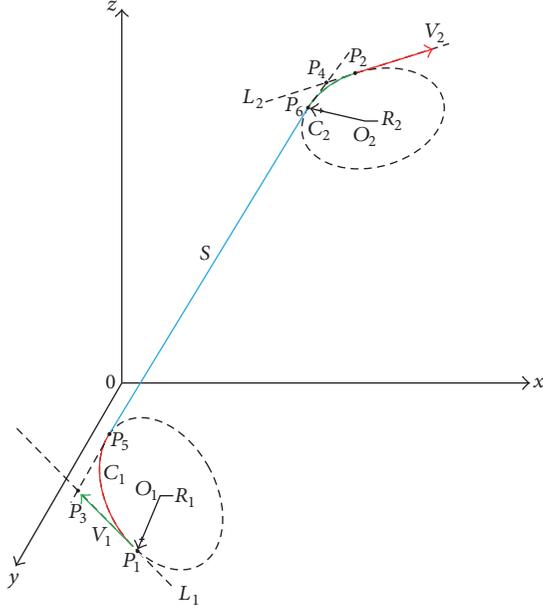


FIGURE 1: The illustration of the typical 3D CSC curve.

$$\begin{aligned} P_3 &= P_1 + k_3 \vec{V}_1, \\ P_4 &= P_2 + k_4 \vec{V}_2, \end{aligned} \quad (1)$$

where k_3 and k_4 are nonzero constants. A special case of (1) should be noticed where either k_3 or k_4 is infinite. It means that the corresponding C segment is semicircular arc and this case will be discussed later. Once k_3 and k_4 are determined, the intersections of C_1 and C_2 with S can be expressed as

$$\begin{aligned} P_5 &= P_3 + k_5 \vec{V}_3, \\ P_6 &= P_4 + k_6 \vec{V}_3, \end{aligned} \quad (2)$$

where k_5 and k_6 are constants and \vec{V}_3 is the vector that indicates the moving direction of the robot on the S segment. Since the Dubins curve is smooth at the intersections of every two adjacent segments, \vec{V}_3 can be obtained

$$\begin{aligned} \vec{V}_3 &= \text{sign}(k_3) \mathbf{g}(\overrightarrow{O_1 P_1} \times \overrightarrow{O_1 P_5} \times \overrightarrow{O_1 P_3}) \\ &= -\text{sign}(k_4) \mathbf{g}(\overrightarrow{O_2 P_6} \times \overrightarrow{O_2 P_2} \times \overrightarrow{O_2 P_4}) = \mathbf{g}(\overrightarrow{P_5 P_6}), \end{aligned} \quad (3)$$

where $\text{sign}(s)$ presents the sign function which returns the sign of the variable s and $\mathbf{g}(\vec{S})$ is the function which normalizes the vector \vec{S} and it is defined as

$$\mathbf{g}(\vec{S}) = \begin{cases} \frac{\vec{S}}{\|\vec{S}\|}, & \text{if } \|\vec{S}\| \neq 0, \\ [0, 0, 0], & \text{if } \|\vec{S}\| = 0. \end{cases} \quad (4)$$

TABLE 1: Points for other Dubins curves.

Type	S	C	CS	SC	CSC
Points	P_1, P_2	P_1, P_2	P_1, P_5, P_2	P_1, P_6, P_2	P_1, P_5, P_6, P_2
Constants	/	O_1, P_3	O_1, P_3	O_2, P_4	O_1, P_3, P_4, O_2
Parameters	/	k_1, k_3	k_1, k_3, k_5	k_2, k_4, k_6	k_1, k_2, \dots, k_6

Due to the fact that P_5 and P_6 are the tangent points of C_1 and C_2 with the S segment, there are

$$\begin{aligned} \overrightarrow{P_5 O_1} \times \vec{V}_3 &= 0, \\ \overrightarrow{P_1 O_1} \times \vec{V}_1 &= 0, \end{aligned} \quad (5)$$

$$\overrightarrow{P_6 O_2} \times \vec{V}_3 = 0,$$

$$\overrightarrow{P_2 O_2} \times \vec{V}_2 = 0,$$

$$\overrightarrow{P_5 O_1} = \overrightarrow{P_1 O_1} = R_1, \quad (6)$$

$$\overrightarrow{P_6 O_2} = \overrightarrow{P_2 O_2} = R_2.$$

Furthermore, it can be concluded from (6) that $\overrightarrow{P_3 O_1}$ and $\overrightarrow{P_4 O_2}$ are the angle bisector of $\angle P_3 P_1 O_1$ and $\angle P_4 P_2 O_2$; therefore there are

$$O_1 = P_3 + k_1 \text{sign}(k_3) (-\vec{V}_1 + \vec{V}_3) \quad (7)$$

$$O_2 = P_4 + k_2 \text{sign}(k_4) (-\vec{V}_2 + \vec{V}_3).$$

The parameters k_1 – k_6 can be calculated based on the specified set of equations to determine the CSC curve by substituting the variables in (5) and (6) with corresponding expression in (1)–(3) and (7) if there is no semicircular C segment in the CSC curve. Likewise, other types of Dubins curves, such as S , C , CS , and SC curves, can be determined easily due to the coplanarity of their initial and final configurations and segments. The key points and constants for their determinations are presented in Table 1 and the set of equations for calculation can be obtained as well based on the presented equations above.

If the CSC curve has semicircular C segments, the equations for calculation are built through modifying the corresponding equations above based on the curve's geometric characters. For example, for a CSC curve with semicircular C_1 , the modified equations for its calculation are presented as

$$\vec{V}_3 = -\vec{V}_1, \quad (8)$$

$$\overrightarrow{P_5 O_1} = \overrightarrow{P_1 O_1}.$$

Meanwhile, (2) is substituted by

$$P_6 = P_5 + k_5 \vec{V}_3, \quad (9)$$

where k_5 is constant that indicates the length of the S segment. With these equations there will be sufficient equations to determine the feasible CSC Dubins curve. The other Dubins curves can be determined by a similar technique.

2.3. Dubins Path Planning. Considering the efficiency and optimality in path planning, several candidate Dubins curves are adopted in this paper which are S, C, CS, SC, and CSC in the order of priority from high to low. Meanwhile, these curves can be divided into two classes which are the coplanar curves and the noncoplanar curves according to the coplanarity of the initial and final configurations. The coplanar curves contain the S, C, CS, SC, and few CSC curves because the initial and final configurations of these curves lie in the same plane. The noncoplanar curves consist of the CSC curves with noncoplanar segments. Therefore, the Dubins curve determination approach is expressed as follows. Firstly the coplanarity of the initial and final configurations is evaluated. If they are coplanar, the coplanar curves are selected as the candidate curves and checked in the order of priority. Otherwise the noncoplanar curve is selected and checked instead. Once the feasible curve is obtained, it is adopted as the feasible path. The algorithm of this approach is presented as Algorithm 1.

To simplify the expression of the Dubins curve determination approach, the calculation process of a certain curve is presented as

$$T(Pt, Cf_1, Cf_2, C_r) = \begin{cases} L, & \text{if the curve exists} \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

where Pt is the desired type of the Dubins curve for determination while Cf_1 and Cf_2 are the initial and final configurations of Pt , respectively, and C_r presents the collection of the feasible radii according to Pt . The function T returns the desired Dubins curve if this curve can be determined; otherwise it returns 0. Besides, the vector \vec{n} is defined to help evaluate the colinearity and coplanarity of the initial and final configurations, where $\vec{n} = \mathbf{g}(\vec{V}_1 \times \vec{V}_2)$.

To prove the validity and efficiency of Algorithm 1, an example is presented and analyzed on the CPU Intel Core I3 6300@3.8 GHz with Matlab. The conditions of this example are set as

$$\begin{aligned} P_1 &= [0, 0, 0], \\ \vec{V}_1 &= [1, 0, 0], \\ P_2 &= [6, 5, 4], \\ \vec{V}_2 &= [-1, -1, -1], \\ R &= 2. \end{aligned} \quad (11)$$

In this example the Dubins curve determination problem is converted into a least squares problem and solved with the Levenberg-Marquardt (L-M) algorithm where $\lambda = 0.005$. The time consumption of Algorithm 1 is 0.683 s with computational accuracy 10^{-5} and the simulation result is illustrated in Figure 2. If the computation resource of the robot is abundant, the evaluation processes of the candidate curves in Algorithm 1 can be implemented at the same time to reduce the time consumption.

Input: $P_1, \vec{V}_1, P_2, \vec{V}_2, R$
Output: feasible Dubins curve L

- (1) **if** $[P_1, \vec{V}_1], [P_2, \vec{V}_2]$ are coplanar, **then**
- (2) **if** $\vec{V}_1 = \vec{V}_2 = \mathbf{g}(\vec{P}_1\vec{P}_2)$, **then**
- (3) $L \leftarrow T(S, [P_1, \vec{V}_1], [P_2, \vec{V}_2])$
- (4) **break**
- (5) **end if**
- (6) **if** $\vec{P}_1\vec{P}_2 = \pm R(\mathbf{g}(\vec{n} \times \vec{V}_1) + \mathbf{g}(\vec{n} \times \vec{V}_2))$, **then**
- (7) $L \leftarrow T(C, [P_1, \vec{V}_1], [P_2, \vec{V}_2], R)$
- (8) **break**
- (9) **end if**
- (10) **if** $\exists P_5, T(C, [P_1, \vec{V}_1], [P_5, \vec{V}_2], R) \neq 0$ and $T(S, [P_5, \vec{V}_2], [P_2, \vec{V}_2]) \neq 0$, **then**
- (11) $L \leftarrow T(CS, [P_1, \vec{V}_1], [P_2, \vec{V}_2], R)$
- (12) **break**
- (13) **end if**
- (14) **if** $\exists P_6, T(S, [P_1, \vec{V}_1], [P_6, \vec{V}_1]) \neq 0$ and $T(C, [P_6, \vec{V}_1], [P_2, \vec{V}_2], R) \neq 0$, **then**
- (15) $L \leftarrow T(SC, [P_1, \vec{V}_1], [P_2, \vec{V}_2], R)$
- (16) **break**
- (17) **end if**
- (18) $L \leftarrow T(CSC, [P_1, \vec{V}_1], [P_2, \vec{V}_2], [R, R])$
- (19) **break**
- (20) **else**
- (21) $L \leftarrow T(CSC, [P_1, \vec{V}_1], [P_2, \vec{V}_2], [R, R])$
- (22) **break**
- (23) **end if**

ALGORITHM 1: Dubins curve determination algorithm.

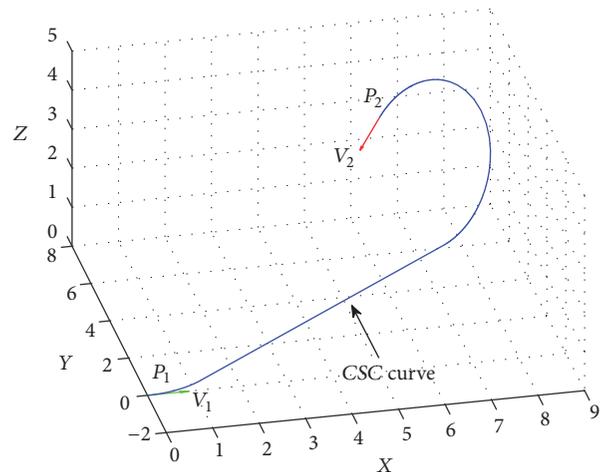


FIGURE 2: The result of Algorithm 1 in Dubins curve determination.

3. Dubins-APF Path Planning

Although the Dubins curves are feasible to form the path for docking with posture constraints, it is difficult to determine the collision-free path and ensure the safety of the robot only using Algorithm 1 when there are unknown obstacles

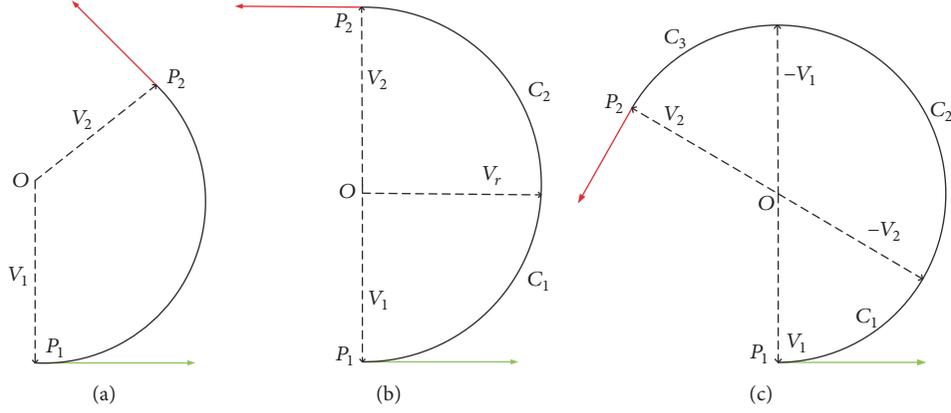


FIGURE 3: Different types of C segments. (a) The minor C segment; (b) the semicircular C segment; (c) the major C segment.

in the workspace. To overcome this shortcoming, a path planning strategy is designed and introduced in this section via combining the Dubins curves with the APF approach to avoid collision with obstacles.

In this paper both static obstacles and moving obstacles are considered. The position and velocity of the obstacle i are written as O_i and \vec{V}_{O_i} separately. Assuming that the minimal safe distance between the robot and obstacle is δ_s , the radius of the obstacle i can be written as $R'_i = R_i + \delta_s$, where R_i is the physical radius of the obstacle.

3.1. Collision Prediction. Collision prediction is one of the fundamental issues of the obstacle avoidance researches [34, 35]. If the planned path is written as L , the motion of the robot following L is presented as $P_r(t)$, where $t \in [0, t_f]$ and t_f is the final time the robot follows L . This issue is easy to address in this paper because the planned path solely consists of the Dubins curves. Therefore once L is obtained, the parameters of these Dubins curves can be utilized to evaluate the feasibility of the planned path in collision prediction. First of all, $\Theta(\vec{V}_i, \vec{V}_j)$ is defined as the operator which calculates and returns the angle between the initial vector \vec{V}_i and the final vector \vec{V}_j .

For following the S segment $\overline{P_1 P_2}$, the movement of the robot on it is simple and can be described as

$$P_r(t) = P_1 + tV_1, \quad (12)$$

where $t \in [0, \|\overline{P_1 P_2}\|/v]$.

There are three types of C segments that one Dubins curve might have which are the minor C segment, the semicircular C segment, and the major C segment. These C segments are illustrated in Figure 3, respectively.

The illustration of the minor C segment is presented in Figure 3(a). For following the minor C segment $\widehat{P_1 P_2}$, $P_r(t)$ is expressed as

$$P_r(t) = O + \frac{\sin(\theta - t(v/R))\vec{V}_1 + \sin(t(v/R))\vec{V}_2}{\sin(\theta)}, \quad (13)$$

TABLE 2: Parameters of c_1 , c_2 , and c_3 .

Index	\vec{V}_i and \vec{V}_j	Central angle θ	Initial and final time
c_1	$\vec{V}_1, -\vec{V}_2$	$\Theta(\vec{V}_1, -\vec{V}_2)$	$0, \theta_1 R/v$
c_2	$-\vec{V}_2, -\vec{V}_1$	$\Theta(-\vec{V}_2, -\vec{V}_1)$	$t_1, t_1 + \theta_2 R/v$
c_3	$-\vec{V}_1, \vec{V}_2$	$\Theta(-\vec{V}_1, \vec{V}_2)$	$t_2, t_2 + \theta_3 R/v$

where $t \in [0, (R/v)\Theta(\overrightarrow{OP_1}, \overrightarrow{OP_2})]$, $\vec{V}_1 = \mathbf{g}(\overrightarrow{OP_1})$, $\vec{V}_2 = \mathbf{g}(\overrightarrow{OP_2})$, and $\theta = \Theta(\vec{V}_1, \vec{V}_2)$.

Similarly, the major C segment is shown in Figure 3(c). $Pr(t)$ can be determined by dividing the major C segment into three minor C segments which are named c_1 , c_2 , and c_3 . If the initial vector and the final vector of the major C segment are expressed as $\vec{V}_1 = \mathbf{g}(\overrightarrow{OP_1})$ and $\vec{V}_2 = \mathbf{g}(\overrightarrow{OP_2})$, the parameters of c_1 , c_2 , and c_3 are concluded in Table 2. Meanwhile, the movement on each minor C segment can be determined according to (13) and $Pr(t)$ of the major C segment can be obtained by their combination.

The semicircular C segment is shown in Figure 3(b). Likewise, it can be divided into two minor C segments which are c_1 and c_2 . The initial and final vectors of c_1 and c_2 are \vec{V}_1 , $\vec{V}_r(0)$ and $\vec{V}_r(0)$, \vec{V}_2 , respectively, where $\vec{V}_r(0)$ is the initial pose of the robot following the semicircular C segment. The movements on c_1 and c_2 can be determined according to (13) and $Pr(t)$ can be obtained by their combination.

Based on $P_r(t)$ of each C segment and S segment of the planned path, the movement of the robot following certain Dubins curve can be specified as well. The feasibility of L can be determined by evaluating these path segments in the temporal order. If there are $\|P_r(t) - O_i(t)\| \geq R'_i$ for all the obstacles, L is collision-free. If L is not collision-free as Figure 4 shows, the time when $\|P_r(t) - O_i(t)\|$ is minimal is defined as the maximal collision time T_{C_i} for O_i . At T_{C_i} the distance between O_i and P_r is defined as the minimal collision distance D_{C_i} for O_i and $D_{C_i} = \|P_r(T_{C_i}) - O_i(T_{C_i})\|$. Therefore if $D_{C_i} < R'_i$, the robot will collide with the obstacle O_i by following L and new L is needed.

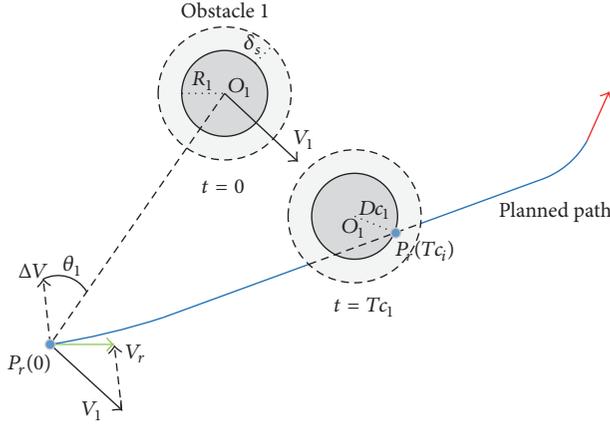


FIGURE 4: The planned path is not collision-free.

3.2. Path Replanning Strategy. Once the planned path is not collision-free, new path is required immediately to avoid collision for the safety of robot. However, sometimes only local knowledge about the environment can be obtained in practice while the replanning process could not be implemented in advance [36–39]. Moreover, if the area for docking is crowded with obstacles, the efficiency of the path replanning process is critical.

To solve this problem, a reactive path replanning strategy is proposed in this paper. The conception of this replanning strategy is to construct the obstacle avoidance potential field to determine the feasible intermediate configuration and the pass-by path is generated to avoid the obstacle based on this intermediate configuration. Then the follow-up path from the intermediate configuration to the docking station is generated to complete the path for docking. The details of this proposed replanning strategy are described as follows.

3.2.1. Obstacle Avoidance Potential Field Design. The obstacle avoidance potential field is proposed in this paper to help determine the intermediate configuration for path replanning inspired by the methods proposed in [30, 31, 40]. The obstacle avoidance potential field is a compound repulsive potential field which consists of two components U_p and U_v . These components are caused by the positions and velocities of the obstacles, respectively. For the obstacle O_i , its corresponding repulsive potential field U_{p_i} is designed as

$$U_{p_i}(\vec{P}_r O_i) = \begin{cases} k_s e^{-\|\vec{P}_r O_i\|}, & \|\vec{P}_r O_i\| < \|\vec{P}_r P_d\| \\ 0, & \text{otherwise,} \end{cases} \quad (14)$$

where k_s is a constant. The corresponding repulsive potential field force \vec{F}_{p_i} is designed as

$$\vec{F}_{p_i}(\vec{P}_r O_i) = -\nabla U_{p_i} = -k_s e^{-\|\vec{P}_r O_i\|} \mathbf{g}(\vec{P}_r O_i), \quad (15)$$

when $\|\vec{P}_r O_i\| < \|\vec{P}_r P_d\|$.

If the obstacle O_i is a moving obstacle, the velocity difference between it and the robot is expressed as $\delta\vec{V} = \vec{V}_r -$

\vec{V}_{O_i} . If \vec{H}_i is defined that $\vec{H}_i = \mathbf{g}(\vec{P}_r O_i \times \delta\vec{V} \times \vec{P}_r O_i)$, the corresponding repulsive potential field U_{v_i} is expressed as

$$U_{v_i}(\vec{H}_i) = \begin{cases} -k_v \|\delta\vec{V}\| e^{-\|\vec{P}_r O_i\|}, & \|\vec{P}_r O_i\| < \|\vec{P}_r P_d\| \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where $\theta_i = \Theta(\vec{V}_r, \vec{P}_r O_i)$ and k_v is a constant. The corresponding repulsive force is expressed as

$$\vec{F}_{v_i} = -\nabla U_{v_i} = k_v \|\delta\vec{V}\| e^{-\|\vec{P}_r O_i\|} \vec{H}_i, \quad (17)$$

when $\|\vec{P}_r O_i\| < \|\vec{P}_r P_d\|$.

The total repulsive field force caused by O_i can be derived from F_{p_i} and F_{v_i} as

$$\vec{F}_r = \sum_{i=1}^N (-\nabla U_{p_i} - \nabla U_{v_i}) = \sum_{i=1}^N (\vec{F}_{p_i} + \vec{F}_{v_i}), \quad (18)$$

where N is the number of the detected obstacles that $\|\vec{P}_r O_i\| < \|\vec{P}_r P_d\|$.

In this paper \vec{F}_r is used to determine the feasible intermediate configuration for obstacle avoidance while considering both the positions and velocities of the obstacles.

3.2.2. Intermediate Configuration Determination. In this paper the feasible intermediate configuration is utilized to generate collision-free path according to \vec{F}_r . The intermediate configuration $[P_m, \vec{V}_m]$ is determined as follows. Firstly, the obstacle O_g is defined as the target obstacle O_g for determining the intermediate configuration, where

$$O_g = \{O_g \mid T_{c_g} = \min\{T_{c_1}, T_{c_2}, \dots, T_{c_i}, \dots, T_{c_N}\}\}. \quad (19)$$

Then the final pose of the path segment that $P_r(T_{c_g})$ is on is chosen as the intermediate pose. Therefore, the original intermediate point P_m is expressed as

$$P_m = O_g(T_{c_g}) + (R'_g + \alpha) \mathbf{g}(\vec{F}_m), \quad (20)$$

where α is the parameter to adjust the position of the intermediate point. The larger α is, the more conservative the path replanning strategy is. \vec{F}_m is defined as the intermediate direction where

$$\vec{F}_m = \begin{cases} \mathbf{g}(\vec{V}_m \times \vec{F}_r \times \vec{V}_m), & \vec{V}_m \times \vec{F}_r \neq 0 \\ \mathbf{g}(\vec{V}_m \times \vec{F}_{\text{rand}} \times \vec{V}_m), & \text{otherwise,} \end{cases} \quad (21)$$

where \vec{F}_{rand} is a nonzero random vector and $\mathbf{g}(\vec{F}_{\text{rand}}) \neq \mathbf{g}(\vec{F}_r)$. At last the Dubins curve L' from $[P_r, \vec{V}_r]$ to $[P_m, \vec{V}_m]$ with radii R_t and R_g is generated as the pass-by path L' .

If L' is collision-free, the intermediate configuration is adopted and L' is adopted as the pass-by path $L_{\text{pass-by}}$. If L' is not collision-free, the new target obstacle can be determined and the new intermediate point P'_m can be obtained as well. The pose of the new intermediate vector is chosen as $\vec{V}'_m = \vec{V}_m$ and the intermediate configuration determination process is repeated.

Input: $[P_r, \vec{V}_r], [P_d, \vec{V}_d], R_t, [O, \vec{V}_O, R'], L, R_1, R_2, k_s, k_v, \alpha$

Output: L

- (1) $\text{rpflag} = 0$
- (2) **while true do**
- (3) $T_{\min} = \min\{T_{C_i} \mid DC_i < R'_i, i \in [0, t_f]\}$
- (4) **if** $T_{\min} > 0$ **then**
- (5) $T_g \leftarrow T_{\min}$, obtain $O_g, R'_g, P_m, \vec{V}_m$
- (6) **if** $\text{rpflag} == 0$ **then**
- (7) $R_1 \leftarrow R_2, R_2 \leftarrow R'_g$
- (8) **end if**
- (9) $L_{\text{pass-by}} \leftarrow T([P_r, \vec{V}_r], [P_m, \vec{V}_m], [R_1, R_2])$
- (10) **if** $L_{\text{pass-by}}$ is not collision-free **then**
- (11) $L \leftarrow L_{\text{pass-by}}, \text{rpflag} = 1$
- (12) **continue**
- (13) **end if**
- (14) $R_1 \leftarrow R_2, R_2 \leftarrow R_t$
- (15) $L_{\text{follow-up}} \leftarrow T([P_m, \vec{V}_m], [P_d, \vec{V}_d], [R_1, R_2])$
- (16) **if** $L_{\text{follow-up}}$ is not collision-free **then**
- (17) Replan $L_{\text{follow-up}}$ with Algorithm 2
- (18) **else**
- (19) $L \leftarrow L_{\text{pass-by}} + L_{\text{follow-up}}$
- (20) **end if**
- (21) **else**
- (22) **break**
- (23) **end if**
- (24) **end while**

ALGORITHM 2: Path replanning algorithm.

3.2.3. Path Generation. The replanned path L consists of two subpaths which are the pass-by path $L_{\text{pass-by}}$ and the follow-up path $L_{\text{follow-up}}$. The initial and final configurations of $L_{\text{pass-by}}$ are $[P_r, \vec{V}_r]$ and $[P_m, \vec{V}_m]$, respectively, while the initial and final configurations of $L_{\text{follow-up}}$ are $[P_m, \vec{V}_m]$ and $[P_d, \vec{V}_d]$. Since $L_{\text{pass-by}}$ is obtained in the intermediate configuration determination process, $L_{\text{follow-up}}$ is generated after $L_{\text{pass-by}}$ is determined. If $L_{\text{follow-up}}$ is collision-free, the combination of $L_{\text{pass-by}}$ and $L_{\text{follow-up}}$ is set as the replanned path for docking and followed till this path is evaluated to be infeasible. Otherwise, $L_{\text{follow-up}}$ is replanned according to the replanning strategy as well.

This proposed path replanning strategy is concluded in Algorithm 2. In Algorithm 2 rpflag is the flag bit which indicates whether a feasible path has been planned ($\text{rpflag} == 1$) or not ($\text{rpflag} == 0$). Once Algorithm 2 is called, the minimal maximal collision time T_{\min} is calculated firstly to evaluate whether a new path should be replanned. If $T_{\min} > 0$, it means the new path is required and the path replanning strategy is implemented. If $T_{\min} \leq 0$, it means the planned path L is collision-free and the replanned path is accepted as the feasible path for docking.

The parameters of Algorithm 2 are selected according the following rules. The obstacle avoidance strategy is affected by k_s/k_v . The larger k_s/k_v is, the more attentions are paid on

avoiding the static obstacles and vice versa. Therefore if the moving obstacles in the environment for docking are large and fast, a small k_s/k_v is preferred to make the path planner sensitive to the moving obstacles. α provides the margin for robot controller in handling the emergency situations such as the velocity change of the moving obstacle and ocean currents. The larger α is, the more potentials are preserved for robot control but, however, the longer the planned path may be. Therefore α is chosen based on the estimation of the environment for docking.

3.3. DAPF Path Planning Approach. The DAPF path planning approach can be concluded as Algorithm 3 based on the Dubins curves and the path replanning strategy. In Algorithm 3 the configuration of the known obstacles is written as \mathcal{C} . The DAPF path planning approach works in a reactive way to the change of the environment as follows. Firstly, an initial path is generated as the target path L and followed by the robot. Then the feasibility of L is evaluated when \mathcal{C} is changed. If L is infeasible, then the new path is generated and L is substituted by the new path. Otherwise \mathcal{C} is updated and L is followed.

An example of the DAPF path planning approach is given in Figure 5, where

$$\begin{aligned}
 P_r &= [0, 0, 0], \\
 \vec{V}_r &= [1, 0, 0], \\
 P_d &= [6, 5, 4], \\
 \vec{V}_d &= [1, -0, 0], \\
 R_t &= 2, \\
 R_s &= 6, \\
 \alpha &= 0.1, \\
 O_1 &= [9, 0, -4], \\
 \vec{V}_{O_1} &= [0, 0, 0.5], \\
 R'_1 &= 3.
 \end{aligned} \tag{22}$$

The result of this scenario shows that the proposed path planning strategy is feasible in generating collision-free path.

4. Simulation and Discussion

In this section simulation experiments are given to prove the validity of the proposed DAPF approach in solving the path planning problem for docking in unknown environment. Meanwhile, the traditional APF approach is adopted in these scenarios to compare the path planning performance with the

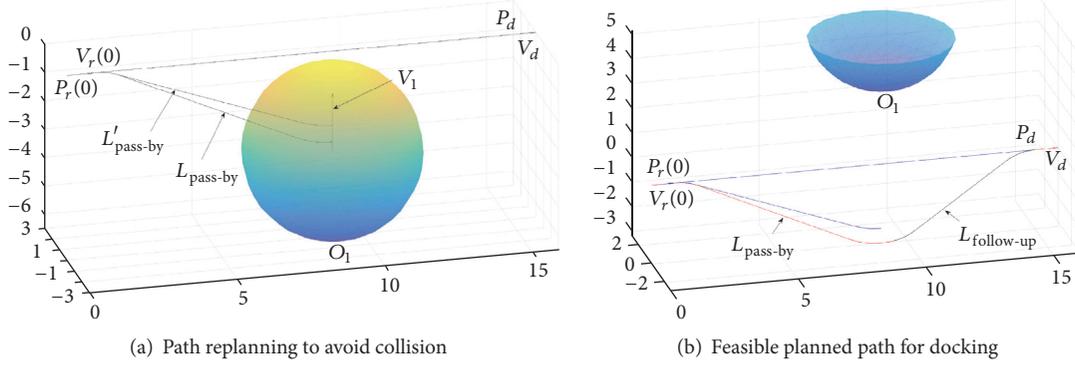


FIGURE 5: Example of the DAPF path planning strategy in environment with one obstacle. (a) The intermediate configuration is determined as well as $L_{\text{pass-by}}$. (b) The planned path is collision-free and feasible for docking.

Input: $[P_r, \vec{V}_r], [P_d, \vec{V}_d], R_r, k_s, k_v, \alpha$
Output: L

- (1) $L \leftarrow T([P_r, \vec{V}_r], [P_d, \vec{V}_d], [R_r, R_t])$
- (2) $R_1 \leftarrow R_r, R_2 \leftarrow R_t$
- (3) $\mathcal{C}' \leftarrow [O, \vec{V}_0, R']$
- (4) **while** $[P_r, \vec{V}_r] \neq [P_d, \vec{V}_d]$ **do**
- (5) **if** $\mathcal{C}' \notin \mathcal{C}$ **then**
- (6) $L \leftarrow \text{Algorithm 2}$
- (7) **end if**
- (8) $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$
- (9) follow L
- (10) **end while**

ALGORITHM 3: DAPF path planning algorithm.

proposed path planning approach. The common parameters of these scenarios are set as

$$\begin{aligned}
 P_r(0) &= [0, 0, 0], \\
 \vec{V}_r(0) &= [1, 0, 0], \\
 P_d &= [15, 0, 0], \\
 \vec{V}_d &= [1, 0, 0], \\
 R_t &= 2, \\
 R_s &= 6,
 \end{aligned} \tag{23}$$

and the step length of these simulations is set as 0.5. The traditional APF approach is designed as

$$\begin{aligned}
 U_{\text{attr}}(\vec{P}_r O_i) &= \frac{k_a}{2} \left(\frac{1}{\|\vec{P}_r O_i\| - R_i} - \frac{1}{\|\vec{P}_r P_d\|} \right)^2, \\
 U_{\text{repu}}(\vec{P}_r O_i) &= k_r \left(\frac{1}{\|\vec{P}_r O_i\| - R_i} \right),
 \end{aligned} \tag{24}$$

where $k_a = 1$ and $k_r = 5$. The parameters of the DAPF approach are $k_s = 1, k_v = 1$, and $\alpha = 0.1$.

4.1. Scenario 1. Both static and moving obstacles are considered in scenario 1. In this scenario, one of the static obstacles is set close to the docking station which simulates a goal nonreachable problem identified in [41, 42]. The configurations of the obstacles are set as

$$\begin{aligned}
 O_1 &= [4, 2, 1], \\
 \vec{V}_1 &= [0, 0, 0], \\
 R'_1 &= 1, \\
 O_2 &= [13, -2, 0], \\
 \vec{V}_2 &= [0, 0, 0], \\
 R'_2 &= 2, \\
 O_3 &= [7, 0, -4], \\
 \vec{V}_3 &= [0, 0, 0.5], \\
 R'_3 &= 3.
 \end{aligned} \tag{25}$$

The path planning results of the traditional APF and DAPF approaches are illustrated in Figure 6 and Table 3. From Figure 6 it can be concluded that the path planned by the APF approach (cyan cross line) is affected by the obstacle near the docking station and the final configuration of its planned path is infeasible for docking. The path planned by the DAPF (black solid line) is collision-free and its final configuration is $V_r = [0.9934, 0.1097, 0.0331]$. It is also proved in this scenario that the DAPF approach is capable of planning the feasible path even if the environment near the docking station is complex.

TABLE 3: Results of scenario 1.

Approach	Final pose	Reach time	$\ \Delta\vec{V}\ $
APF	[0.0051, 0.0564, -0.9984]	21.0000	1.4106
DAPF	[0.9934, 0.1097, 0.0331]	24.5102	0.1148

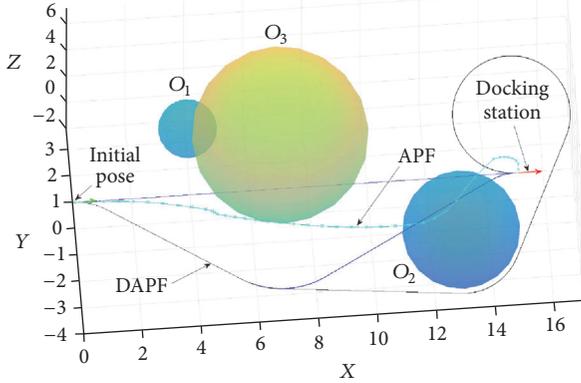


FIGURE 6: Scenario 1: path planning for docking in the environment with static and moving obstacles.

4.2. *Scenario 2.* In scenario 2, the local minima problem is considered and modeled as Figure 7 shows, where the configurations of the obstacles are set as

$$\begin{aligned}
 O_1 &= [7, -2, -2], \\
 O_2 &= [7, -2, 2], \\
 O_3 &= [7, 2, 2], \\
 O_4 &= [7, 2, -2], \\
 O_5 &= [7, -2\sqrt{2}, 0], \\
 O_6 &= [7, 0, 2\sqrt{2}], \\
 O_7 &= [7, 0, -2\sqrt{2}], \\
 O_8 &= [7, 2\sqrt{2}, 0], \\
 O_9 &= [9, 0, 0], \\
 R_1 &= 2, \\
 R_2 &= 2, \\
 R_3 &= 2, \\
 R_4 &= 2, \\
 R_5 &= 3,
 \end{aligned} \tag{26}$$

and the velocities of these obstacles are all set as $[0, 0, 0]$. Therefore, these obstacles form a compound concave obstacle together which will cause the local minima problem in the traditional APF approach.

As Figure 7 and Table 4 show, the path planned by the APF approach (cyan cross line) encounters the local minima

TABLE 4: Results of scenario 2.

Approach	Final pose	Reach time	$\ \Delta\vec{V}\ $
APF	/	∞	/
DAPF	[0.9999, 0.0000, 0.0001]	17.9533	0.0001

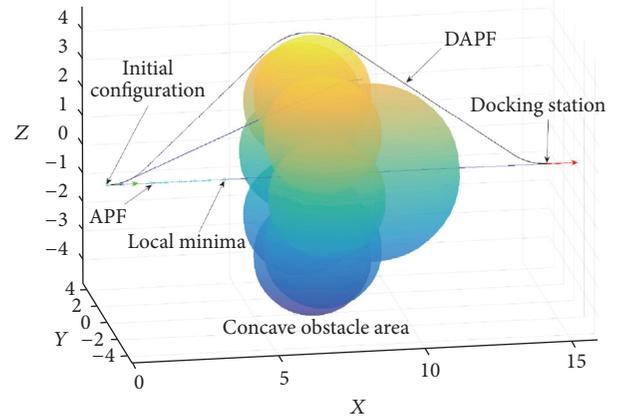


FIGURE 7: Scenario 2: path planning for docking in environment with concave obstacle.

problem and gets stuck at the local minima area while the path planned by the DAPF approach (black solid line) is free from the affection of the local minima and reaches the docking station with feasible pose in spite of the fact that a modified APF is utilized in path replanning.

5. Conclusion and Future Work

In this paper a path planning approach named DAPF is presented for autonomous robot docking based on the combination of the Dubins curves and the artificial potential field approach. Firstly, the determination approach of the Dubins curves is proposed and the collision prediction approach of the planned path is realized based on this approach. Then the path replanning strategy is proposed with the help of the obstacle avoidance potential field. In the replanning strategy the intermediate configuration is calculated based on the obstacle avoidance potential field and adopted to determine the new path. The path planning task is completed through implementing the planning and replanning process in a reactive mode to the changes of environment. Simulation results are also presented to prove the feasibility of the DAPF approach through comparison with the traditional APF approach.

In this proposed approach, little prior knowledge about the environment is required and the planned path is feasible for autonomous robot to follow. As a kind of reactive path planning approaches, this approach is easy to execute by the CPU of the robot in practice. Better performance can be achieved in the future work via adapting parallel computation techniques to enhance the real-time capability of this approach. Specific models of the autonomous robots

will be considered to improve the feasibility of this approach in practical applications such as underwater docking and parking.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant nos. 51579210, 61472325, 61472326, 61733014, and 61633002.

References

- [1] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460–466, 2015.
- [2] Z. Li, S. Zhao, J. Duan, C. Su, C. Yang, and X. Zhao, "Human cooperative wheelchair with brain machine interaction based on shared control strategy," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 185–195, 2017.
- [3] C. Chin and M. Lau, "Modeling and testing of hydrodynamic damping model for a complex-shaped remotely-operated vehicle for control," *Journal of Marine Science and Application*, vol. 11, no. 2, pp. 150–163, 2012.
- [4] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive Neural Network Control of AUVs With Control Input Nonlinearities Using Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, 2017.
- [5] J. Petit and S. E. Shladover, "Potential Cyberattacks on Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, 2015.
- [6] C. S. Chin and S. H. Lum, "Rapid modeling and control systems prototyping of a marine robotic vehicle with model uncertainties using xPC Target system," *Ocean Engineering*, vol. 38, no. 17–18, pp. 2128–2141, 2011.
- [7] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.
- [8] C. W. Bac, E. J. Van Henten, J. Hemming, and Y. Edan, "Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead," *Journal of Field Robotics*, vol. 31, no. 6, pp. 888–911, 2014.
- [9] D. Verstraete, A. Gong, D. D.-C. Lu, and J. L. Palmer, "Experimental investigation of the role of the battery in the AeroStack hybrid, fuel-cell-based propulsion system for small unmanned aircraft systems," *International Journal of Hydrogen Energy*, vol. 40, no. 3, pp. 1598–1606, 2015.
- [10] L. Wu, Y. Li, S. Su, P. Yan, and Y. Qin, "Hydrodynamic analysis of AUV underwater docking with a cone-shaped dock under ocean currents," *Ocean Engineering*, vol. 85, pp. 110–126, 2014.
- [11] N. Mathew, S. L. Smith, and S. L. Waslander, "Multirobot rendezvous planning for recharging in persistent tasks," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 128–142, 2015.
- [12] Z. Yan, C. Deng, D. Chi, T. Chen, and S. Hou, "Path planning method for UUV homing and docking in movement disorders environment," *Scientific World Journal*, vol. 2014, Article ID 246469, 2014.
- [13] K. Teo, E. An, and P.-P. J. Beaujean, "A robust fuzzy autonomous underwater vehicle (AUV) docking approach for unknown current disturbances," *IEEE Journal of Oceanic Engineering*, vol. 37, no. 2, pp. 143–155, 2012.
- [14] D. Herrero, J. Villagr, and H. Martínez, "Self-configuration of waypoints for docking maneuvers of flexible automated guided vehicles," *IEEE Transactions on Automation Science Engineering*, vol. 10, no. 2, pp. 470–475, 2013.
- [15] C. Yang, Z. Li, R. Cui, and B. Xu, "Neural network-based motion control of an underactuated wheeled inverted pendulum model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2004–2016, 2014.
- [16] J. Zhang, S. Zhao, Y. Zhang, and Y. Li, "Optimal planning approaches with multiple impulses for rendezvous based on hybrid genetic algorithm and control method," *Advances in Mechanical Engineering*, vol. 7, no. 3, pp. 1–11, 2015.
- [17] D. Zhu, W. Li, M. Yan, and S. X. Yang, "The path planning of AUV based on D-S information fusion map building and bio-inspired neural network in unknown dynamic environment," *International Journal of Advanced Robotic Systems*, vol. 11, no. 1, pp. 415–429, 2014.
- [18] O. Motlagh, D. Nakhaeinia, S. H. Tang, B. Karasfi, and W. Khaksar, "Automatic navigation of mobile robots in unknown environments," *Neural Computing and Applications*, vol. 24, no. 7–8, pp. 1569–1581, 2014.
- [19] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT*," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2016.
- [20] B. Kovács, G. Szayer, F. Tajti, M. Burdelis, and P. Korondi, "A novel potential field method for path planning of mobile robots by adapting animal motion attributes," *Robotics and Autonomous Systems*, vol. 82, pp. 24–34, 2016.
- [21] Q. Huang and G. Zheng, "Route Optimization for Autonomous Container Truck Based on Rolling Window," *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, Article ID 64116, 2016.
- [22] Á. V. F. M. De Oliveira and M. A. C. Fernandes, "Dynamic planning navigation strategy for mobile terrestrial robots," *Robotica*, vol. 34, no. 3, pp. 568–583, 2016.
- [23] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.
- [24] A. M. Shkel and V. Lumelsky, "Classification of the Dubins set," *Robotics and Autonomous Systems*, vol. 34, no. 4, pp. 179–202, 2001.
- [25] R. G. Sanfelice, S. Z. Yong, and E. Frazzoli, "On minimum-time paths of bounded curvature with position-dependent constraints," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 50, no. 2, pp. 537–546, 2014.
- [26] M. Shanmugavel, A. Tsoordos, B. White, and R. Zbikowski, "Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs," *Control Engineering Practice*, vol. 18, no. 9, pp. 1084–1092, 2010.
- [27] Y. Lin and S. Saripalli, "Path planning using 3D dubins curve for unmanned aerial vehicles," in *Proceedings of the 2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014*, pp. 296–304, Orlando, FL, USA, May 2014.
- [28] O. Cetin, I. Zagli, and G. Yilmaz, "Establishing obstacle and collision free communication relay for UAVs with artificial

- potential fields,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 69, no. 1-4, pp. 361–372, 2013.
- [29] A. Melingui, R. Merzouki, J. B. Mbede, and T. Chettibi, “A novel approach to integrate artificial potential field and fuzzy logic into a common framework for robots autonomous navigation,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 228, no. 10, pp. 787–801, 2014.
- [30] O. Montiel, R. Sep, U. Orozco-Rosas, and R. Sepúlveda, “Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field,” *Journal of Intelligent Robotic Systems*, vol. 79, no. 2, p. 21, 2015.
- [31] H. Min, Y. Lin, S. Wang, F. Wu, and X. Shen, “Path planning of mobile robot by mixing experience with modified artificial potential field method,” *Advances in Mechanical Engineering*, vol. 7, no. 11, Article ID 1687814015619276, 2015.
- [32] M. Shanmugavel, A. Tsourdos, R. Zbikowski, and B. A. White, “3D dubins sets based coordinated path planning for swarm of UAVs,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference 2006*, pp. 1418–1437, August 2006.
- [33] Y. Wang, S. Wang, M. Tan, C. Zhou, and Q. Wei, “Real-time dynamic Dubins-Helix method for 3-D trajectory smoothing,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 730–736, 2015.
- [34] T. M. Howard, “Model-predictive motion planning: Several key developments for autonomous mobile robots,” *IEEE Robotics Automation Magazine*, vol. 21, no. 21, pp. 64–73, 2014.
- [35] Y. Lu, Z. Xi, and J.-M. Lien, “Online collision prediction among 2D polygonal and articulated obstacles,” *International Journal of Robotics Research*, vol. 35, no. 5, pp. 476–500, 2015.
- [36] P. Mobadersany, S. Khanmohammadi, and S. Ghaemi, “A fuzzy multi-stage path-planning method for a robot in a dynamic environment with unknown moving obstacles,” *Robotica*, vol. 33, no. 9, pp. 1869–1885, 2015.
- [37] J. Peng, W. Luo, W. Liu, W. Yu, and J. Wang, “A suboptimal and analytical solution to mobile robot trajectory generation amidst moving obstacles,” *Autonomous Robots*, vol. 39, no. 1, 2015.
- [38] M. Otte and E. Frazzoli, “RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning,” *International Journal of Robotics Research*, vol. 35, no. 7, pp. 797–822, 2016.
- [39] B. H. Lee, J. D. Jeon, and J. H. Oh, “Velocity obstacle based local collision avoidance for a holonomic elliptic robot,” *Autonomous Robots*, pp. 1–17, 2016.
- [40] Y.-b. Chen, G.-c. Luo, Y.-s. Mei, J.-q. Yu, and X.-l. Su, “UAV path planning using artificial potential field method updated by optimal control theory,” *International Journal of Systems Science. Principles and Applications of Systems and Integration*, vol. 47, no. 6, pp. 1407–1420, 2016.
- [41] S. S. Ge and Y. J. Cui, “New potential functions for mobile robot path planning,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.
- [42] U. Orozco-Rosas, O. Montiel, and R. Sepúlveda, “Pseudo-bacterial potential field based path planner for autonomous mobile robot navigation,” *International Journal of Advanced Robotic Systems*, vol. 12, 2015.

Research Article

Consensus Based Platoon Algorithm for Velocity-Measurement-Absent Vehicles with Actuator Saturation

Maode Yan, Ye Tang, Panpan Yang, and Lei Zuo

School of Electronic and Control Engineering, Chang'an University, Xi'an 710064, China

Correspondence should be addressed to Maode Yan; mdyan@chd.edu.cn

Received 1 June 2017; Accepted 19 July 2017; Published 20 August 2017

Academic Editor: Cheng S. Chin

Copyright © 2017 Maode Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We investigate the vehicle platoon problems, where the actuator saturation and absent velocity measurement are taken into consideration. Firstly, a novel algorithm, where a smooth function is introduced to deal with the sharp corner of the input signals, is proposed for a group of vehicles with actuator saturation by using the consensus theory. Secondly, by applying an auxiliary system for the followers to estimate the velocities, a control strategy for the vehicle platoon with actuator saturation and absent velocity measurement is designed via the adaptive control approach. Finally, numerical simulations are provided to illustrate the effectiveness of the proposed approaches.

1. Introduction

Vehicle platoon is well studied in the field of formation control, which is an effective approach to enhance the traffic safety and efficiency [1]. The main objective of vehicle platoon is to avoid traffic jams and improve the traffic security. Usually, a vehicle platoon contains a group of vehicles, which are moving in a lane with certain velocities. From this definition, we can clearly find that the key point of vehicle platoon control is to adjust the velocities and positions of the vehicles, such that they can reach the desired platoon.

In the past few years, a lot of results, like sliding mode control [2], consensus [3], neural network [4], and so on [5–7], have been proposed to deal with the platoon control problems in some ways. For instance, a robust acceleration tracking control with vehicle longitudinal dynamics is presented in [8] for platoon-level automation. In [9], an adaptive bidirectional platoon control method is proposed by using a coupled sliding mode control algorithm. Furthermore, a novel distributed control architecture is given in [10] for heterogeneous platoons, where a group of linear time-invariant autonomous vehicles are considered.

To further study the vehicle platoon control problems, some limitations are taken into account. For example, in [11],

range-limit sensors and delayed actuators are handled by guaranteed-cost platoon control method. Considering the capacity limitation and random packet loss, a closed-form methodology for vehicular platoon control is proposed in [12]. Moreover, a hierarchical platoon control framework is established in [13] to deal with the negative effect of the tracking parameter. The practical string stability of platoon is investigated in [14] for parasitic time delays and lags of the actuators and sensors. In [15], the authors propose control laws in the case of losing communications between the lead vehicle and the other vehicles in a platoon. Besides, disturbances, nonlinear, control input nonlinearities are also taken into consideration in many literatures [16, 17]. However, the fundamental limitations, associated with both actuator saturation and absent velocity measurement, have not been fully studied.

Actuator saturation is one of the most common constraints in practical control systems and it is still a challenge to design the control law with actuator saturation. Considerable attention is paid to the systems with actuator saturation since 1950s [18]. In [19], an online approximation is developed for automation train operation with the presence of actuator saturation. In [20], a position and heading trajectory tracking control law is developed for Swedish wheeled robots with

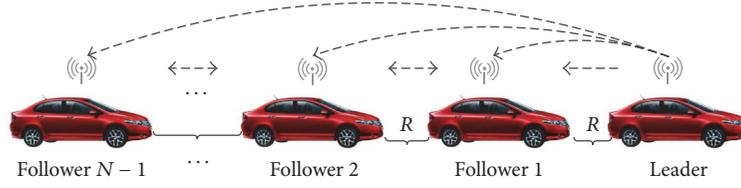


FIGURE 1: Topology structure of vehicle platoon.

unaccounted actuator saturation. In [21], a modified fault tolerant control strategy is presented in the flexible spacecraft to ensure that the control signal will never incur saturation.

For the vehicle platoon with absent velocity measurement, there are also some results. For instance, in [22], passive filters are employed to design an output feedback coordinated formation controller in multiple spacecrafts with absent velocity measurement. A decentralized control strategy is proposed with the neighboring topology for velocity constraints in double-integrator dynamics in [23]. In [24], an adaptive MIMO-ESO is designed to estimate the unmeasurable linear and angular velocities of underwater robots. Moreover, leader-follower output feedback synchronization scheme is given in [25] for controlling the attitude of two satellites when angular velocity measurements are not available. However, to the best of our knowledge, few results have dealt with both issues, absent velocity measurement and actuator saturation, simultaneously in vehicle platoon control.

Motivated by this fact, we propose a novel control strategy to dispose of both the cases: actuator saturation and absent velocity measurement in vehicle platoon. The main contributions of this paper are twofolds:

- (i) We solve the “actuator saturation” problem in vehicle platoon, which occurs when the desired force exceeds the maximum output force of the engines. Transient period or even long period of such behavior will increase the loss of engine and reduce its life. A smooth function $\tanh(\cdot)$ is introduced to handle this problem such that the control inputs can always be below the maximum inputs.
- (ii) We solve the “velocity absent measurement” problem in vehicle platoon control, where intensive oscillation may arise without velocity measurement. We apply an auxiliary system into the control law to estimate the velocity values so that the vehicles can relax the information requirements and communication burden.

The remainder of the paper is organized as follows. Section 2 gives the problem formulation. In Section 3, the controller for vehicle platoon control systems with actuator saturation is designed. Section 4 further proposes an algorithm with both actuator saturation and absent velocity measurement. Numerical simulations are presented in Section 5. Finally, Section 6 draws the conclusion.

2. Problem Formulation

Modeling is an important step in understanding the behavior of the system [26]. In this paper, we consider a group of vehicles, which are composed of one leader and $N - 1$ followers, moving in one-dimensional plane. The topological structure of this vehicle platoon is shown in Figure 1, where R is the desired distance between two consecutive vehicles.

The dynamics of the leader is shown as

$$\dot{r}_L = v_L, \quad (1)$$

where r_L and v_L are the position and velocity of the leader, respectively.

The dynamic models of followers are described as

$$\begin{aligned} \dot{r}_i &= v_i, \\ \dot{v}_i &= \frac{1}{M_i} u_i, \end{aligned} \quad (2)$$

$$i = 1, 2, \dots, N - 1,$$

where $r_i \in \mathbb{R}^m$ and $v_i \in \mathbb{R}^m$ are the position and velocity of vehicle i , respectively. M_i represents the mass of vehicle i and $u_i \in \mathbb{R}^m$ denotes the control input.

In practical applications, the engine will work at maximum output status when there is actuator saturation, which will increase the loss of engine and reduce its life. In addition, due to the loss of velocity information, namely, “absent velocity measurement,” the performance of vehicle platoon control will degrade.

Therefore, the main objective of this paper is to control the velocity-measurement-absent vehicles with actuator saturation such that they can move along a platoon with the desired velocities. Meanwhile, they can keep distance from avoiding collision during the process and the safe distances can converge to a stable constant when they reach the desired states.

3. Vehicle Platoon with Actuator Saturation

Due to the physical limitations of actuators, the control inputs u_i can be written as

$$\begin{aligned} u_i(t) &= \text{sat}(-u_M, u(t), u_M) \\ &= \begin{cases} \text{sign}(u(t)) u_M, & |u(t)| \geq u_M \\ u(t), & |u(t)| < u_M, \end{cases} \end{aligned} \quad (3)$$

where u_M is the upper bound of $u_i(t)$.

As there is a sharp corner between $u_i(t)$ and $u(t)$, it is unrealizable to use this kind of input signals in practice. Hence, we use a smooth function to approximate the control inputs, which is defined as [27]

$$\begin{aligned} g(u(t)) &= u_M \times \tanh\left(\frac{u(t)}{u_M}\right) \\ &= u_M \frac{e^{u(t)/u_M} - e^{-u(t)/u_M}}{e^{u(t)/u_M} + e^{-u(t)/u_M}}. \end{aligned} \quad (4)$$

Then, $\text{sat}(u(t))$ in (3) is expressed as

$$\begin{aligned} \text{sat}(u(t)) &= g(u(t)) + d_1(u(t)) \\ &= u_M \times \tanh\left(\frac{u(t)}{u_M}\right) + d_1(u(t)), \end{aligned} \quad (5)$$

where $d_1(u(t)) = \text{sat}(u(t)) - g(u(t))$ is a bounded function and its bound can be obtained as

$$\begin{aligned} |d_1(u(t))| &= |\text{sat}(u(t)) - g(u(t))| \\ &\leq u_M(1 - \tanh(1)) = D_1. \end{aligned} \quad (6)$$

As $0 \leq |u(t)| \leq u_M$, the bound $d_1(t)$ increases from 0 to D_1 as $|u(t)|$ varies from 0 to u_M . Otherwise, the bound $d_1(u(t))$ decreases from D_1 to 0.

On this basis, an algorithm for the vehicle platoon with actuator saturation is proposed as follows:

$$\begin{aligned} u_i &= \dot{v}_r - \underbrace{\sum_{j=1}^n k_{ij} \tanh(\lambda^k R_{ij})}_{(a)} \\ &\quad - \underbrace{\sum_{j=1}^n \gamma_{ij} \tanh[\lambda^\gamma (v_i - v_j)]}_{(b)}. \end{aligned} \quad (7)$$

Section (a) represents the position control, where $R_{ij} = (r_i - r_j) - (\delta_i - \delta_j)$ and $(\delta_i - \delta_j)$ represents the desired distance between each two vehicles. In addition, $K = [k_{ij}]_{n \times n}$ is the weighted adjacency matrix with $k_{ij} \geq 0$ and λ^k is the strictly positive scalar gain.

Section (b) is the velocity control, where $\gamma = [\gamma_{ij}]_{n \times n}$ is also the weighted adjacency matrix with $\gamma_{ij} \geq 0$ and λ^γ is the strictly positive scalar gain.

Furthermore, the following lemma is introduced for the proposed control system.

Lemma 1. Suppose $\eta \in \mathbb{R}^m$, $\xi \in \mathbb{R}^m$, $\rho \in \mathbb{R}^m$, and $K = [k_{ij}] \in \mathbb{R}^{n \times n}$. If K is symmetric ($k_{ij} = k_{ji}$), then

$$\begin{aligned} &\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n k_{ij} (\eta_i - \eta_j)^T \tanh[\rho(\xi_i - \xi_j)] \\ &= \sum_{i=1}^n \sum_{j=1}^n k_{ij} \eta_i^T \tanh[\rho(\xi_i - \xi_j)]. \end{aligned} \quad (8)$$

Then, the theorem of platoon control with actuator saturation is presented as follows.

Theorem 2. Consider a group of vehicles with actuator saturation in a one-dimensional platoon. By using the control law in (7), the vehicles can reach the platoon with desired distances and velocities. Namely,

- (i) $|u_i| \leq |\dot{v}_r|_{\max} + |\sum_{j=1}^n k_{ij}| + |\sum_{j=1}^n \gamma_{ij}|$;
- (ii) $\lim_{t \rightarrow \infty} (v_i - v_j) = \lim_{t \rightarrow \infty} (v_i - v_L) = 0$;
- (iii) $\lim_{t \rightarrow \infty} (r_i - r_j) = \|i - j\|R$.
- (iv) $\lim_{t \rightarrow \infty} (r_i - r_L) = i \cdot R$.

Proof. Based on absolute value inequality, we can easily find that the control input in (7) is bounded as

$$|u_i| \leq |\dot{v}_r|_{\max} + \left| \sum_{j=1}^n k_{ij} \right| + \left| \sum_{j=1}^n \gamma_{ij} \right|. \quad (9)$$

Let $\tilde{v}_i = v_i - v_r$. Consider the following Lyapunov function candidate:

$$V_1 = \frac{1}{2} \sum_{i=1}^n \tilde{v}_i^T \tilde{v}_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{k_{ij}}{\lambda^k} \mathbf{1}_m^T \log(\cosh(\lambda^k R_{ij})), \quad (10)$$

where $\mathbf{1}_m \in \mathbb{R}^m$ is a vector with all elements equal to one.

Note that V_1 is positive definite and radially unbound with respect to R_{ij} . Taking the derivative of V_1 , we have

$$\begin{aligned} \dot{V}_1 &= \sum_{i=1}^n \tilde{v}_i^T \dot{\tilde{v}}_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n k_{ij} (v_i - v_j)^T \tanh(\lambda^k R_{ij}) \\ &= \sum_{i=1}^n \tilde{v}_i^T \left[-\sum_{j=1}^n k_{ij} \tanh(\lambda^k R_{ij}) \right. \\ &\quad \left. - \sum_{j=1}^n \gamma_{ij} \tanh[\lambda^\gamma (v_i - v_j)] \right] + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n k_{ij} (v_i - v_j)^T \\ &\quad \cdot \tanh(\lambda^k R_{ij}). \end{aligned} \quad (11)$$

According to Lemma 1, we obtain that

$$\dot{V}_1 = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \gamma_{ij} (\tilde{v}_i - \tilde{v}_j)^T \tanh[\lambda^\gamma (\tilde{v}_i - \tilde{v}_j)] \leq 0. \quad (12)$$

This reveals that $V_1 \leq V_1(0)$, and hence v_i , $r_i - r_j$ are globally bounded.

Since $(\dot{v}_i - \dot{v}_j) = (\dot{v}_i - \dot{v}_j)$ is bounded, we can conclude that \ddot{V}_1 is bounded. Based on *Barb\~{a}lat* Lemma, we have $\lim_{t \rightarrow \infty} (\tilde{v}_i - \tilde{v}_j) = 0$ and $\lim_{t \rightarrow \infty} (v_i - v_j) = 0$.

As $\lim_{t \rightarrow \infty} (\dot{v}_i - \dot{v}_r) = \lim_{t \rightarrow \infty} (u_i - \dot{v}_r) = 0$, it leads to

$$\begin{aligned} &\lim_{t \rightarrow \infty} \left[-\sum_{j=1}^n \gamma_{ij} \tanh[\lambda^\gamma (v_i - v_j)] \right. \\ &\quad \left. - \sum_{j=1}^n k_{ij} \tanh(\lambda^k R_{ij}) \right] = 0. \end{aligned} \quad (13)$$

Then, we have

$$\lim_{t \rightarrow \infty} \sum_{j=1}^n k_{ij} \tanh(\lambda^k R_{ij}) = 0. \quad (14)$$

Eq. (14) satisfies

$$\begin{aligned} & \lim_{t \rightarrow \infty} \sum_{i=1}^n \sum_{j=1}^n k_{ij} (r_i - \delta_i)^T \tanh(\lambda^k R_{ij}) \\ &= \frac{1}{2} \lim_{t \rightarrow \infty} \sum_{i=1}^n \sum_{j=1}^n k_{ij} R_{ij}^T \tanh(\lambda^k R_{ij}) = 0. \end{aligned} \quad (15)$$

As a result, $\lim_{t \rightarrow \infty} R_{ij} = 0$. \square

Remark 3. The purpose of using $(\delta_i - \delta_j)$ as a safe distance is to complete the proof by Lyapunov function. With the algorithm (7) proposed in this section, we can obtain that $(r_i - r_j) \rightarrow (\delta_i - \delta_j) \rightarrow \|i - j\|R$.

4. Vehicle Platoon with Actuator Saturation and Absent Velocity Measurement

In this section, we take both the actuator saturation and absent velocity measurement into consideration and propose a novel platoon control law. The main strategy of this proposed method is to design an auxiliary system for the followers, so that they can estimate the velocities effectively.

Invoking the adaptive control theory, the algorithm for the vehicle platoon with both actuator saturation and absent velocity measurement can be presented as

$$\begin{aligned} u_i &= \dot{v}_r - \sum_{j=1}^n k_{ij} \tanh(\lambda^k R_{ij}) - \zeta_i^v \tanh(\lambda^\zeta (r_i - \Psi_i)) \\ &+ \dot{\Theta}_i, \end{aligned} \quad (16)$$

where $K = [k_{ij}]_{n \times n}$ is the weighted adjacency matrix with $k_{ij} \geq 0$. λ^k and λ^ζ are positive scalar gains, and $\zeta_i^v > 0$ is defined as the scalar gain.

The auxiliary system, denoted by Θ_i and Ψ_i , is defined as follows:

$$\dot{\Theta}_i = -\sum_{j=1}^n k_{ij} \tanh(\lambda^k R_{ij}) - k_i^\theta \tanh(\lambda^\theta \Theta_i) \quad (17)$$

$$- \zeta_i^v \tanh(\lambda^\zeta (r_i - \Psi_i)),$$

$$\dot{\Psi}_i = v_r + k_i^\psi (r_i - \Psi_i), \quad (18)$$

where λ^θ , k_i^θ , and k_i^ψ are strictly positive constants, and Θ_i and Ψ_i have arbitrarily initial values.

Theorem 4. Consider a group of vehicles with actuator saturation and absent velocity measurement in a platoon. By using the control law in (16), the vehicles can reach the platoon with desired distances and velocities. Namely,

$$(i) |u_i| \leq |\dot{v}_r|_{\max} + 2 \times [\sum_{j=1}^n (|k_{ij}| + |\zeta_j^v|)] + |k_i^\theta|;$$

$$(ii) \lim_{t \rightarrow \infty} (v_i - v_j) = \lim_{t \rightarrow \infty} (v_i - v_L) = 0;$$

$$(iii) \lim_{t \rightarrow \infty} (r_i - r_j) = \|i - j\|R.$$

$$(iv) \lim_{t \rightarrow \infty} (r_i - r_L) = i \cdot R.$$

Proof. From Theorem 2, we can obtain that the control laws in (16) are bounded as

$$|u_i| \leq |\dot{v}_r|_{\max} + 2 \times \left[\sum_{j=1}^n (|k_{ij}| + |\zeta_j^v|) \right] + |k_i^\theta|. \quad (19)$$

Then denote $\tilde{v}_i = v_i - v_r$. Consider the following Lyapunov function candidate

$$\begin{aligned} V_2 &= \frac{1}{2} \sum_{i=1}^n (\tilde{v}_i - \Theta_i)^T (\tilde{v}_i - \Theta_i) + \frac{1}{2} \sum_{i=1}^n \Theta_i^T \Theta_i \\ &+ \sum_{i=1}^n \frac{\zeta_i^v}{\lambda^\zeta} \mathbf{1}_m^T \log(\cosh(\lambda^\zeta (r_i - \Psi_i))) \\ &+ \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{k_{ij}}{\lambda^k} \mathbf{1}_m^T \log[\cosh(\lambda^k R_{ij})], \end{aligned} \quad (20)$$

where $\mathbf{1}_m \in \mathbb{R}^m$ is a vector with all elements equal to one and the functions $\log(\cdot)$ and $\cosh(\cdot)$ are defined elements-wise for a vector.

Invoking (2) and (16), the time derivative of V_2 is given by

$$\begin{aligned} \dot{V}_2 &= \sum_{i=1}^n (\tilde{v}_i - \Theta_i)^T (\dot{\tilde{v}}_i - \dot{\Theta}_i) + \sum_{i=1}^n \Theta_i^T \dot{\Theta}_i + \sum_{i=1}^n \zeta_i^v (\dot{r}_i \\ &- \dot{\Psi}_i)^T \tanh(\lambda^\zeta (r_i - \Psi_i)) + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n k_{ij} (\dot{R}_{ij})^T \\ &\cdot \tanh(\lambda^k R_{ij}) = \sum_{i=1}^n (\tilde{v}_i - \Theta_i)^T \\ &\cdot \left[-\sum_{j=1}^n k_{ij} \tanh(\lambda^k R_{ij}) - \zeta_i^v \tanh(\lambda^\zeta (r_i - \Psi_i)) \right] \\ &+ \sum_{i=1}^n \Theta_i^T \left[-k_i^\theta \tanh(\lambda^\theta \Theta_i) - \sum_{j=1}^n k_{ij} \tanh(\lambda^k R_{ij}) \right. \\ &- \zeta_i^v \tanh(\lambda^\zeta (r_i - \Psi_i)) \left. \right] - \sum_{i=1}^n \zeta_i^v k_i^\psi (r_i - \Psi_i)^T \\ &\cdot \tanh(\lambda^\zeta (r_i - \Psi_i)) + \sum_{i=1}^n \zeta_i^v \tilde{v}_i \tanh(\lambda^\zeta (r_i - \Psi_i)) \\ &+ \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n k_{ij} (v_i - v_j)^T \tanh(\lambda^k R_{ij}). \end{aligned} \quad (21)$$

From Lemma 1, we know that $\tilde{v}_i = v_i - v_r$ and $v_i - v_j = \tilde{v}_i - \tilde{v}_j$. Then, we obtain that

$$\begin{aligned} \dot{V}_2 = & -\sum_{i=1}^n \zeta_i^\nu k_i^\Psi (r_i - \Psi_i)^\top \tanh(\lambda^\zeta (r_i - \Psi_i)) \\ & - \sum_{i=1}^n k_i^\theta \Theta_i^\top [\tanh(\lambda^\theta \Theta_i)] \leq 0. \end{aligned} \quad (22)$$

This shows that $V_2 \leq V_2(0)$, and v_i , Θ_i , R_{ij} , and $(r_i - \Psi_i)$ are globally bounded.

From the above derivations, we conclude that $\dot{\Psi}$ is bounded. In addition, due to the boundedness of $\tanh(\cdot)$, $\dot{\Theta}_i$ and \dot{V}_2 are also bounded. On the basis of the above analysis, we obtain that $\lim_{t \rightarrow \infty} \Theta_i(t) = 0$ and $\lim_{t \rightarrow \infty} (r_i(t) - \Psi_i(t)) = 0$. Since $\dot{\Psi}_i$, v_i , and Θ_i are bounded, we know that $\ddot{\Theta}_i$ is also bounded. From *Barbalat* Lemma, we can testify that $\lim_{t \rightarrow \infty} \ddot{\Theta}_i(t) = 0$ with the fact that $\lim_{t \rightarrow \infty} \Theta_i(t) = 0$ and $\dot{\Theta}_i$ is bounded.

Then, we have that

$$\lim_{t \rightarrow \infty} \sum_{n=1}^n k_{ij} \tanh(\lambda^k R_{ij}) = 0. \quad (23)$$

With $k_{ij} = k_{ji}$, (23) satisfies

$$\begin{aligned} \lim_{t \rightarrow \infty} \sum_{i=1}^n k_{ij} (r_i - \delta_i)^\top \tanh(\lambda^k R_{ij}) \\ = \frac{1}{2} \lim_{t \rightarrow \infty} \sum_{i=1}^n k_{ij} R_{ij}^\top \tanh(\lambda^k R_{ij}). \end{aligned} \quad (24)$$

Therefore, $\lim_{t \rightarrow \infty} R_{ij} = 0$. According to *Barbalat* Lemma, we can obtain that $\lim_{t \rightarrow \infty} (v_i - v_j) = 0$ because $\tilde{v}_i = \dot{v}_i - \dot{v}_r$ is bounded.

As the derivative of (18), $\dot{\Psi} = \dot{v}_r + k_i^\Psi (\dot{v}_i - \dot{\Psi}_i)$, is bounded and u_i is also bounded, $(\ddot{r}_i - \ddot{\Psi}_i)$ is proven to be bounded. Then, we can conclude that $\lim_{t \rightarrow \infty} (r_i - \Psi_i) = 0$, and $\lim_{t \rightarrow \infty} (v_i - \dot{\Psi}_i) = 0$. Furthermore, we know from (18) that $\lim_{t \rightarrow \infty} (v_i - v_r) = 0$.

Based on the above analysis, we can conclude that $\lim_{t \rightarrow \infty} (v_i - v_r) = 0$ for all vehicles. \square

Remark 5. As the auxiliary system has the same structure with the actual velocity mode, the proposed approach in this paper provides an *indirect* asymptotic estimation of the velocity. By applying this auxiliary system, the platoon control system can generate the necessary damping for the overall closed stability [28, 29].

5. Simulations

To verify the effectiveness of the proposed algorithms in this paper for handling both the ‘‘actuator saturation’’ and ‘‘absent velocity measurement’’ problem, we design a series of simulations, which consist of 6 followers and 1 leader. The initial conditions and safe distance are shown in Table 1.

TABLE 1: Initial conditions and safe distance.

Symbol	Value	Unit
$\delta_i - \delta_j$	5	m
$r_L(0)$	38	m
$r_i(0)$	[0; 10; 14; 20; 26; 31]	m
$v_L(0)$	0	m/s
$v_i(0)$	0	m/s

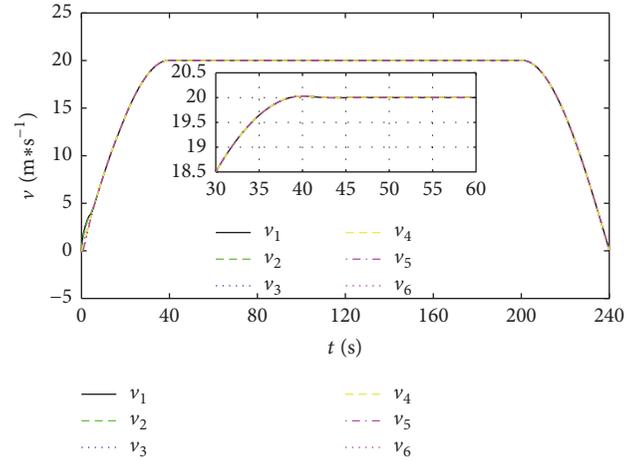


FIGURE 2: Velocities of vehicles with actuator saturation in vehicle platoon.

Then, the reference velocities of the leader are presented as follows, which includes the processes of acceleration, cruise, and deceleration:

$$\begin{aligned} v_{r1} &= 20 \sin\left(\frac{\pi}{80}t\right), \quad t < 40; \\ v_{r2} &= 20, \quad 40 \leq t < 200; \\ v_{r3} &= 20 \sin\left(\frac{\pi}{80}t\right), \quad 200 \leq t < 240. \end{aligned} \quad (25)$$

5.1. Vehicle Platoon with Actuator Saturation. In this part, we only consider the control strategy in (7) and select the controller gains such as $k_{ij} = \gamma_{ij} = \lambda^k = \lambda^\nu = 1$.

In the simulations, Figure 2 shows the velocities of vehicles, from which we can see that all the vehicles follow the desired velocities based on the present control law. In Figure 3, it is observed that the followers track the positions of leader vehicle and the distance between two consecutive vehicles converges to a desired constant value. In addition, we can see that there is no collision during this process.

Note that, in Figure 4, control laws generate traction force for vehicles during the acceleration phase, while they generate braking force when the desired velocities decline. In particularly, the maximum traction force and maximum braking force are limited. Aforementioned simulations confirm the effectiveness of our proposed approach with actuator saturation.

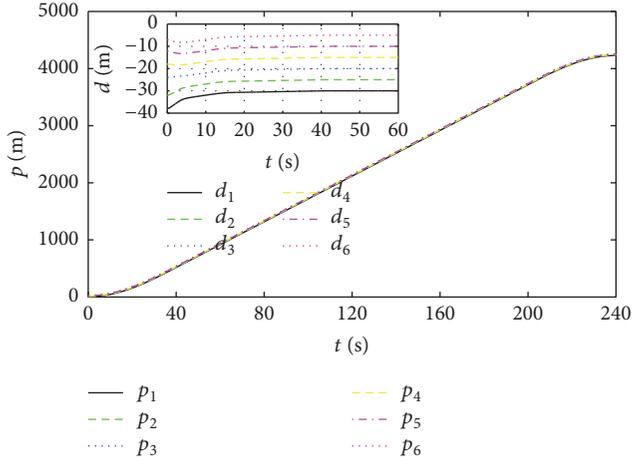


FIGURE 3: Positions of each vehicle and position gaps between vehicles.

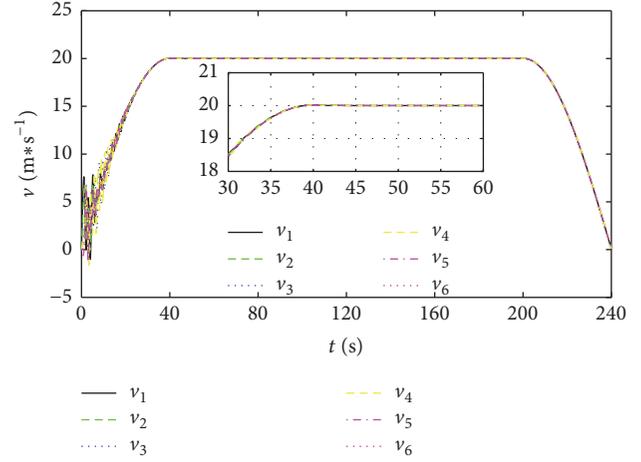


FIGURE 5: Velocities of vehicles with actuator saturation and absent velocity measurement.

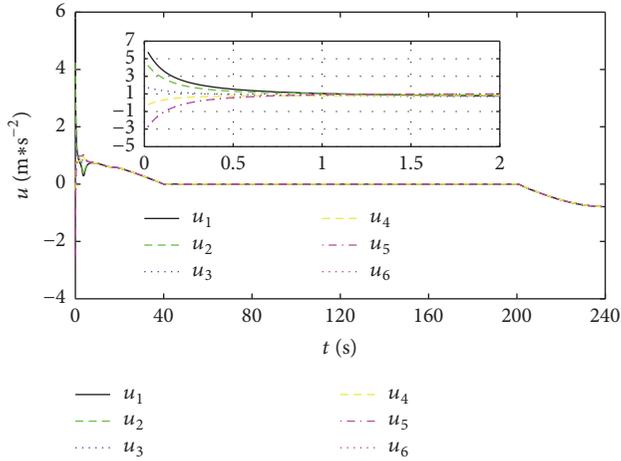


FIGURE 4: Control inputs of vehicles with actuator saturation in vehicle platoon.

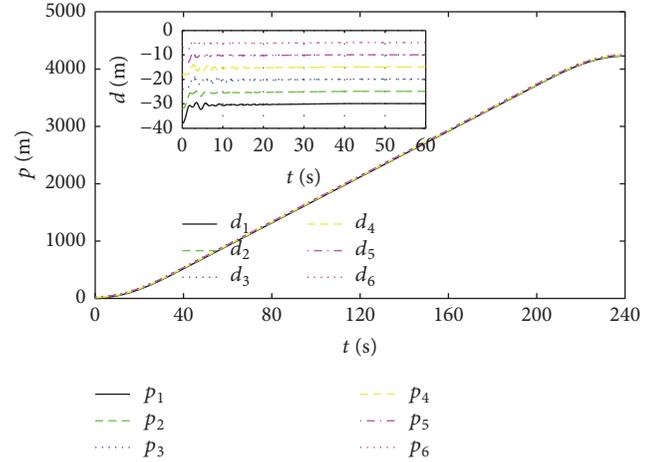


FIGURE 6: Positions of each vehicle and position gaps between vehicles.

5.2. Vehicle Platoon with Actuator Saturation and Absent Velocity Measurement. In this part, we consider control law (16)–(18). We set the initial values of signal Θ_i as $\Theta_i = 0$ and $\dot{\Theta}_i$ as $\dot{\Theta}_i = 0$. Similarly, $\Psi_i = \dot{\Psi}_i = 0$. The controller gains are chosen as $\lambda^k = \lambda^\zeta = \zeta_i^v = 1$. Figures 5–7 are depicted to validate the effectiveness of control algorithm in (16).

Velocities are illustrated in Figure 5, from which we can see that followers can reach the desired velocity, while oscillations present at the beginning in vehicle platoon. This is caused by the damping from auxiliary in order to handle the absent velocities. Figure 6 depicts the position trajectories of vehicles, which has the similar trend with Figure 3. As shown in Figure 6, position gaps converge to the desired value asymptotically, which indicates that platoon maneuver is achieved.

Control inputs in vehicle platoon with actuator saturation and absent velocity measurement are shown in Figure 7, where active control inputs exist in the first period of acceleration and then they can make a quick and smooth change

to accommodate the various desired velocities. Furthermore, we can also see that maximum control input in Figure 7 is $12 \text{ m}\cdot\text{s}^{-2}$, which is within the controllable range in (19) due to the limitations of traction behavior and braking behavior. All the simulations in this part demonstrate control strategy in (16) can achieve similar control performance as control law in (7) despite absent velocity measurement.

5.3. Comparative Analysis. In order to illustrate the advantages of the proposed algorithm, we further design two comparative simulations between the proposed method in this paper and the other related methods.

Firstly, we verify the superiority of the algorithm with the smooth function $\tanh(\cdot)$ in Section 4. A comparison simulation is designed for the proposed algorithm in Section 4 with an existing algorithm, where the upper bounded input is $15 \text{ m}\cdot\text{s}^{-2}$, the lower bounded input is $-15 \text{ m}\cdot\text{s}^{-2}$, and other parameters are the same. Under the control of the algorithm

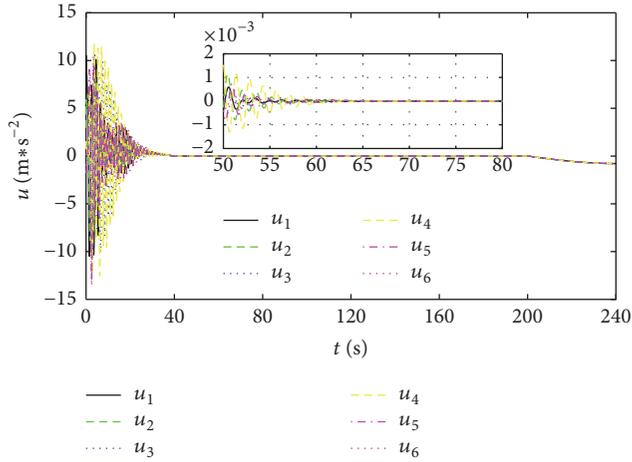


FIGURE 7: Control inputs of vehicles with actuator saturation and absent velocity measurement.

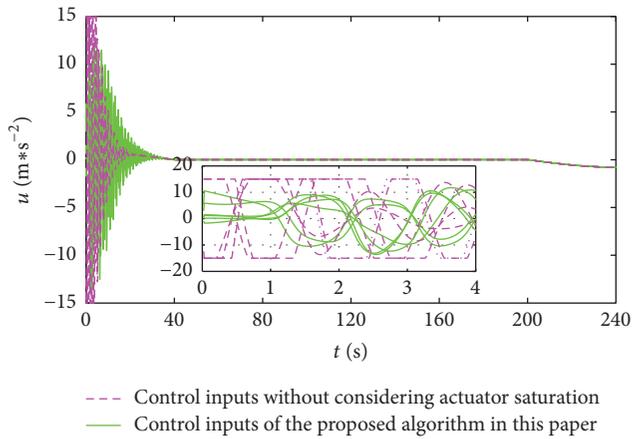


FIGURE 8: Performance comparison with actuator saturation.

in Section 4, the vehicle platoon can maintain the desired states without actuator saturation. The results are as follows.

From Figure 8, we can see that the proposed algorithm can deal with the control inputs saturation, while saturation occurs in other related algorithm.

Secondly, we verify the superiority of the algorithm with the auxiliary system in Section 4 for handling the absent velocity measurement problem. The velocity measurement can be estimated by the auxiliary system on the basis of position information. An existing algorithm without considering the absent velocity measurement is performed as the contrast. The results are as follows.

From Figure 9, we can see that, without velocity measurement, other algorithm will lead to intensive oscillation in the control inputs. However, the proposed algorithm in this paper can effectively deal with this problem and the control inputs can converge to zero within a short time. In addition, the velocities in some related literatures are varying during all the simulations.

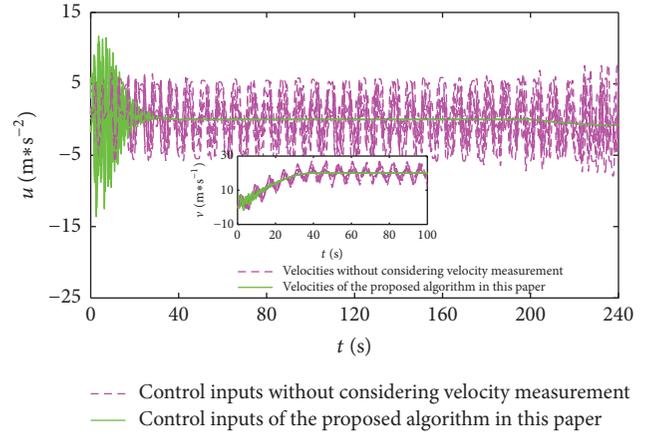


FIGURE 9: Performance comparison with absent velocity measurement.

6. Conclusions

This study proposed a method to investigate vehicle platoon with actuator saturation and absent velocity measurement. Firstly, we applied the consensus theory to the vehicle platoon control strategy with actuator saturation. Based on this achievement, we further proposed vehicle platoon control algorithm for the velocity-measurement-absent vehicles with actuator saturation. The stability of these proposed vehicle platoon systems was proved through the Lyapunov theorem. Finally, numerical simulations were provided to verify the proposed approaches.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is partially supported by the Fundamental Research Funds for the Central Universities of China (310832163403, 310832171004, and 310832101012), the Key Science and Technology Program of Shaanxi Province (2017JQ6060), and the China Postdoctoral Science Foundation Project (2017M613030).

References

- [1] S. Solyom and E. Coelingh, "Performance limitations in vehicle platoon control," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 4, pp. 112–120, 2013.
- [2] C. S. Chin and C. Wheeler, "Sliding-mode control of an electromagnetic actuated conveyance system using contactless sensing," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 5315–5324, 2013.
- [3] C. Santos, F. Espinosa, E. Santiso, M. Martinez, and M. Mazo, "Aperiodic consensus control for tracking nonlinear trajectories of a platoon of vehicles," in *Proceedings of the 18th IEEE*

- International Conference on Intelligent Transportation Systems (ITSC '15)*, pp. 1983–1988, IEEE, September 2015.
- [4] R. Cui, B. Ren, and S. S. Ge, “Synchronised tracking control of multi-agent system with high-order dynamics,” *IET Control Theory & Applications*, vol. 6, no. 5, pp. 603–614, 2012.
 - [5] W. Lin and C. S. Chin, “Block diagonal dominant remotely operated vehicle model simulation using decentralized model predictive control,” *Advances in Mechanical Engineering*, vol. 9, no. 4, pp. 1–24, 2017.
 - [6] Y.-F. Peng, C.-F. Hsu, C.-M. Lin, and T.-T. Lee, “Robust intelligent backstepping longitudinal control of vehicle platoons with H_{∞} tracking performance,” in *Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics*, pp. 4648–4653, IEEE, October 2006.
 - [7] C. S. Chin, M. W. S. Lau, E. Low, and G. G. L. Seet, “Robust and decoupled cascaded control system of underwater robotic vehicle for stabilization and pipeline tracking,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 222, no. 4, pp. 261–278, 2008.
 - [8] S. E. Li, F. Gao, D. Cao, and K. Li, “Multiple-Model Switching Control of Vehicle Longitudinal Dynamics for Platoon-Level Automation,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4480–4492, 2016.
 - [9] J.-W. Kwon and D. Chwa, “Adaptive bidirectional platoon control using a coupled sliding mode control method,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2040–2048, 2014.
 - [10] Ş. Sabı̇u, C. OarĂ, S. Warnick, and A. Jadbabaie, “Optimal distributed control for platooning via sparse coprime factorizations,” in *Proceedings of the 2016 American Control Conference (ACC '16)*, pp. 2591–2598, July 2016.
 - [11] G. Guo and W. Yue, “Autonomous platoon control allowing range-limited sensors,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 7, pp. 2901–2912, 2012.
 - [12] G. Guo and S. Wen, “Communication scheduling and control of a platoon of vehicles in VANETs,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1551–1563, 2016.
 - [13] A. Ghasemi, R. Kazemi, and S. Azadi, “Stable decentralized control of a platoon of vehicles with heterogeneous information feedback,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4299–4308, 2013.
 - [14] L. Xiao and F. Gao, “Practical string stability of platoon of adaptive cruise control vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1184–1194, 2011.
 - [15] S. Sheikholeslam and C. A. Desoer, “Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: a system level study,” *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, pp. 546–554, 1993.
 - [16] J. Si and C. Chin, “An adaptable walking-skid for seabed ROV under strong current disturbance,” *Journal of Marine Science and Application*, vol. 13, no. 3, pp. 305–314, 2014.
 - [17] C. S. Chin, M. W. S. Lau, and E. Low, “Supervisory cascaded controller design: experiment test on a remotely operated vehicle,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 225, no. 3, pp. 584–603, 2011.
 - [18] R. Miller and M. Pachter, “Maneuvering flight control with actuator constraints,” *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 4, pp. 729–734, 1997.
 - [19] S. Gao, H. Dong, Y. Chen, B. Ning, G. Chen, and X. Yang, “Approximation-based robust adaptive automatic train control: an approach for actuator saturation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1733–1742, 2013.
 - [20] G. Indiveri, J. Paulus, and P. Plöger, “Motion control of swedish wheeled mobile robots in the presence of actuator saturation,” *RoboCup 2006: Robot Soccer World Cup X*, pp. 35–46, 2007.
 - [21] B. Xiao, Q. Hu, and Y. Zhang, “Adaptive sliding mode fault tolerant attitude tracking control for flexible spacecraft under actuator saturation,” *IEEE Transactions on Control Systems Technology*, vol. 20, no. 6, pp. 1605–1612, 2012.
 - [22] B. Zhang, S. Song, Z. Zheng, and X. Wei, “Distributed coordinated formation control of multiple spacecraft with and without velocity measurements,” in *Proceedings of the 31st Chinese Control Conference (CCC '12)*, pp. 6105–6111, July 2012.
 - [23] T. A. Jesus, L. C. A. Pimenta, L. A. B. Tórres, and E. M. A. M. Mendes, “Consensus for double-integrator dynamics with velocity constraints,” *International Journal of Control, Automation and Systems*, vol. 12, no. 5, pp. 930–938, 2014.
 - [24] R. Cui, L. Chen, C. Yang, and M. Chen, “Extended state observer-based integral sliding mode control for an underwater robot with unknown disturbances and uncertain nonlinearities,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6785–6795, 2017.
 - [25] A. K. Bondhus, K. Y. Pettersen, and J. T. Gravdahl, “Leader/Follower synchronization of satellite attitude without angular velocity measurements,” in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC '05)*, pp. 7270–7277, December 2005.
 - [26] C. S. Chin, “Systematic modeling and model-based simulation of a remotely operated vehicle using matlab and simulink,” *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 2, no. 4, pp. 481–511, 2011.
 - [27] Y. Li, S. Tong, and T. Li, “Direct adaptive fuzzy backstepping control of uncertain nonlinear systems in the presence of input saturation,” *Neural Computing and Applications*, vol. 23, no. 5, pp. 1207–1216, 2013.
 - [28] A. Abdessameud and A. Tayebi, “Attitude synchronization of a group of spacecraft without velocity measurements,” *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 54, no. 11, pp. 2642–2648, 2009.
 - [29] A. Abdessameud and A. Tayebi, “Velocity-free consensus algorithms for double-integrator dynamics with input saturations constraints,” in *Proceedings of the 2010 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 4486–4491, December 2010.

Research Article

Experimental Autonomous Road Vehicle with Logical Artificial Intelligence

Sergey Sergeevich Shadrin,¹ Oleg Olegovich Varlamov,² and Andrey Mikhailovich Ivanov¹

¹*Automobiles Department, Moscow Automobile and Road Construction State Technical University (MADI), Leningradskiy Pr. 64, Moscow 125319, Russia*

²*Mivar Ltd., Oktyabrskaya Street 72, Moscow 127521, Russia*

Correspondence should be addressed to Sergey Sergeevich Shadrin; shadrin@madi.ru

Received 2 May 2017; Revised 13 June 2017; Accepted 2 July 2017; Published 2 August 2017

Academic Editor: Juan C. Cano

Copyright © 2017 Sergey Sergeevich Shadrin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article describes some technical issues regarding the adaptation of a production car to a platform for the development and testing of autonomous driving technologies. A universal approach to performing the reverse engineering of electric power steering (EPS) for the purpose of external control is also presented. The primary objective of the related study was to solve the problem associated with the precise prediction of the dynamic trajectory of an autonomous vehicle. This was accomplished by deriving a new equation for determining the lateral tire forces and adjusting some of the vehicle parameters under road test conductions. A Mivar expert system was also integrated into the control system of the experimental autonomous vehicle. The expert system was made more flexible and effective for the present application by the introduction of hybrid artificial intelligence with logical reasoning. The innovation offers a solution to the major problem of liability in the event of an autonomous transport vehicle being involved in a collision.

1. Introduction

The desire for the development of more efficient transport services through the use of innovative organizational and technical solutions has fueled the development and implementation of intelligent transport systems (ITS) and intelligent (autonomous) vehicles [1]. A decrease in the effect of the human factor on the process of driving is expected to improve road safety and facilitate the achievement of more efficient transport services. Reduction of the severity of road traffic accidents, preservation of human capacity, and increased labor productivity are some of the advantages of autonomous technologies in everyday life.

The automation of road vehicles has been driven by approximately a century of research in different countries [2]. Several scientific studies [2–8] have been dedicated to address the issues of automatic vehicle control, such as the object component base, the mathematical apparatus and its implementation, and technical vision systems. However, only

a few Advanced Driver Assistance Systems (ADASs) have reached the market, such as serial production vehicles.

The significant stimulus that the field of autonomous road vehicles has received in the last few years has been related to the results of the DARPA Grand Challenge competitions [3]. Developmental projects on autonomous vehicle technologies are currently undertaken by all global motor manufacturers, especially in the US, Germany, Japan, and China. Automobile groups such as Ford, Daimler, Volkswagen, Toyota, Honda, GM, Geely, Tata, and Tesla, as well as other major technological organizations such as Google, Continental, Delphi, Siemens, and Bosch, are involved in these projects as well. Other players in the field include defense departments and agencies of different countries (e.g., the United States' Defense Advanced Research Projects Agency (DARPA)), universities (e.g., Stanford University, Carnegie Mellon University, Technical University of Munich, University of Karlsruhe, Fraunhofer Institute, University of Minnesota, Universidad Politécnica de Madrid), and many other institutions.

In this section, we will consider the main levels of road vehicle control, with the purpose of highlighting some aspects of the present study. In accordance with SAE J3016 approaches [9, 10] the overall act of driving can be divided into three levels of driver efforts: Strategic, Tactical, and Operational levels. The Strategic level involves trip planning; the Tactical level involves maneuvering the vehicle through traffic (lane change, overtaking, speed limit, etc.); and the Operational level involves innate actions such as steering, braking, and other vehicle system control for trajectory tracking. The present study focused on the Tactical and the Operational levels of autonomous vehicle control.

Through observation of the successes of foreign developmental efforts on autonomous technologies and domestic test driving of an experimental car in the autonomous mode, the following main challenges have been determined:

- (1) Operational-level vehicle movement control models are either simplified or parameterized, with the parameters of the mathematical models manually calibrated for each specific vehicle, weight condition, tire type, and so on. In addition, the modeled vehicles move along curved trajectories at only low velocities [3, 11, 12].
- (2) The Tactical level of control is strongly interconnected with the scene recognition of the machine vision systems, which generally utilize deep learning neural networks [13], which in turn, despite their advantages, increase the hardware requirements. It is thus challenging to carry out analyses to justify specific management decisions.

The rest of this paper is organized as follows. In Section 2, we provide a detailed technical description of production car procedures that potentially enable the development of autonomous vehicles. In Section 3, we introduce the half-empirical mathematical model of vehicle dynamics, adaptable to varying external conditions (individual tire traction on different road types, road inclination, etc.), and the properties of vehicles (overall and mass parameters, steering and stability properties, etc.) that enable precise real-time vehicle trajectory prediction. The adjustment of the parameters of the mathematical model using the proposed method constitutes a novel contribution to the literature. In Section 4, we describe our application of Logical Artificial Intelligence, which is based on a Mivar expert system, to Tactical-level control strategies. The results of the application are presented and discussed in Section 5, as well as the conclusions of the study and scope of further work.

2. Open Platform for Development and Testing of Autonomous Road Vehicle Technologies

The experimental autonomous road vehicle used for this study, which is based on Chevrolet Orlando production car (2012MY), was built at the Automobiles department of Moscow Automobile and Road Construction State Technical

University (MADI) as a platform for the development and testing of relevant autonomous vehicle technologies.

The experimental car was equipped with the following hardware:

- (i) A vehicle electronic systems control module. The small, specially designed dual-processor board is hidden above the pedal assembly and used to control the power steering system, inner CAN-bus, and external Ethernet communications, as well as the external servo drives.
- (ii) Brake pedal servo drive.
- (iii) Automatic-transmission shifting mechanism (servo drive).
- (iv) Ethernet network with two switches.
- (v) Two external side video cameras used by the lane keeping assistance system.
- (vi) One back video camera.
- (vii) Three front video cameras (one for lane keeping and two high-speed 2 K cams for stereovision).
- (viii) Forward 3D scanning device. The bionic vision device is a specially designed Light Detection and Ranging (LIDAR) device with one fast changeable beam orientation based on the UDP protocol.
- (ix) Five video processing modules.
- (x) An autonomous driving module. There are two versions of this module. The first is a simple high-performance computer located in the baggage compartment, while the second is a modern Android smartphone with Wi-Fi connection to the vehicle Ethernet network.
- (xi) GLONASS/GPS navigation receiver.
- (xii) Access point to LTE/4G networks. This is used for the remote control function of the vehicle.

The cost of all the additional hardware installed on the experimental vehicle was about \$6000. The open protocol of the communication with the vehicle and all the technical vision devices were specially developed. External control of the electric power steering (EPS) was achieved by a special “electronic gate” installed between the steering column torque sensor and the EPS electronic control unit (ECU). This system has several advantages: (1) When the electronic gate modulates a signal, the EPS ECU “thinks” that the driver has applied a real torque to the steering wheel and begins to turn the wheels. (2) The instances at which the driver wants to take back control of the vehicle can be determined by monitoring the difference between the real torque sensor data and the data modulated by the electronic gate. (3) The EPS functionality is preserved. (4) It affords the cheapest means of taking control of the steering system of the car.

The steering column torque sensor digital signal of the experimental car appears like a voltage ripple on the sensor power supply wire (8 V) within 7.5–8 V. The duration of one data package is about 207 μ s, while the time interval between packages is about 280 μ s. With the assumption of Manchester

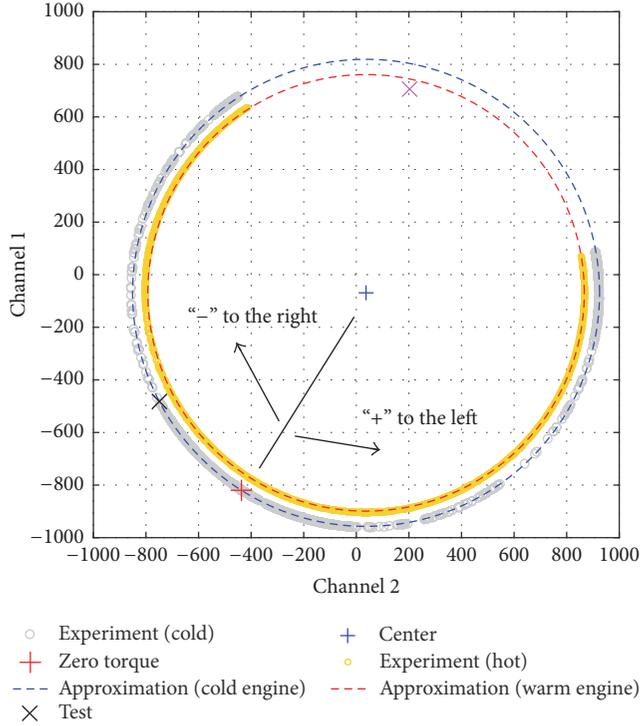


FIGURE 1: Protocol interpretation of the steering column torque sensor.

encoding [14], we chose a time gap of $3.916 \mu\text{s}$ and interpreted a signal fall as a logical zero. We then transformed the data package into a sequence of four 12-bit parts, implying a total of 48 bits on one data line. Following is a sample sequence of three packages:

```
028 1FF 220 3AA
029 1AA 2DF 38E
02A 1EA 231 3E2
```

Further analysis revealed that the first four bits of each segment did not change because there were four segments in the package. The first 12-bit segment was a circular counter. The result of the final transformation of the given sequence of packages was as follows:

```
F20 FAA
ADF A8E
E31 AE2
```

It was thus determined that, in each package received from the steering column torque sensor, there were two important digits, which were the values of the two channels. Our interpretation of the steering column torque sensor protocol is graphically illustrated in Figure 1.

In Figure 1, the red cross indicates the point with coordinates corresponding to the channel values, which are transmitted by the digital protocol from the steering column torque sensor to the EPS ECU. This implies that external torque is not applied to the steering wheel shaft.

When an external torque is applied to the steering wheel, the torque is directly proportional to the angle between the radius (of the approximate circle generated by the torque) that passes through the point of zero moment (red cross) and the

radius that passes through the variable point with coordinates (channel 2; channel 1). This angle is hereafter referred to as the equivalent torque of the steering wheel shaft.

The torque applied by the driver through the steering wheel to the steering wheel shaft is given by

$$M_{SW} = k_{SW} \cdot M_{SWE}, \quad (1)$$

where M_{SW} is the torsion torque on the steering wheel shaft [Nm], k_{SW} is a proportionality factor [Nm/rad], and M_{SWE} is the equivalent torque of the steering wheel shaft [rad].

The following empirical equation of the equivalent torque of the steering wheel shaft was obtained:

$$M_{SWE} = \left(\arccos \left(\frac{2 \cdot 888^2 - (\lambda_2 + 437)^2 - (\lambda_1 + 820)^2}{2 \cdot 888^2} \right) \right) \cdot \text{sgn}(\lambda_2 - \lambda_1 - 368), \quad (2)$$

where λ_1 and λ_2 are, respectively, channel 1 and channel 2 values.

To enable control of the EPS motor, the electronic gate sends packages with the appropriate channel values to the PS ECU. The channel values are calculated as

$$\lambda_1 = -69 + 888 \cdot \sin \left(M_{SWE} - \arccos \left(-\frac{474}{888} \right) \right), \quad (3)$$

$$\lambda_2 = 37 + 888 \cdot \cos \left(M_{SWE} - \arccos \left(-\frac{474}{888} \right) \right).$$

As can be observed from Figure 1, the channel values are dependent on the engine temperature. Our experiments revealed that two data sets were sufficient for an approximate circle of radius 888 for a cold engine (see (2) and (3)) and an approximate circle of radius 830 for a heated engine.

The separate processor (STM32F105RCT6 LQFP64) of the external vehicle electronic systems control module satisfied the requirements of the electronic gate of the EPS system. The main processor (STM32F407VGT6 LQFP100) also satisfied the requirements of all the other tasks. Communication between the processors was achieved through the Serial Peripheral Interface (SPI) at a bit rate of 1 Mbit/s. The vehicle electronic systems control module was equipped with a TJA1042T.118 CAN-bus driver and HLK-RM04 Wi-Fi module with a universal asynchronous receiver/transmitter (UART).

To avoid the need for the installation of additional sensors on the experimental car [1], the messages from the high-speed CAN-bus were decoded to determine the pedal positions, steering wheel angle, wheel velocities, lateral acceleration, yaw rate, and other parameters. The results in DBC format are as follows:

```
BO_201 Pedals_and_RPM: 5 Vector_XXX
SG_Accel_pedal: 39\8@0 + (0.392157,0) [0\255] "%
Vector_XXX
SG_Engine_RPM: 15\16@0 + (0.25,0) [0\255] "RPM"
Vector_XXX
```

BO_485 Steering: 7 Vector_XXX

SG_Steer_wheel_angle_1: 14\15@0-(0.0625,0) [-16383\16383] "grad" Vector_XXX

SG_Steer_wheel_angle_2: 46\15@0-(0.0625,0) [0\32767] "grad" Vector_XXX

BO_840 Wheels_front: 4 Vector_XXX

SG_Front_left_wheel: 7\16@0 + (0.0305,0) [0\255] "km/h" Vector_XXX

SG_Front_right_wheel: 23\16@0 + (0.0305,0) [0\255] "km/h" Vector_XXX

BO_842 Wheels_rear: 4 Vector_XXX

SG_Rear_left_wheel: 7\16@0 + (0.0305,0) [0\255] "km/h" Vector_XXX

SG_Rear_right_wheel: 23\16@0 + (0.0305,0) [0\255] "km/h" Vector_XXX

BO_241 Brakes: 2 Vector_XXX

SG_Brake_pedal: 15\8@0 + (0.392157,0) [0\255] "%" Vector_XXX

BO_249 Speed: 5 Vector_XXX

SG_Vehicle_speed: 31\16@0 + (0.03,0) [0\255] "km/h" Vector_XXX

BO_489 Vehicle_dynamics: 6 Vector_XXX

SG_omega_z: 35\12@0-(0.0625,0) [-2047\2047] "grad/s" Vector_XXX

SG_a_y: 3\12@0-(0.0161,0) [-2047\2047] "m/s2" Vector_XXX

BO_501 Gearbox: 4 Vector_XXX

SG_Gearbox_position: 31\8@0 + (1,0) [0\4] "" Vector_XXX

Braking and automatic-transmission gear shifting were accomplished through electric motors (LA12 GoMotor-World). The electronic gate between the electric accelerator pedal and the engine ECU was used (physically in the main STM407 processor) to control the vehicle acceleration.

We thus achieved a platform for the development and testing of autonomous road vehicles technologies. The experimental vehicle currently has software for the test implementation of the following functions:

- (i) Automatic braking and distance keeping.
- (ii) High-accuracy positioning and route recording.
- (iii) Controlling and holding the vehicle on a road lane.
- (iv) Driving through a predetermined trajectory.
- (v) Real-time dynamic trajectory prediction.
- (vi) Remote control.

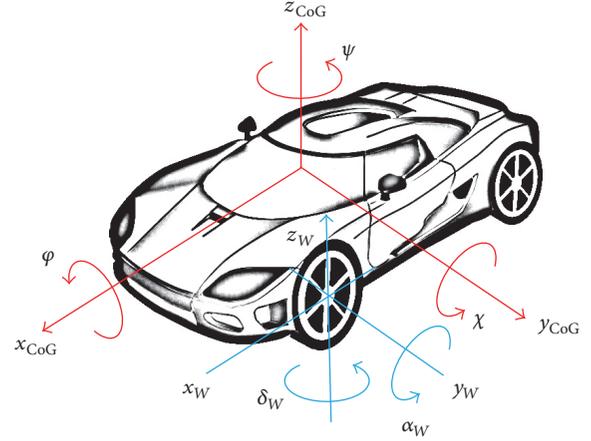


FIGURE 2: Degrees of freedom of a vehicle.

3. High-Precision Vehicle Dynamic Trajectory Prediction

Figure 2 shows the degrees of freedom of a vehicle, the positive directions of the coordinate axes, and the conventional notations. With the exception of the fixed inertial system, all the coordinates move with the vehicle.

A Burckhardt model [15] was used to describe the tire interaction with the road surface. In this approach, the friction coefficient between the tire and the road was calculated using

$$\mu_{\text{Res}}(S_{\text{Res}}) = (c_1 \cdot (1 - \exp(-c_2 \cdot S_{\text{Res}})) - c_3 \cdot S_{\text{Res}}) \cdot \exp(-c_4 \cdot S_{\text{Res}} \cdot V_{\text{CoG}}) \cdot (1 - c_5 \cdot F_Z^2), \quad (4)$$

where the coefficients c_1 , c_2 , and c_3 are determined by the type of road surface (asphalt, concrete, cobblestones, snow, etc.), with the relevant values available in literature [15–17]; c_4 is used to correct the equation for high driving velocities; c_5 is used to consider the dynamic vertical load change; V_{CoG} is the center of gravity (CoG) velocity; S_{Res} is the resultant wheel slip; and F_Z is the vertical load (vertical wheel contact force).

The following system of differential equations was used to analyze the body motion of the vehicle:

$$m_{\text{CoG}} \cdot \begin{bmatrix} \ddot{x}_{\text{In}} \\ \ddot{y}_{\text{In}} \\ \ddot{z}_{\text{In}} \end{bmatrix} = T_{U\text{In}} \cdot \begin{bmatrix} F_{\text{XFL}} + F_{\text{XFR}} + F_{\text{XRL}} + F_{\text{XRR}} + F_{\text{windX}} + F_{\text{GX}} + F_{\text{RR}} \\ F_{\text{YFL}} + F_{\text{YFR}} + F_{\text{YRL}} + F_{\text{YRR}} + F_{\text{windY}} + F_{\text{GY}} \\ F_{\text{ZFL}} + F_{\text{ZFR}} + F_{\text{ZRL}} + F_{\text{ZRR}} + F_{\text{windZ}} + F_{\text{GZ}} \end{bmatrix},$$

$$I_X \cdot \ddot{\varphi} + (I_Z - I_Y) \cdot \dot{\chi} \cdot \dot{\psi} = (F_{\text{ZFL}} - F_{\text{ZFR}}) \cdot 0.5 \cdot b_F$$

$$+ (F_{\text{ZRL}} - F_{\text{ZRR}}) \cdot 0.5 \cdot b_R - m_{\text{CoG}} \cdot \ddot{y}_{\text{CoG}} \cdot h_{\text{CoG}},$$

$$I_Y \cdot \ddot{\chi} + (I_X - I_Z) \cdot \dot{\varphi} \cdot \dot{\psi} = -(F_{\text{ZFL}} + F_{\text{ZFR}}) \cdot l_F$$

$$+ (F_{\text{ZRL}} + F_{\text{ZRR}}) \cdot l_R + m_{\text{CoG}} \cdot \ddot{x}_{\text{CoG}} \cdot h_{\text{CoG}},$$

$$\begin{aligned}
I_Z \cdot \ddot{\psi} + (I_Y - I_X) \cdot \dot{\phi} \cdot \dot{\chi} &= (F_{YFR} + F_{YFL}) \\
&\cdot (I_F - 0.5 \cdot (n_{LFL} + n_{LFR}) \cdot \cos \delta_W) - (F_{YRL} + F_{YRR}) \\
&\cdot (I_R + 0.5 \cdot (n_{LRL} + n_{LRR}) \cdot \cos \delta_W) + (F_{XRR} - F_{XRL}) \\
&\cdot 0.5 \cdot b_R + F_{XFR} \cdot (0.5 \cdot b_F + n_{LFR} \cdot \sin \delta_W) - F_{XFL} \\
&\cdot (0.5 \cdot b_F + n_{LFL} \cdot \sin \delta_W),
\end{aligned} \tag{5}$$

where m_{CoG} is the vehicle mass; \ddot{x}_{In} , \ddot{y}_{In} , and \ddot{z}_{In} are the projections of the CoG acceleration vector in the inertial (fixed) coordinate system; \ddot{x}_{CoG} , \ddot{y}_{CoG} , and \ddot{z}_{CoG} are the projections of the CoG acceleration vector in the CoG (chassis) coordinate system; T_{UIn} is the transformation matrix for rotation of the vector from the undercarriage to the inertial coordinate system; the indices X, Y , and Z determine the orientation of the vectors in the undercarriage coordinate system; F_{windX} , F_{windY} , and F_{windZ} are the projections of the aerodynamic resistance force (wind force); F_{GX} , F_{GY} , and F_{GZ} are the projections of the vehicle gravitational force; I_X , I_Y , and I_Z are the main moments of inertia; and h_{CoG} is the height of the vehicle CoG.

There is the known dynamics problem of precisely determining the lateral wheel force in an unsteady state, due to the pneumatic tire elasticity. The same also applies in a quasisteady state such as during driving with a constant velocity along a constant curvature with a lateral acceleration above 4 m/s^2 [16]. There are many different approaches to solving this problem. The approach employed in this study was a modification of that in [16, 18] using [19, 20]. The following proposed equation for calculating the lateral wheel force was thus obtained:

$$\begin{aligned}
\frac{\partial F_{Sij}}{\partial t} &= \frac{V_W}{L_{y0}} \cdot \left(k_{red,ij} \cdot \left(1 - \frac{F_{Zij}}{k_{y1}} \right) \cdot F_{Zij} \right. \\
&\cdot \left. \arctan(k_{y2} \cdot \alpha(t)) - F_{Sij} \right),
\end{aligned} \tag{6}$$

where V_W is the wheel velocity; α is the tire side slip angle; L_{y0} is the pneumatic tire relaxation length; k_{red} is the reduction factor, which prevents an increase of the lateral force above the adhesion limit during simulation; and k_{y1} and k_{y2} are the dynamic slip resistance coefficients.

The proposed equation (6) describes the decrease in the amplitude of the lateral wheel force under rapid changes in the external impacts (actions), as well as the delay in the growth of the lateral wheel force due to the properties of the tire elasticity.

The complex mathematical model was implemented in MATLAB Simulink. We developed a methodology for searching the unknown model parameters by nonlinear optimization through comparison of the simulation results with those of road tests using real cars. The adjusted parameters of the mathematical model for different cars are presented in Table 1. These parameters obtained by the present methodology constitute a novel contribution to the literature. The model adequacy was examined by comparing the results of road tests with those of simulations and further statistical

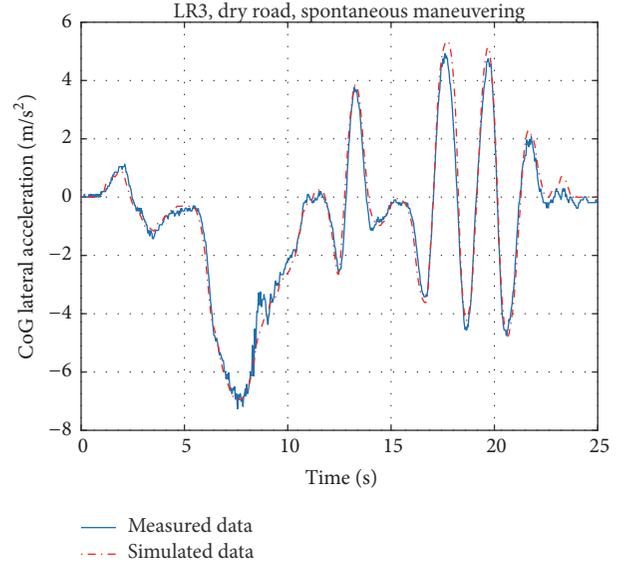


FIGURE 3: Model validation and comparison of lateral acceleration.

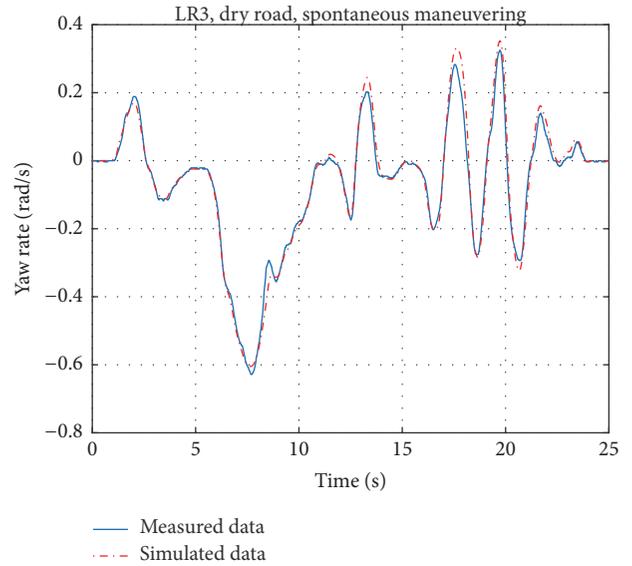


FIGURE 4: Model validation and comparison of yaw rates.

analysis of the error functions. The relative errors of the estimated motion parameters were within 12%.

As an example, Figures 3 and 4 show comparisons of experimental and simulation data. The road test was conducted on a proving ground with spontaneous maneuvering comprising a combination of consecutive actions, namely, rapid acceleration, right turn, slalom, and emergency braking.

The mathematical model of the present experimental autonomous vehicle (Chevrolet Orlando) with the adjusted parameters was used to obtain a huge dataset of possible future trajectories of the vehicle body over the next 3 s of motion with respect to the incoming parameters (current CoG velocity, CoG longitudinal and lateral acceleration, yaw

TABLE 1: Adjusted parameters of mathematical model for different cars.

Number	Parameter	GAZ 322132	LR 3 (discovery 3)	Chevrolet Orlando
(1)	Coefficient of aerodynamic drag, c_{airX} (—)	0.60	0.41	0.32
(2)	Coefficient of overall driving resistance (dry road), f_{01} (—)	0.03	0.04	0.03
(3)	Coefficient for considering increase in wheel rolling resistance with increasing vehicle velocity, k_f (m^2/s^2)	$5.54 \cdot 10^{-6}$	$5.41 \cdot 10^{-6}$	$5.29 \cdot 10^{-6}$
(4)	Dynamic slip resistance coefficient, k_{y1} (N)	15746	18729	13808
(5)	Dynamic slip resistance coefficient, k_{y2} (rad^{-1})	16.39	25.42	20.38
(6)	Tire relaxation length, L_{y0} (m)	1.19	0.18	0.36
(7)	Attenuation factor (degenerates the Kamm circle into an ellipse), k_s (—)	0.80	0.97	0.98

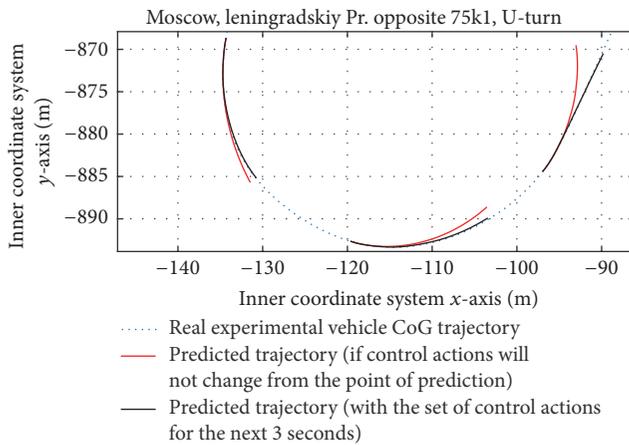


FIGURE 5: Trajectories of the experimental autonomous vehicle.

rate and its derivative, steering wheel angle and its derivative, and type of road under each wheel and corresponding sets of control actions over the next 3 s). The dataset was transformed into a multidimensional array for use in the autonomous vehicle motion planning system (Operational level) during driving along a fixed trajectory. This vehicle trajectory prediction method can be used in real-time systems for collision avoidance.

In Figure 5, the blue dotted line represents the actual trajectory of the experimental vehicle's CoG in an inner coordinate system of the hybrid navigation during the implementation of a U-turn in real traffic in the city of Moscow. For safety reasons, the vehicle was controlled by a human and the calculations were performed after the experiment. The red lines represent the predicted trajectories for the case in which the control actions that are relevant to a certain point of prediction are unchanged over the next 3 s. The black lines represent the predicted trajectories with the input of a multidimensional array of motion planning of the real control actions, which were recorded during the road experiment.

The match of the predicted (black) and real (blue) trajectories in Figure 5 within acceptable limits confirms the

efficiency of the proposed approach. We thus successfully built an autonomous vehicle motion planning system with precise trajectory prediction (Operational level). The development of the system was based on a complex mathematical model of vehicle dynamics, the parameters of which can be adjusted for different types of road vehicles. The model also takes into consideration the changeable environmental conditions such as the road type under each tire, the wind velocity, and the road inclinations, as well as changes in the vehicle itself, such as load and tire changes.

4. Logical Artificial Intelligence in Control Strategy of Autonomous Road Vehicle

It has become clear that, to achieve the fifth level of SAE automation [9], road vehicles should be equipped with hybrid artificial intelligence systems, which would enable the solution of driving tasks under undetermined conditions. According to [21], there have been three main waves in the development of artificial intelligence (AI):

- (1) Wave of "handcrafted knowledge": this was the stage in the development of expert systems in which engineers created the sets of rules that represented the available knowledge in well-defined domains. The developed systems were characterized by poor perception, no learning, and no abstracting abilities, but good reasoning.
- (2) Wave of "statistical learning": this was the era of the development of neural networks such as CNN and deep learning algorithms, when engineers created statistical models for specific problem domains and then trained them using big data. The creations were characterized by good perception and good learning, but poor abstracting and reasoning abilities.
- (3) Wave of "contextual adaptation": this is the current era of AI hybrid systems, which combine the advantages of the above two innovations. The outlook of the current innovations seems to be positive on all counts.

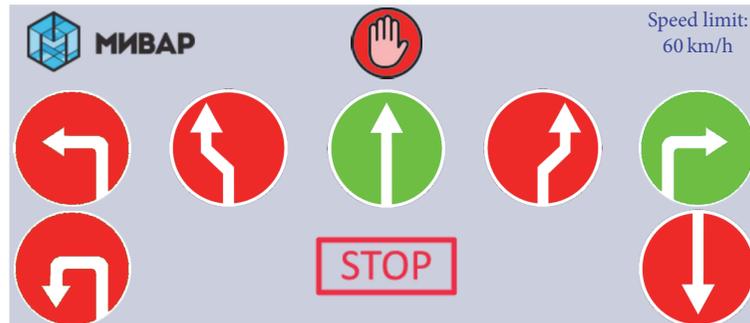


FIGURE 6: Dashboard interface with resulting AI data (10 parameters).

We suppose that, in the driving of a road vehicle in the fully autonomous mode, an expert system with a knowledge database containing formalized traffic rules would initially be sufficient. Attempts to formalize traffic rules in machine language began in 1970. The main challenge is that traffic rules contained in legal texts are mostly written in natural language and are thus often abstract and imprecise. A history of the attempts at formalizing traffic rules and the development of a new approach to the task using Isabelle/HOL are available in [22].

In the present study, we used our own expert system technological platform to formalize the traffic rules. The expert system, which is referred to as the Mivar expert system, consists of multidimensional databases, data processing algorithms, and a cognitive model for describing objects and their relationships. As a software solution, it also consists of tools for designing knowledge models and for model usage (server kernel). The Mivar models are structured as networks of objects (conditions, etc.) and the links (rules) among them. Similar to rational human reasoning, a Mivar kernel initially identifies a means in the uploaded model for solving a request and only thereafter performs a computation. This approach to data processing produces a higher performance compared to the use of classical programming languages and structures like “if-else” (“switch-case”) with billions of rules (combinations). More information about the Mivar expert system can be found in [23, 24].

In the formalization of traffic rules, we divided all the data into two main groups, namely, incoming and resulting parameters. The incoming data was obtained by the scene recognition module and consisted of actual road signs (319 pieces with triggers like “yes”/“no”), traffic lights (up to three objects with 10 different types and four substates for each one), road markings (up to 10 objects with seven different types of road markings for each one), weather conditions (five states), road type (five states), and all the surrounding objects in the relative grid sections around the vehicle, as well as collision prediction flags for the next 3 s of driving (48 sections with seven classes and three states of collision flags). The incoming data formed a structure with 382 cells with a maximum of 1846 independent variables. The exhaustion of all the possible combinations of the variables involved $13e10$ iterations. There were a total of approximately 800 traffic rules, and parallel computing could not be applied to them owing to the requirement for consecutive reasoning.

Indeed, the application of classical programming languages and hardware to such would require approximately $8e2000$ computational iterations per request. However, the Mivar expert system enables the performance of the task about 20 times per second (20 Hz) using common hardware.

The Mivar expert system outputs 10 parameters with permissions for performing the nearest maneuvers. This is graphically portrayed in Figure 6 for a subsequent road situation, where only forward driving and right turn are allowed, the speed limit is 60 km/h, the give way indicator is active, and stopping (slowing down) is required to avoid a forecasted collision. If a collision occurs under this condition, our vehicle would be guilty because it had no right of way based on the traffic rules.

Figure 7 shows an example of a Mivar graph (representation of the objects, rules, and links) for crossing a road intersection in the forward direction according to the traffic rules.

One of the challenges of autonomous road vehicles is the making of fast and correct decisions during operation, with further logging of each reasoned step. Our principal scheme for integrating Logical Artificial Intelligence into an autonomous driving control system is illustrated in Figure 8.

However, an expert system is by itself a very rigid system. In the event of a scenario that is not recognized in the knowledge database, there will be no solution. For example, if the autonomous car stops in front of a broken-down traffic light with the red light on, the car would remain there for eternity. To achieve greater flexibility of the expert system and enhance its applicability to autonomous driving, we added two additional units, namely, “risk management” and “experience analysis.” The added scheme, which was named Logical Artificial Intelligence (LAI), is color-shaded in Figure 8. The knowledge database of LAI consists of three sectors: (1) basic traffic rules; (2) based-on-experience rules; and (3) temporary rules. For example, in the situation of the broken-down traffic light, the uncertainty trigger will lead the program to the “risk management” unit, where all the potential future maneuvers are predicted with respect to the proposed criteria such as the risk of accident in civil driving, time loss in military tasks, and economic factors in business. The optimal maneuver will then be chosen. This will in turn cause the knowledge database sector 3 to change with or without trajectory or route change. After the execution of the maneuver, the “experience analysis” unit will decide on whether to

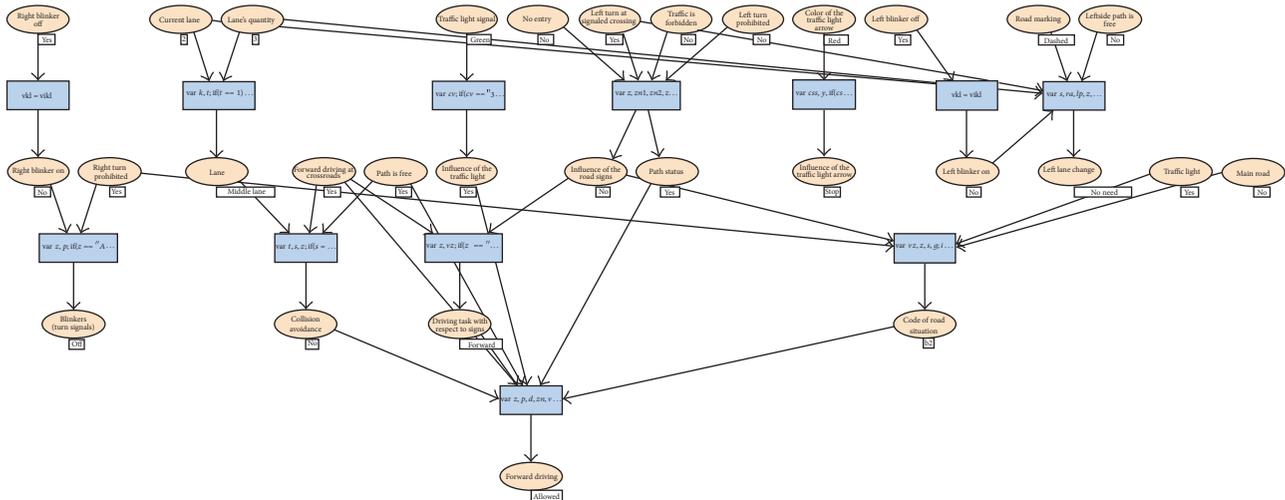


FIGURE 7: Example of a cognitive Mivar graph.

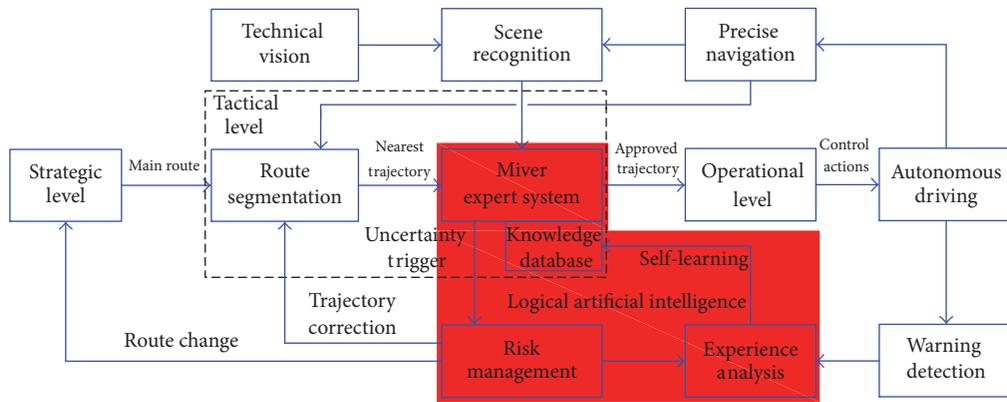


FIGURE 8: Principal scheme for integration of Logical Artificial Intelligence into an autonomous driving control system.

transfer the temporary rule to sector 2 or not. The experience analysis unit may change sector 2 of the knowledge database without the involvement of the “risk management,” and this would amount to a social adaptation. The high performance of the LAI expert system enables driving at high velocities. Overall, the system offers the advantages of logically proven reasoning, high performance, and self-learning.

One significant obstacle to the broad implementation of autonomous driving technologies is the question of liability in the event of a collision. To address this issue, we propose the use of a “black box” as a mandatory device of autonomous vehicles, for use in conjunction with the expert system unit containing the formalized traffic rules (Figure 9). The main challenge of this approach is the standardization of incoming data from the scene recognition unit.

The first road experiments performed using our autonomous vehicle with the integrated LAI system encountered problems with the detection of dynamic scene changes (traffic signs active areas detection/not those from digital map/poor prediction of other traffic participants’ intentions). There were also risk management problems in poorly determined road scenes, as well as some technical issues. Nevertheless,

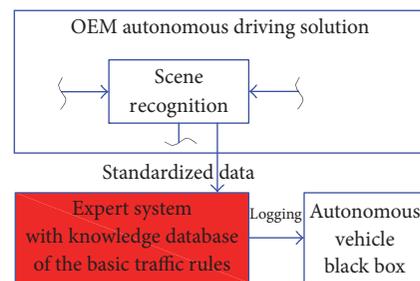


FIGURE 9: Accountability of autonomous vehicles.

overall, the proposed approach was determined to be efficient.

5. Conclusion

5.1. Results. (1) An experimental autonomous road vehicle was built based on Chevrolet Orlando production car as a platform for the development and testing of relevant technologies. A universal approach to performing the reverse

engineering of electric power steering (EPS) for purposes of external control was also developed. The cost of all the additional hardware installed on the experimental vehicle was about \$6000. (2) The vehicle motion planning system is capable of highly precise trajectory prediction, which is achieved by a complex mathematical model of the vehicle dynamics. The model is capable of adjusting its parameters for different types of road vehicles and takes into consideration changes in the environmental conditions such as the road type under each tire, the wind velocity, and the road inclinations, as well as changes in the vehicle itself, such as load and tires changes. The adjustment of the parameters of the mathematical model and the proposed equation for determining the lateral tire forces through trajectory prediction constitute novel contributions to the literature. Multiple road tests have shown that the relative errors of the estimated motion parameters were within 12%. (3) A new Logical Artificial Intelligence (LAI) scheme was developed and integrated into the control system of the autonomous road vehicle. The Mivar expert system enables the performance of the requests to knowledge database of traffic rules with 800 rules and 1846 variables about 20 times per second (20 Hz) using one core of common i7 processor without use of any GPU hardware. The resultant Mivar expert system with LAI was found to be efficient. (4) A method for addressing the issue of accountability in the event of a collision involving an autonomous vehicle was proposed.

5.2. Discussion. (1) The technical processes of the adaptation of a production car to an autonomous vehicle can be applied only to research and testing purposes. In addition, for legal and practical reasons, such a vehicle can only be driven in closed areas and on proving grounds. (2) Sometimes abstract and imprecise expression of traffic rules constitute a challenge to formalizing them, but the application of a global collision avoidance system can be used to overcome the challenge. (3) The standardization of incoming data from the scene recognition unit poses a significant challenge to the use of black boxes in conjunction with expert systems to address the issue of liability when an autonomous vehicle is involved in a collision.

5.3. Conclusion. The results of the present study indicate that any modern vehicle with electric power steering (EPS), automatic transmission (AT), and electronic braking assistance (EBA) can be modified for autonomous driving through appropriate wiring. The real-time dynamic trajectory prediction method adopted in this study can be applied to different autonomous motion planning systems, as well as active collision avoidance safety systems. In addition, the proposed modification of the expert system through the integration of a hybrid artificial reasoning system, namely, the LAI system, enhances the efficiency of autonomous driving control.

5.4. Future Work. We plan to integrate some V2X technologies in the developed autonomous driving platform and conduct further experiments using more experimental vehicles in the form of an autonomous connected fleet. We would also like to address the highlighted challenges of the LAI scheme.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. S. Shadrin and A. M. Ivanov, "Technical aspects of external devices into vehicles' networks integration," *International Journal of Applied Engineering Research*, vol. 11, no. 10, pp. 7003–7006, 2016.
- [2] A. A. Yurchevskiy, *The study of a car as an object of programmable control*, [Ph.D. Dissertation], 1968.
- [3] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, vol. 651, Springer, 2009.
- [4] Ü. Özgüner, T. Acarman, and K. Redmill, *Autonomous Ground Vehicles*, vol. 289, Artech House, 2011.
- [5] K. Nonami, M. Kartidjo, K.-J. Yoon, and A. Budiyo, Eds., *Autonomous Control Systems and Vehicles – Intelligent Unmanned Systems*, vol. 306, Springer, Tokyo, Japan, 2013.
- [6] R. Gonzalez, F. Rodriguez, and J. L. Guzman, *Autonomous Tracked Robots in Planar Off-Road Conditions. Modelling, Localization, and Motion Control*, vol. 122, Springer, 2014.
- [7] A. M. Saikin, S. V. Bakhmutov, A. S. Terenchenko, D. V. Endachev, K. E. Karpukhin, and V. V. Zarubkin, "Tendency of creation of "driverless" vehicles abroad," *Biosciences Biotechnology Research Asia*, vol. 11, pp. 241–246, 2014.
- [8] J. E. Naranjo, M. Clavijo, F. Jimenez, O. Gomez, J. L. Rivera, and M. Anguita, "Autonomous vehicle for surveillance missions in off-road environment," in *Proceedings of the 2016 IEEE Intelligent Vehicles Symposium, IV 2016*, pp. 98–103, Gothenburg, Sweden, June 2016.
- [9] SAE J3016, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, SAE, 2016.
- [10] J. A. Michon, "A critical view of driver behavior models: what do we know, what should we do?" in *Human Behavior and Traffic Safety*, L. Evans and R. C. Schwing, Eds., pp. 485–520, Plenum Press, New York, NY, USA, 1985.
- [11] T. C. Martin, M. E. Orchard, and P. V. Sanchez, "Design and simulation of control strategies for trajectory tracking in an autonomous ground vehicle," in *Proceedings of the 6th IFAC/ACM Conference on Management and Control of Production and Logistics, MCPL 2013*, pp. 118–123, September 2013.
- [12] H. Cheng, *Autonomous Intelligent Vehicles – Theory, Algorithms, and Implementation*, vol. 163, Springer, 2011.
- [13] Mobileye, CES 2016 Presentation, 2016, <http://ir.mobileye.com/investor-relations/events-and-presentations/CES-2016-Presentation/>.
- [14] W. Stallings, *Data and Computer Communications*, Prentice Hall, 7th edition, 2004.
- [15] M. Burckhardt, *Fahrwerktechnik: Radschlupf-Regelsysteme*, Vogel Fachbuch, Würzburg, Germany, 1993.
- [16] U. Kiencke and L. Nielsen, *Automotive Control Systems for Engine, Driveline, and Vehicle*, vol. 512, Springer Verlag, Berlin, Germany, 2nd edition, 2004.
- [17] M. Mitschke and H. Wallentowitz, *Dynamik der Kraftfahrzeuge*, Springer, Berlin, Germany, 4th edition, 2004.
- [18] M. Hiemer, U. Kiencke, T. Matsunaga, and K. Shirasawa, "Cornering stiffness adaptation for improved side slip angle

- observation,” in *Proceedings of the First IFAC Symposium on advances in Automotive Control (AAC04)*, pp. 667–672, Salerno, Italy, 2004.
- [19] M. Guiggiani, *Dinamica del Veicolo*, vol. 384, Citta Studi Edizioni, 2007.
- [20] G. Vandi, D. Moro, F. Ponti et al., “Vehicle dynamics modeling for real-time simulation,” *SAE Technical Papers*, no. 2013-24-0144, article 11, 2013.
- [21] J. Launchbury, *A DARPA Perspective on Artificial Intelligence*, 2017, <http://www.darpa.mil/about-us/darpa-perspective-on-ai>.
- [22] A. Rizaldi and M. Althoff, “Formalising traffic rules for accountability of autonomous vehicles,” in *Proceedings of the 18th IEEE International Conference on Intelligent Transportation Systems, ITSC 2015*, pp. 1658–1665, September 2015.
- [23] O. O. Varlamov, I. A. Danilkin, and I. A. Shoshev, “MIVAR technologies in knowledge representation and reasoning,” in *Proceedings of the 13 International Conference of Pattern Recognition and Information Processing*, pp. 30–32, Minsk, Republic of Belarus, May, 2016.
- [24] O. Varlamov, “MIVAR: Transition from Productions to Bipartite Graphs MIVAR Nets and Practical Realization of Automated Constructor of Algorithms Handling More than Three Million Production Rules,” <https://arxiv.org/abs/1111.1321>.

Research Article

A Variable Interval Rescheduling Strategy for Dynamic Flexible Job Shop Scheduling Problem by Improved Genetic Algorithm

Lei Wang,¹ Chaomin Luo,² and Jingcao Cai¹

¹*School of Mechanical and Automotive Engineering, Anhui Polytechnic University, Wuhu 241000, China*

²*Department of Electrical and Computer Engineering, University of Detroit Mercy, Detroit, MI 48221, USA*

Correspondence should be addressed to Lei Wang; wangdalei2000@126.com

Received 27 April 2017; Accepted 14 June 2017; Published 19 July 2017

Academic Editor: Rongxin Cui

Copyright © 2017 Lei Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In real-world manufacturing systems, production scheduling systems are often implemented under random or dynamic events like machine failure, unexpected processing times, stochastic arrival of the urgent orders, cancellation of the orders, and so on. These dynamic events will lead the initial scheduling scheme to be nonoptimal and/or infeasible. Hence, appropriate dynamic rescheduling approaches are needed to overcome the dynamic events. In this paper, we propose a dynamic rescheduling method based on variable interval rescheduling strategy (VIRS) to deal with the dynamic flexible job shop scheduling problem considering machine failure, urgent job arrival, and job damage as disruptions. On the other hand, an improved genetic algorithm (GA) is proposed for minimizing makespan. In our improved GA, a mix of random initialization population by combining initialization machine and initialization operation with random initialization is designed for generating high-quality initial population. In addition, the elitist strategy (ES) and improved population diversity strategy (IPDS) are used to avoid falling into the local optimal solution. Experimental results for static and several dynamic events in the FJSP show that our method is feasible and effective.

1. Introduction

The flexible job shop scheduling problem (FJSP) is an extension of the classical job shop scheduling problem (JSP) [1]. FJSP can be divided into two subproblems [2]: One is to determine operation sequences, and the other one is to select corresponding machine for these operations. The feasible solution is generated by using these two subproblems in order to make one or some objective functions optimal or near-optimal. FJSP is more complex than JSP because not only the sequencing problem for all operations but also an available machine for each operation needs to be done. Therefore, FJSP is considered as an extremely NP-hard problem [3, 4]. Hence, it cannot be solved by some exact approaches [5] because of the increased computational time in an exponential manner. In recent years, numerous heuristics approaches are proposed to solve FJSP, and these methods include simulated annealing (SA) [6], tabu search (TS) [7], ant colony optimization (ACO) [8], particle swarm optimization (PSO) [9], artificial bee colony (ABC) [10], GA [11], evolutionary algorithm (EA) [12], and improved or hybrid algorithms [13, 14].

As we know, in these heuristics approaches, GA is one of the most popular methods for solving the FJSP because of its potential parallelism and robustness. However, GA has a slow convergence rate and tends to fall into the local optimum; therefore many improved or hybrid genetic algorithms are proposed to deal with the above-mentioned disadvantages. Reference [1] proposed a modified genetic algorithm based on a fuzzy roulette wheel selection, new crossover operator, and new mutation operator for flexible job shop scheduling problems. Zhang et al. [2] designed global and local selection methods in order to generate high-quality initial population for minimizing the makespan. A new chromosome representation and different crossover and mutation strategies are used to minimize the makespan in [11]. A hybrid genetic algorithm with tabu search is proposed to solve FJSP with transportation and bounded processing times constraints in [15]. An improved genetic algorithm for solving the distributed and flexible job shop scheduling problem was introduced in [14]. A novel variable neighborhood genetic algorithm for multiobjective flexible job shop scheduling problems was given in [16].

However, in fact, in real-world floor shop, the production environments are uncertain, and several dynamic or random events such as machine failure, unexpected processing times, random arrival of the urgent orders, and cancellation of the orders may often occur [17]. These dynamic events will lead the initial so-called static scheduling scheme to be infeasible; therefore many dynamic flexible job shop scheduling approaches are proposed to satisfy the new environments [18]. The robust scheduling for multiobjective flexible job shop problem with random machine breakdowns was investigated in [19]. A two-stage hybrid genetic algorithm is proposed to solve FJSP with random machine breakdowns by Al-Hinai and Elmekawy [20]. A genetic algorithm was used to generate a robust and stable scheduling for one machine scheduling with random machine breakdowns in [21]. Zhao et al. proposed a dynamic rescheduling model by using multiagent system in [22]. A heuristic was used to perform the reactive scheduling in a dynamic job shop where jobs arrive over time in [23]. The random job arrivals and machine breakdowns are considered in [24], and a variable neighborhood search is used to solve these two kinds of dynamic events in the FJSP. Considering efficiency and stability, a metaheuristic approach based on GA is developed for dynamic flexible job shop scheduling in [25]. A hybrid genetic algorithm is presented to minimize makespan in dynamic job shop scheduling problem in [26]. Two evolutionary algorithms are used for multiobjective optimization in the FJSP under random machine breakdown in [27]. A hybrid method based on artificial immune systems and priority dispatching rules is proposed by Qiu and Lau [28] for dealing with dynamic online job shop scheduling problem. A hybrid tabu search and genetic algorithm integrated with a simulator to deal with the dynamic job shop scheduling problem with machine breakdowns and random job arrivals in [29]. The gravitational emulation local search algorithm is proposed to solve the multiobjective flexible dynamic job shop scheduling problem in [30]. A multicontextual dispatching rule is proposed by Lu and Romanowski to solve job shops with dynamic job arrival [31].

Although the dynamic or uncertain events in job shop problems or flexible job shop scheduling problems have attracted many researchers in recent years, most of the researches on dynamic scheduling problems focus on certain or primary events such as machine failure or random job's arrival, yet the research on uncertain job's damage is comparatively limited. Meanwhile, as mentioned above, GA has a slow convergence rate and tends to fall into the local optimum; the second obvious weakness of the conventional GA-based approaches is that their performance deteriorates rapidly for complex flexible job shop scheduling problems; the third problem is how to generate high-quality initial population.

Therefore, in this paper, the dynamic FJSP with several uncertain dynamic events is studied. An effective rescheduling strategy is proposed. An improved GA is proposed to minimize the makespan. Our first contribution is that a random initialization population by combining initialization machine and initialization operation with random initialization is designed for generating high-quality initial

TABLE 1: Processing information of FJSP with 2 jobs and 2 machines.

Job	Operation	Processing time		
		M1	M2	M3
J1	O11	2	3	4
	O12	3	4	5
	O13	3	—	—
J2	O21	7	2	3
	O22	4	4	—
	O23	2	—	5

population. Our second contribution is that the elitist strategy (ES) and improved population diversity strategy (IPDS) are used to avoid falling into the local optimal solution. The third contribution is that the variable interval rescheduling strategy is proposed to reduce the impact of dynamic events on scheduling results. Simulation results demonstrate that the proposed method is effective in both scheduling stage and rescheduling stage.

2. Mathematical Formulation with Dynamic Events

In FJSP, there is a set of n jobs $J = \{1, 2, 3, \dots, n\}$ to be processed on m machines indexed by $M = \{1, 2, 3, \dots, m\}$. Each job has one or more operations denoted by o_{ij} , $j \in \{1, 2, \dots, n_i\}$ (where n_i is the total operations of job J_i). Each operation in any job can be processed by one machine out of a predetermined set of available machines. The processing time of operation o_{ij} on machine k is denoted by t_{ijk} . FJSP is going to allocate each operation to a suitable machine and determine the sequence of allocated operations on each machine [32].

The assumptions on the FJSP are as follows:

- (1) At time zero, each machine can be used.
- (2) At time zero, each job can be started.
- (3) One operation only can be processed on a machine at a time.
- (4) Once an operation is processed on a machine, it cannot be interrupted.
- (5) Both due dates and release times are not specified.
- (6) The transportation and setup times of the jobs are ignored.

The objective is to minimize makespan denoted by C , and it can be calculated according to

$$\min C = \max_{1 \leq i \leq n} \{C_i\}, \quad (1)$$

where C_i is the complete time of the i th job.

Table 1 shows an example of FJSP with 3 jobs and 2 machines. Each row and each column refer to an operation and one machine, respectively. In Table 1, the first row indicates that the first operation of $J1$ can be processed by

$M1$ (2 time unites) or $M2$ (3 time unites) or $M3$ (4 time unites). On the other hand, the sixth row shows that the third operation of $J2$ is allowed to be processed on $M1$ or $M3$. Symbol “—” means the machine cannot process corresponding operation.

As mentioned above, the production environments are uncertain, and several dynamic or random events such as machine failure, unexpected processing times, arrival of an urgent order, and cancellation of orders may often occur. When a dynamic event occurs, rescheduling is needed in order to adapt to the dynamic environments.

At present, there are three types of dynamic rescheduling strategies and they are periodic rescheduling, event-driven rescheduling, and hybrid periodic with event-driven rescheduling.

Event-driven rescheduling means that when the scheduling system encounters unexpected events such as machine failure and urgent order insertion while the original scheduling scheme is being executed, the system starts the rescheduling immediately and reoptimizes the remaining unprocessed operations according to the actual production conditions. Event-based scheduling has a good emergent response to unexpected events, but it lacks the global analysis. Periodic rescheduling means that the entire process is divided into several time periods according to the actual processing conditions and machine load, and each period is a cycle. The system automatically performs another rescheduling for the remaining unprocessed operations after completing one period until the end of the entire process. Periodic scheduling can have a global consideration on the system based on the production situation of each processing period, but it cannot effectively deal with some emergencies. The hybrid periodic and event-driven rescheduling strategy can make full use of their advantages; however the results obtained by this hybrid rescheduling method are not necessarily optimal.

A variable interval rescheduling strategy (VIRS) is proposed to deal with the dynamic flexible job shop scheduling problem in this paper. VIRS means that the operations in a certain range are rescheduled after the dynamic event occurs. The range is determined according to operations affected directly by the dynamic event. The operations in other intervals which are not directly affected by the dynamic events keep the initial optimized processing sequence. Meanwhile, the corresponding machines are also unchanged.

The steps for determining the rescheduling interval are as follows.

Step 1. Determine the affected jobs by the dynamic events.

Step 2. Delete the affected jobs from the initial scheduling scheme.

Step 3. Merge the remaining unaffected jobs.

Step 4. Reschedule the affected and unaffected jobs by the dynamic events together.

TABLE 2: Processing information of FJSP with 10 jobs and 5 machines.

Job	Operation	Processing time				
		M1	M2	M3	M4	M5
J1	O_{11}	2	3	2	5	2
	O_{12}	5	3	2	4	3
	O_{13}	5	2	5	2	4
J2	O_{21}	2	2	2	2	4
	O_{22}	5	3	4	4	3
	O_{23}	3	3	3	3	5
J3	O_{31}	2	2	3	2	5
	O_{32}	3	5	5	3	4
	O_{33}	4	2	3	5	4
J4	O_{41}	4	4	4	4	2
	O_{42}	2	4	3	5	2
	O_{43}	2	4	2	3	3
J5	O_{51}	3	4	2	3	5
	O_{52}	5	3	2	2	4
	O_{53}	4	4	4	5	2
J6	O_{61}	3	3	2	3	5
	O_{62}	2	4	2	3	5
	O_{63}	4	2	2	2	4
J7	O_{71}	5	2	4	5	5
	O_{72}	3	5	4	5	4
	O_{73}	3	4	5	2	4
J8	O_{81}	2	2	4	5	3
	O_{82}	4	2	3	5	5
	O_{83}	2	2	4	3	4
J9	O_{91}	4	3	5	4	4
	O_{92}	2	3	2	4	5
	O_{93}	2	2	3	2	5
J10	O_{101}	4	2	3	4	3
	O_{102}	4	4	2	4	5
	O_{103}	2	5	2	3	3

Step 5. Determine the starting time t_1 and the finishing time t_2 of the job directly affected by dynamic events after rescheduling.

Step 6. The rescheduling interval is $[t_1, t_2]$.

Taking the processing information of FJSP in Table 2 as an example, firstly, a static scheduling scheme is obtained and the scheduling Gantt chart is shown in Figure 1. Assume that $M1$ is failure and cannot be repaired at time 5. Therefore jobs 2, 7, 8, and 9 are affected directly by the dynamic event, as shown in Figure 2. The rescheduling Gantt chart obtained by using the dynamic scheduling strategy based on variable rescheduling interval is presented in Figure 3. It can be seen from Figure 3 that the second operation of job 5 is shifted backward, and the third operation of job 3 and the first operation of job 1 are shifted backward so that the processing order of the job which is not directly affected by the dynamic

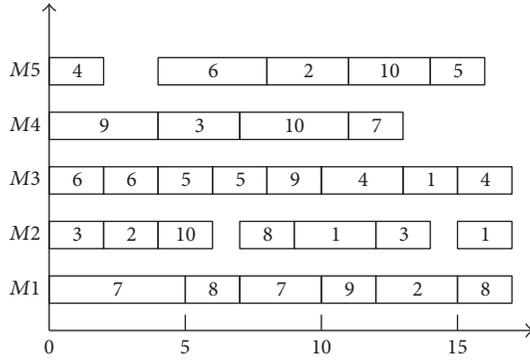


FIGURE 1: Static scheduling Gantt chart.

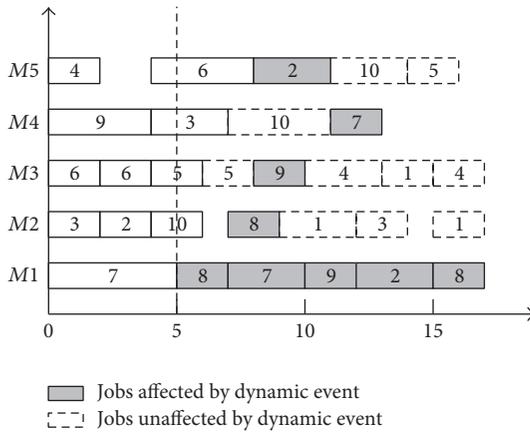


FIGURE 2: M1 failure at time 5.

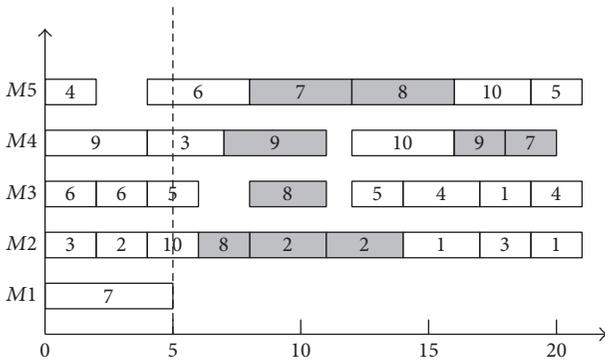


FIGURE 3: Dynamic rescheduling Gantt chart.

event keeps unchangeable. Meanwhile, the corresponding machines are also unchanged.

3. Proposed Genetic Algorithm

3.1. *Encoding.* As mentioned above, there are two subproblems in the FJSP; therefore the feasible encoding of FJSP has two parts. Figure 4 shows a feasible solution for the instance shown in Table 1. The first part of the chromosome is the sequence of operations while the second part is the machine allocation for executing the operations in the first part.

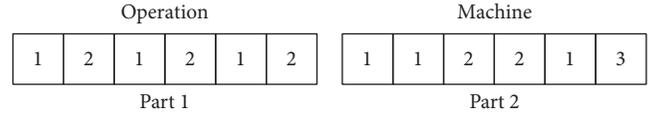


FIGURE 4: Chromosome structure.

The chromosome consists of 12 genes. Numbers 1 and 2 which belong to part 1 represent jobs J_1 and J_2 , respectively. As J_1 and J_2 both have three operations, therefore numbers 1 and 2 appear three times in part 1. On the other hand, numbers 1, 2, and 3 which occur in part 2 represent machines M_1 , M_2 , and M_3 , respectively. For example, the first gene, 1, in part 2 means that the first operation of J_1 is to be processed by machine M_1 , the last gene, 3, in part 2 means that the third operation of J_2 is to be processed by machine M_3 .

3.2. *Initialization.* The quality of the initialization population plays an important role for GA's performance. The random initialization population is used by most researches, while this method probably generates the inferior population. As a result, the population number or evolutionary number needs to be enlarged to obtain the near-optimal solution. Kacem et al. proposed a localization method for generating initial population [34]; however, this method is dependent strongly on the order in which machines and operations are given in the table [35]. A mix of random initialization population by combining initialization machine and initialization operation with random initialization is proposed in this paper.

3.2.1. *Initialization Machine.* Since the objective is the makespan, therefore the machine with less processing time has a priority to process the corresponding operation. The processing order of the jobs is not considered in this stage, and the specific steps are as follows.

Step 1. Read time table T for all processing operations.

Step 2. Find the minimal value T_min_i for each operation.

Step 3. Determine the probability (see (2)) for selecting machine to process corresponding operation according to T_min_i . In (2), apq stands for the total operations of the i th job.

Step 4. Select randomly machine according to the selection probability calculated in Step 3.

Step 5. Repeat Step 4 until the initialization machine is finished.

$$p_i = \frac{1/T_min_i}{\sum_i^{apq} (1/T_min_i)}. \quad (2)$$

3.2.2. *Initialization Operation*

Step 1. Read the current processing information table.

Step 2. Calculate the completion time mak_cur_i after selecting an optional job.

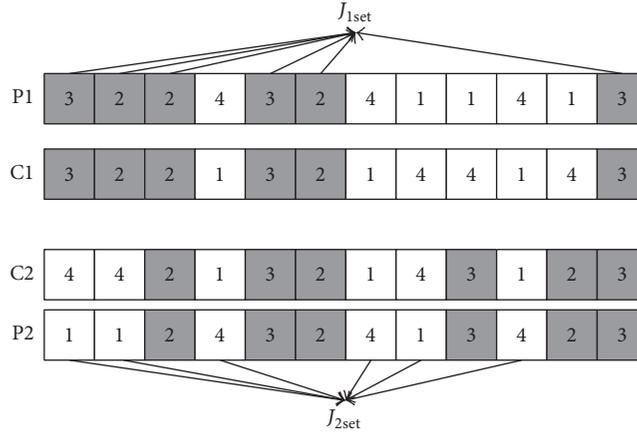


FIGURE 5: Crossover operation for operation part.

Step 3. Determine the probability for selecting each job according to (3).

Step 4. Determine the selected job according to the selection probability calculated in Step 3.

Step 5. Repeat Step 4 until the initialization operation is finished.

$$P_i = \frac{1/mak_cur_i}{\sum_i^n (1/mak_cur_i)} \quad (3)$$

3.2.3. Random Initialization

Step 1. Select randomly an optional machine.

Step 2. Select randomly the job from an optional set of the jobs.

Step 3. Repeat Steps 1 and 2 until the initialization population is finished.

3.3. Selection. The selection based on elitist strategy (ES) and improved population diversity strategy (IPDS) is used in this paper. The elitist strategy can copy some of the best individuals from one parent to the offspring. Compared with the traditional probabilistic selection, the pure elitist strategy can keep the best solution monotonically improving from one generation to the next. However this may lead to a local minimum. Therefore, the IPDS is used to avoid falling into the local optimal solution. The specific steps of IPDS are as follows.

Step 1. Read the population information pop , and set i and j to 1 ($1 \leq i \leq pops$), respectively.

Step 2. Get the i th individual of the pop .

Step 3. Let $j = i + 1$ and then compare the i th individual and the j th individual. If individuals i and j have high similarity, then delete the i th individual.

Step 4. Repeat Step 3 and $j = j + 1$.

Step 5. Repeat Steps 2, 3, and 4 until $i = pops - 1$.

3.4. Crossover Operation. For operation part, the precedence operation crossover (POX) in [36] is used. Assume that chromosomes P1 and P2 are two parents and chromosomes C1 and C2 are two offspring, the detailed crossover steps of the POX are as follows.

Step 1. Randomly divide the job set $\{1, 2, 3, \dots, n\}$ into two nonempty subsets J_{1set} and J_{2set} .

Step 2. Copy the genes belonging to J_{1set} in chromosome P1 to C1. Copy the genes belonging to J_{1set} in chromosomes P2 to C2 and keep these genes in the same position.

Step 3. Copy the genes belonging to J_{2set} in chromosomes P2 to C1. Copy the genes belonging to J_{2set} in chromosomes P1 to C2 and keep these genes in the same order.

An example is shown in Figure 5.

For machine part, a multipoint crossover operation is adopted. The procedure of crossover operation is as follows: A set $rand(0, 1)$, which is composed of 0 and 1, is randomly generated. The length of the chromosome is equal to the machine encoding. Exchange the genes, which are the same as the genes in position 1 in set $rand(0, 1)$, in parents P1 and P2, and then create two offspring C1 and C2.

In addition, for partial flexible job shop scheduling problem (P-FJSP), when the machine number is greater than the total number of the available machines by this crossover operation, then one machine is randomly selected for processing the corresponding operation.

An example is shown in Figure 6.

3.5. Mutation Operation. For operation part, a mutation method by swapping genes is adopted. The detailed mutation procedure of this method is shown as follows.

Step 1. Select randomly two corresponding positions in parent P1.

Step 2. Swap the genes to generate an offspring C1.

An example is shown in Figure 7.

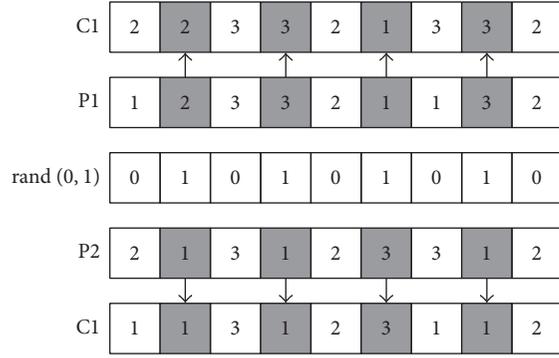


FIGURE 6: Crossover operation for machine part.

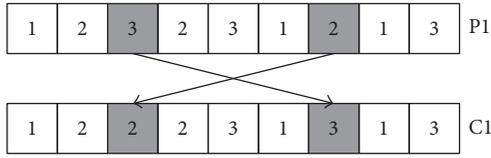


FIGURE 7: Mutation operation for operation part.

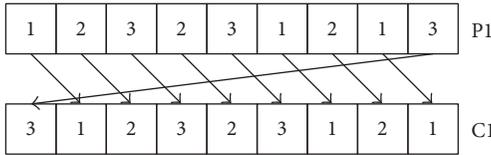


FIGURE 8: Mutation operation for machine part.

For machine part, a mutation method by moving genes backward in the chromosome P1 is used to generate C1.

If the machine number obtained by this method is greater than the total number of the available machines by this crossover operation, then a machine is selected in the machine set to process the corresponding operation in a random manner.

An example is shown in Figure 8.

The flow chart of the proposed improved genetic algorithm and dynamic rescheduling strategy is shown in Figure 9.

4. Experiments and Results

4.1. Experiments of the Initialization Population Method. In order to verify the effectiveness of our proposed initialization population method, we compare our improved genetic algorithm with the basic genetic algorithm on the fitness of the best individual, the average fitness of the initial population, and the evolutionary generation. Table 3 shows the processing information for 5-job and 5-machine FJSP.

Parameters are as follows: population size is 100, evolutionary generation is 100, the crossover probability is 0.7, and the mutation probability is 0.1.

Table 4 shows the comparison results by using our proposed improved initializing population method and random

TABLE 3: Processing information for 5-job and 5-machine FJSP.

Job	Operation	Processing time				
		M1	M2	M3	M4	M5
J1	O11	5	4	7	4	10
	O12	4	4	10	3	9
	O13	5	8	3	6	10
J2	O21	8	9	9	9	4
	O22	3	2	2	1	7
	O23	8	2	4	7	8
J3	O31	7	2	1	9	9
	O32	8	7	7	8	2
	O33	4	6	1	7	1
J4	O41	1	8	7	9	8
	O42	7	8	3	4	10
	O43	8	3	1	6	6
J5	O51	3	7	7	10	6
	O52	2	10	9	5	10
	O53	2	2	5	9	5

TABLE 4: Comparison of the initializing population methods.

Method	The fitness of the best individual	The average fitness of the initial population
Improved initializing population	18.92	33.71
Random initializing population	26.02	42.15

initializing population method by running randomly 100 times. It can be seen from Table 4 that our proposed initializing population method can obtain high-quality individuals.

Figure 10 shows the evolutionary curve obtained by the traditional random initializing population method. Figure 11 shows the evolutionary curve obtained by our proposed improved initializing population method. It can be seen that the random method takes more than 60 generations to obtain the optimal solution, while the proposed improved initializing population method only needs less than 30 generations to obtain the optimal solution. What is more, compared with the basic GA, the optimal solution obtained by using our improved GA is also better than that. Therefore, the evolutionary speed and solution quality can be improved obviously by using our proposed improved initializing population method. Figure 12 shows the Gantt chart.

4.2. Experiments of the Dynamic Rescheduling Strategy. In order to verify the effectiveness of our proposed improved genetic algorithm and dynamic rescheduling strategy based on variable interval scheduling, the comparative experiments were done. The case about FJSP from literature [33] is shown in Table 5. The static scheduling Gantt chart obtained by our proposed improved genetic algorithm is shown in Figure 13, and the optimal makespan is 35 which is smaller than the optimal value obtained by Wu et al. [33].

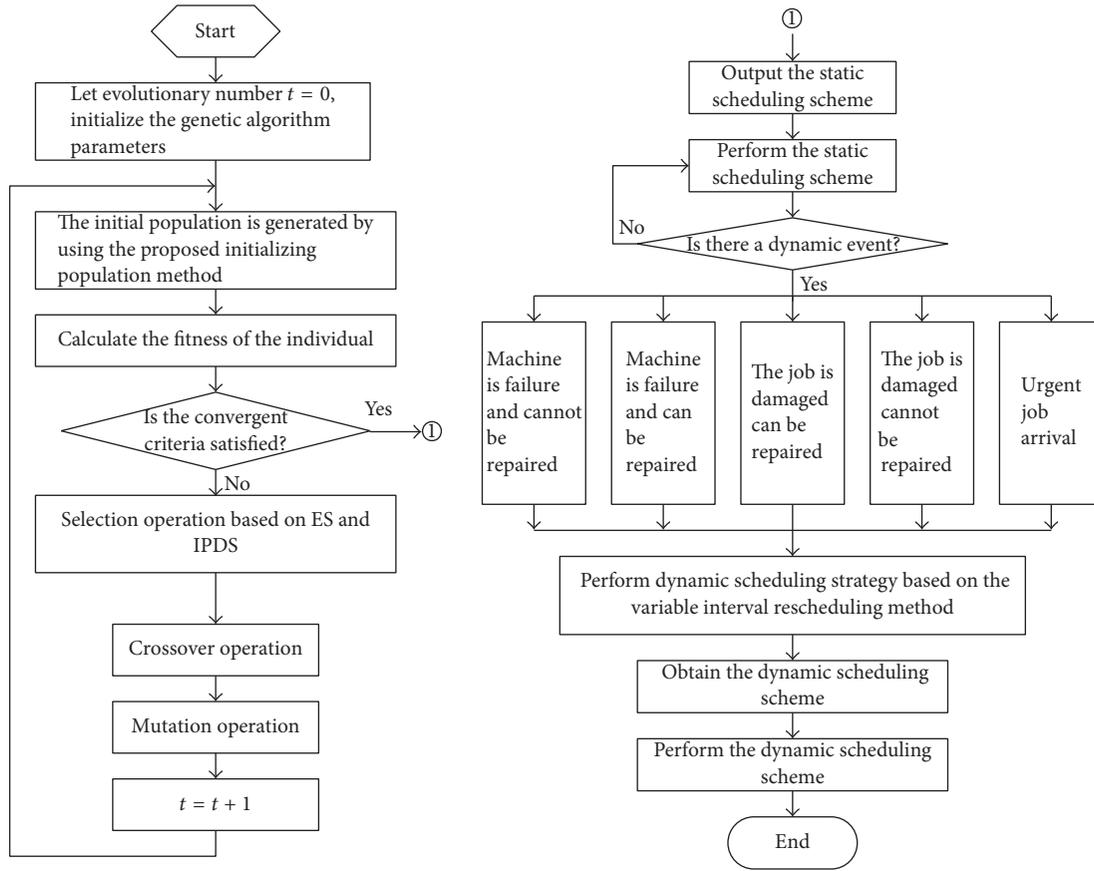


FIGURE 9: Flow chart of the proposed improved genetic algorithm and dynamic rescheduling strategy.

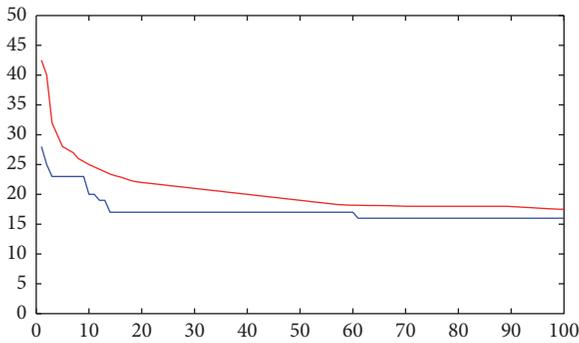


FIGURE 10: Evolutionary curve obtained by random initializing method.

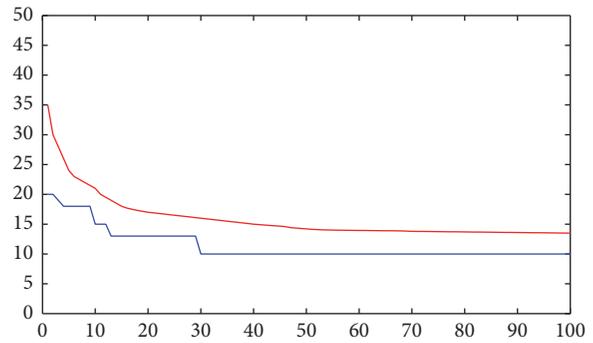


FIGURE 11: Evolutionary curve obtained by proposed initializing method.

The dynamic events often happen in an actual production environment. In this section, the machine failure is considered. Suppose that $M6$ is failure at time 20 s, the rescheduling Gantt chart is shown in Figure 14, and the comparison results for rescheduling are shown in Table 6.

From Table 6, it is not difficult to see that the involving operations of the jobs that need to be processed on the failure machine can be directly transferred to other machines with considerable processing capacity after using the dynamic scheduling strategy based on the variable interval

rescheduling method proposed in this paper, and there is no need to wait for the failure machine to continue processing the involving operations after repairing, so that the completion time is greatly reduced.

4.3. *Dynamic Rescheduling for Some Other Dynamic Events.* In Section 4.2, the machine failure is considered. In order to further verify the effectiveness of our proposed improved genetic algorithm and dynamic rescheduling strategy based on variable interval scheduling, the following dynamic events

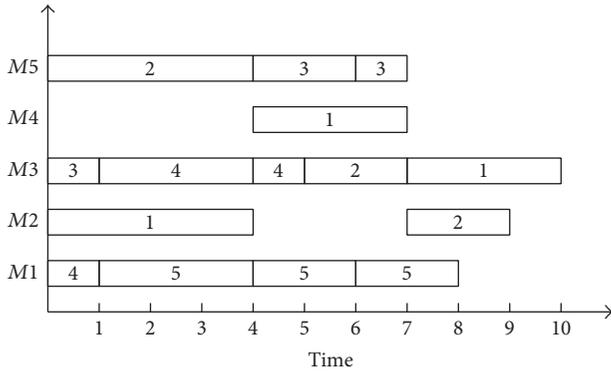


FIGURE 12: Gantt chart.

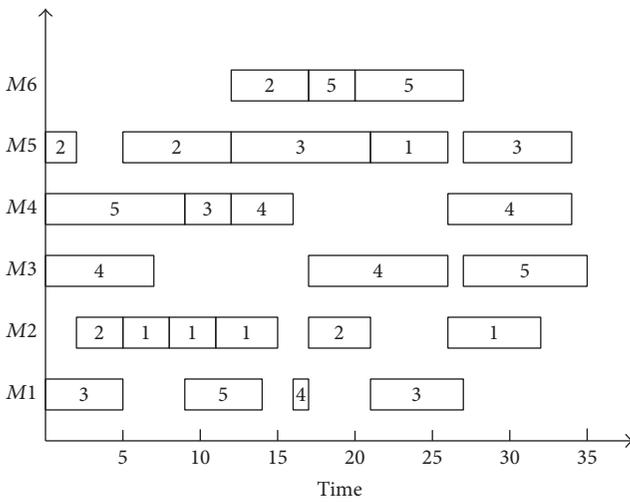


FIGURE 13: Static scheduling Gantt chart.

are considered: (1) machine failure: it can be repaired after a period of time; here we name this kind of dynamic events as DE1; (2) the job is damaged and it cannot be repaired; here we name this kind of dynamic events as DE2; (3) the job is damaged and it can be repaired; here we name this kind of dynamic events as DE3; (4) urgent job arrival; here we name this kind of dynamic events as DE4.

The processing information in Table 5 is still used in our experiments; the dynamic event descriptions are shown in Table 7. The static scheduling scheme is shown in Figure 13.

(1) *Dynamic Rescheduling for DE1.* When DE1 happens, M6 cannot be used between time 20 and time 30. The normal scheduling result is shown in Figure 15. However, if dynamic rescheduling strategy based on variable interval scheduling is carried out, the rescheduling result is shown in Figure 16. From Figures 15 and 16, it can be seen that the scheduling time is reduced 10 units after performing our proposed dynamic rescheduling strategy.

(2) *Dynamic Rescheduling for DE2.* When DE2 happens at time 20, job 4 is damaged and it cannot be repaired; therefore job 4 needs to be rescheduled. The normal scheduling result

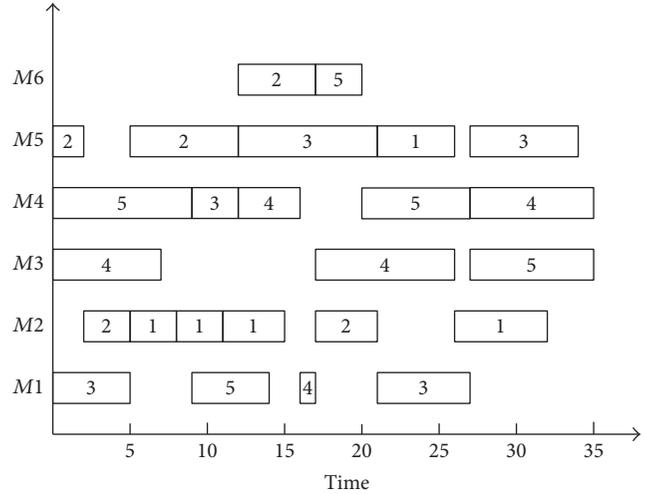


FIGURE 14: Rescheduling Gantt chart after M6 failure.

TABLE 5: Processing information for 5-job and 6-machine FJSP.

Job	Operation	Processing time					
		M1	M2	M3	M4	M5	M6
J1	O11	2	3	4	—	—	—
	O12	—	3	—	2	4	—
	O13	1	4	5	—	—	—
	O14	4	—	—	3	5	—
	O15	—	6	8	7	6	9
J2	O21	3	—	5	—	2	—
	O22	4	3	—	—	6	—
	O23	—	—	4	—	7	11
	O24	—	5	—	7	—	5
	O25	4	5	7	—	5	—
J3	O31	5	6	—	—	—	—
	O32	—	4	—	3	5	—
	O33	—	—	13	—	9	12
	O34	6	5	7	4	8	—
	O35	8	6	—	—	7	8
J4	O41	9	—	7	9	—	—
	O42	—	6	—	4	—	5
	O43	1	—	3	—	—	3
	O44	6	—	9	7	5	4
	O45	—	8	7	8	8	—
J5	O51	4	3	7	9	3	6
	O52	5	6	—	4	—	5
	O53	6	—	4	—	—	3
	O54	—	5	—	7	—	7
	O55	7	—	8	7	8	—

is shown in Figure 17. However, if the dynamic rescheduling strategy is carried out, the rescheduling result is shown in Figure 18. From Figures 17 and 18, it can be seen that the scheduling time is reduced by 9 units after performing our proposed dynamic rescheduling strategy.

TABLE 6: Comparison results for rescheduling.

	Actual enterprise's scheduling result	Method proposed in [33]	Our proposed method
Failure time	20	20	20
Machine repair time	30	0	0
Ideal makespan	37	37	35
Actual makespan	67	37	35

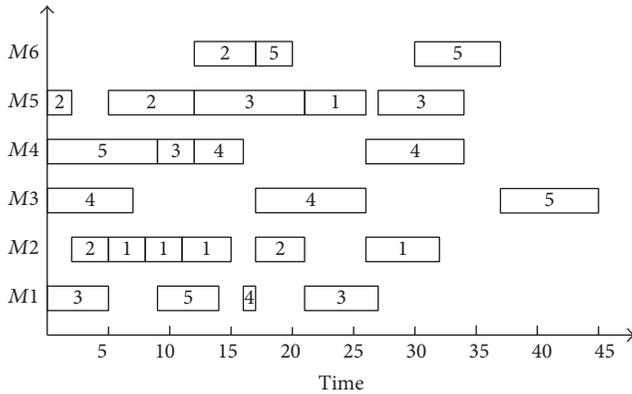


FIGURE 15: Scheduling Gantt chart by waiting for M6 repair.

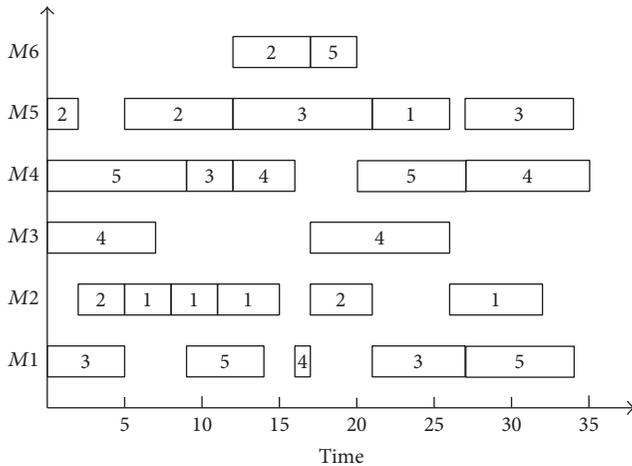


FIGURE 16: Rescheduling Gantt chart by using dynamic rescheduling strategy.

(3) *Dynamic Rescheduling for DE3.* When DE3 happens at time 20, the fourth operation of job 4 is damaged and it can be repaired; therefore this operation needs to be reprocessed. The normal scheduling result is shown in Figure 19. However, if the dynamic rescheduling strategy is carried out, the rescheduling result is shown in Figure 20. From Figures 21 and 22, it can be seen that the scheduling time is reduced by 2 units after performing our proposed dynamic rescheduling strategy.

(4) *Dynamic Rescheduling for DE4.* When DE4 happens at time 20, urgent job 6 arrives, and it is necessary to be added

TABLE 7: Information for dynamic events.

Type of dynamic events	Dynamic event descriptions
DE1	At time 20, M6 is failure, and it is repaired after time 30
DE2	At time 20, job 4 is damaged and it cannot be repaired
DE3	At time 20, the fourth operation of job 4 is damaged and it can be repaired
DE4	At time 20, an urgent job 6 arrives

TABLE 8: Processing information of job 6.

Job	Operation	Processing time					
		M1	M2	M3	M4	M5	M6
J6	O61	6	7	3	6	2	5
	O62	6	7	2	6	7	2
	O63	2	2	5	5	2	3
	O64	3	4	7	7	2	3
	O65	3	3	5	7	5	5

TABLE 9: Comparisons of dynamic events.

Dynamic event	Makespan before optimization	Makespan after optimization
DE1	45	35
DE2	52	43
DE3	37	35
DE4	50	37

to the current scheduling sequences. The processing information of job 6 is shown in Table 8. The normal scheduling result is shown in Figure 21. However, if the dynamic rescheduling strategy is carried out, the rescheduling result is shown in Figure 22. From Figures 21 and 22, it can be seen that the scheduling time is reduced by 13 units after performing our proposed dynamic rescheduling strategy.

The scheduling results for these dynamic events are shown in Table 9. It can be seen from Table 9 that the makespans obtained by using our proposed method for all these dynamic events are better than that before optimization. Therefore, we firmly believe that the proposed method is feasible and effective for static and dynamic events.

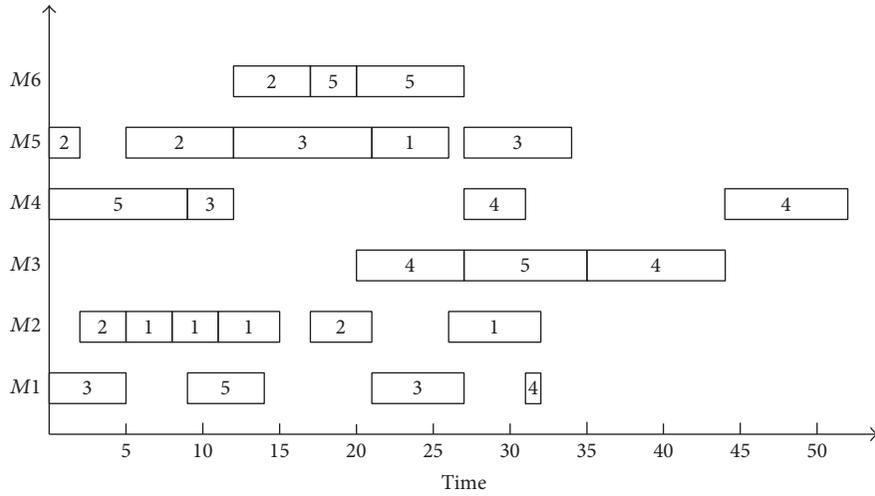


FIGURE 17: Scheduling Gantt chart for DE2 before optimizing.

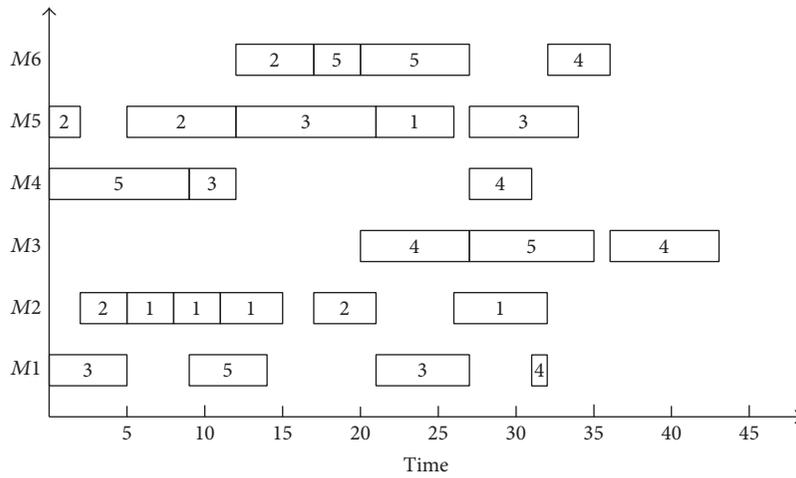


FIGURE 18: Scheduling Gantt chart for DE2 after optimizing.

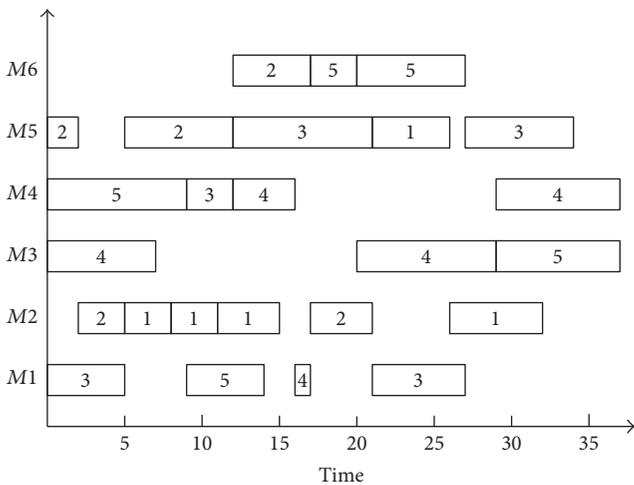


FIGURE 19: Scheduling Gantt chart for DE3 before optimizing.

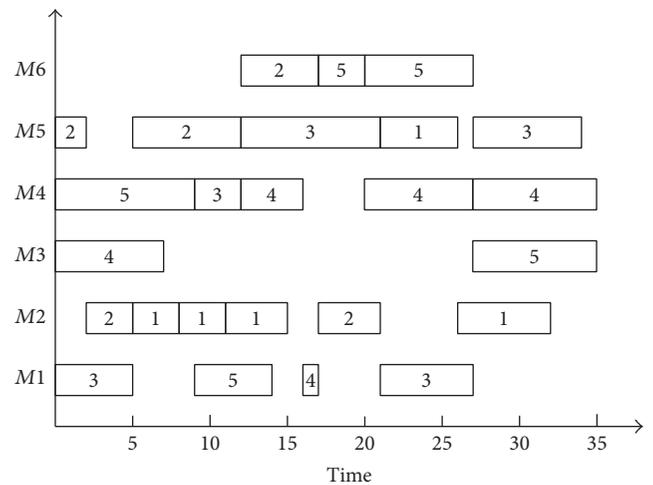


FIGURE 20: Scheduling Gantt chart for DE3 after optimizing.

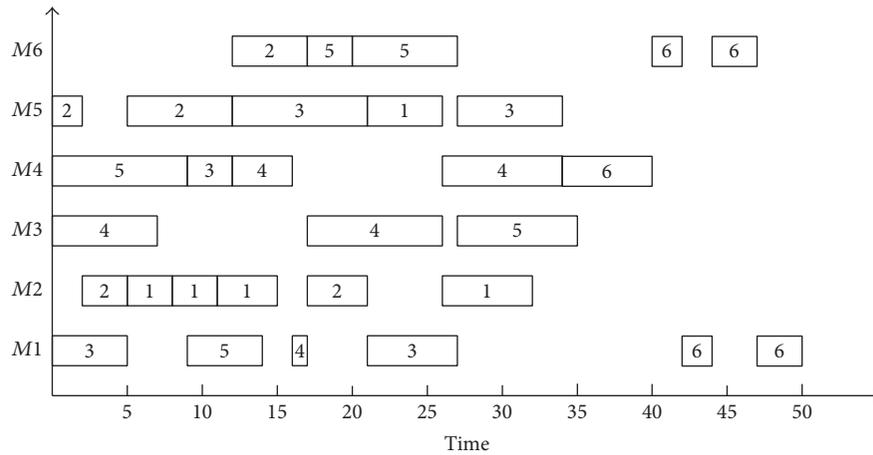


FIGURE 21: Scheduling Gantt chart for DE4 before optimizing.

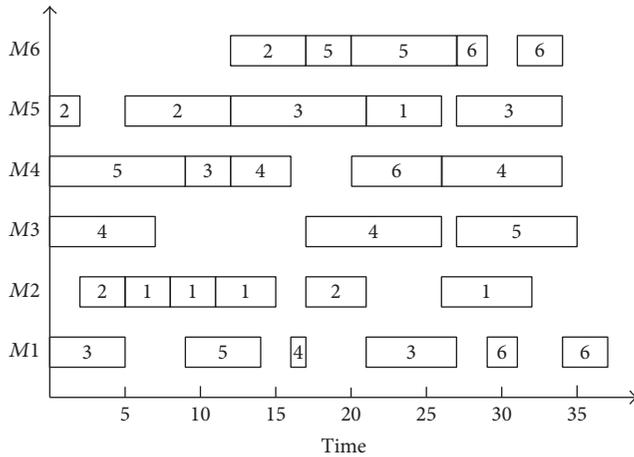


FIGURE 22: Scheduling Gantt chart for DE4 after optimizing.

5. Conclusions

The flexible job shop scheduling problem is known as NP-hard combinatorial optimization, and dynamic flexible job shop scheduling is very important to the implementation of the real-world manufacturing systems. In order to respond to dynamic flexible job shop scheduling problem for minimizing makespan, this paper develops an improved genetic algorithm based on variable interval rescheduling strategy for generating new schedules. Several dynamic events are carried out to evaluate the performance of the proposed method. Experimental results for static and several dynamic events in the FJSP demonstrate that our method is feasible and effective.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This paper is supported by Anhui Provincial Natural Science Foundation (Grant no. 1708085ME129), the National Natural Science Foundation of China (Grant no. 51305001), and Key Support Projects for Outstanding Young Talents of Anhui Province Universities (Grant no. gxyqZD2016125).

References

- [1] W. Teekeng and A. Thammano, "Modified genetic algorithm for flexible job-shop scheduling problems," *Procedia Computer Science*, vol. 12, no. 12, pp. 122–128, 2012.
- [2] G. Zhang, L. Gao, and Y. Shi, "An effective genetic algorithm for the flexible job-shop scheduling problem," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3563–3573, 2011.
- [3] X.-N. Shen and X. Yao, "Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems," *Information Sciences*, vol. 298, pp. 198–224, 2015.
- [4] P. Fattahi, M. S. Mehrabad, and F. Jolai, "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. 18, no. 3, pp. 331–342, 2007.
- [5] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Mathematics of Operations Research*, vol. 1, no. 2, pp. 117–129, 1976.
- [6] N. Shivasankaran, P. S. Kumar, and K. V. Raja, "Hybrid sorting immune simulated annealing algorithm for flexible job shop scheduling," *International Journal of Computational Intelligence Systems*, vol. 8, no. 3, pp. 455–466, 2015.
- [7] G. Vilcot and J.-C. Billaut, "A tabu search algorithm for solving a multicriteria flexible job shop scheduling problem," *International Journal of Production Research*, vol. 49, no. 23, pp. 6963–6980, 2011.
- [8] A. Rossi, "Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships," *International Journal of Production Economics*, vol. 153, pp. 253–267, 2014.

- [9] D. Tang, M. Dai, M. A. Salido, and A. Giret, "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization," *Computers in Industry*, vol. 81, pp. 82–95, 2016.
- [10] L. Wang, G. Zhou, Y. Xu, S. Wang, and M. Liu, "An effective artificial bee colony algorithm for the flexible job-shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 60, no. 1–4, pp. 303–315, 2012.
- [11] I. Driss, K. N. Mouss, and A. Laggoun, "A new genetic algorithm for flexible job-shop scheduling problems," *Journal of Mechanical Science and Technology*, vol. 29, no. 3, pp. 1273–1281, 2015.
- [12] S. H. A. Rahmati, M. Zandieh, and M. Yazdani, "Developing two multi-objective evolutionary algorithms for the multi-objective flexible job shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 64, no. 5–8, pp. 915–932, 2013.
- [13] J.-Q. Li, Q.-K. Pan, and K.-Z. Gao, "Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems," *The International Journal of Advanced Manufacturing Technology*, vol. 55, no. 9–12, pp. 1159–1169, 2011.
- [14] L. De Giovanni and F. Pezzella, "An improved genetic algorithm for the distributed and flexible job-shop scheduling problem," *European Journal of Operational Research*, vol. 200, no. 2, pp. 395–408, 2010.
- [15] Q. Zhang, H. Manier, and M.-A. Manier, "A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times," *Computers & Operations Research*, vol. 39, no. 7, pp. 1713–1723, 2012.
- [16] G. Zhang, L. Gao, and Y. Shi, "A novel variable neighborhood genetic algorithm for multi-objective flexible job-shop scheduling problems," *Advanced Materials Research*, vol. 118–120, pp. 369–373, 2010.
- [17] D. Rahmani and R. Ramezani, "A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: a case study," *Computers and Industrial Engineering*, vol. 98, pp. 360–372, 2016.
- [18] B. Liu, Y. Fan, and Y. Liu, "A fast estimation of distribution algorithm for dynamic fuzzy flexible job-shop scheduling problem," *Computers and Industrial Engineering*, vol. 87, pp. 193–201, 2015.
- [19] J. Xiong, L.-N. Xing, and Y.-W. Chen, "Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns," *International Journal of Production Economics*, vol. 141, no. 1, pp. 112–126, 2013.
- [20] N. Al-Hinai and T. Y. Elmekawy, "Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm," *International Journal of Production Economics*, vol. 132, no. 2, pp. 279–291, 2011.
- [21] L. Liu, H. Y. Gu, and Y. G. Xi, "Robust and stable scheduling of a single machine with random machine breakdowns," *International Journal of Advanced Manufacturing Technology*, vol. 31, no. 7–8, pp. 645–654, 2007.
- [22] F. Zhao, J. Wang, J. Wang, and J. Jonrinaldi, "A dynamic rescheduling model with multi-agent system and its solution method," *Strojnicki Vestnik/Journal of Mechanical Engineering*, vol. 58, no. 2, pp. 81–92, 2012.
- [23] L. Nie, L. Gao, P. Li, and X. Shao, "Reactive scheduling in a job shop where jobs arrive over time," *Computers and Industrial Engineering*, vol. 66, no. 2, pp. 389–405, 2013.
- [24] M. A. Adibi, M. Zandieh, and M. Amiri, "Multi-objective scheduling of dynamic job shop using variable neighborhood search," *Expert Systems with Applications*, vol. 37, no. 1, pp. 282–287, 2010.
- [25] P. Fattahi and A. Fallahi, "Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability," *CIRP Journal of Manufacturing Science and Technology*, vol. 2, no. 2, pp. 114–123, 2010.
- [26] N. Kundakci and O. Kulak, "Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem," *Computers and Industrial Engineering*, vol. 96, pp. 31–51, 2016.
- [27] E. Ahmadi, M. Zandieh, M. Farrokhi, and S. M. Emami, "A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms," *Computers and Operations Research*, vol. 73, pp. 56–66, 2016.
- [28] X. Qiu and H. Y. K. Lau, "An AIS-based hybrid algorithm with PDRs for multi-objective dynamic online job shop scheduling problem," *Applied Soft Computing Journal*, vol. 13, no. 3, pp. 1340–1351, 2013.
- [29] L. Zhang, L. Gao, and X. Li, "A hybrid genetic algorithm and tabu search for a multi-objective dynamic job shop scheduling problem," *International Journal of Production Research*, vol. 51, no. 12, pp. 3516–3531, 2013.
- [30] A. A. R. Hosseinabadi, H. Siar, S. Shamshirband, M. Shojafar, and M. H. N. Md. Nasir, "Using the gravitational emulation local search algorithm to solve the multi-objective flexible dynamic job shop scheduling problem in Small and Medium Enterprises," *Annals of Operations Research*, vol. 229, no. 1, pp. 451–474, 2014.
- [31] M.-S. Lu and R. Romanowski, "Multicontextual dispatching rules for job shops with dynamic job arrival," *International Journal of Advanced Manufacturing Technology*, vol. 67, no. 1–4, pp. 19–33, 2013.
- [32] H. Karimi, S. H. A. Rahmati, and M. Zandieh, "An efficient knowledge-based algorithm for the flexible job shop scheduling problem," *Knowledge-Based Systems*, vol. 36, no. 6, pp. 236–244, 2012.
- [33] Z. Wu, H. He, and C. Huang, "Flexible job shop dynamic scheduling problem research with machine fault [J]," *Machine Design and Research*, vol. 31, no. 3, pp. 94–98, 2015.
- [34] I. Kacem, S. Hammadi, and P. Borne, "Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 32, no. 1, pp. 1–13, 2002.
- [35] J. Tang, G. Zhang, B. Lin, and B. Zhang, "A hybrid algorithm for flexible job-shop scheduling problem," *Procedia Engineering*, vol. 15, no. 1, pp. 3678–3683, 2011.
- [36] X. Li and L. Gao, "An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem," *International Journal of Production Economics*, vol. 174, pp. 93–110, 2016.

Research Article

Motion Planning for Omnidirectional Wheeled Mobile Robot by Potential Field Method

Weihaio Li,¹ Chenguang Yang,^{1,2} Yiming Jiang,¹ Xiaofeng Liu,^{3,4,5} and Chun-Yi Su¹

¹Key Lab of Autonomous Systems and Networked Control, Ministry of Education, South China University of Technology, Guangzhou 510640, China

²Zienkiewicz Centre for Computational Engineering, Swansea University, Swansea SA1 8EN, UK

³College of IoT Engineering, Hohai University, Nanjing, China

⁴Jiangsu Key Laboratory of Special Robots, Hohai University, Nanjing, China

⁵Changzhou Key Laboratory of Robotics and Intelligent Technology, Changzhou, China

Correspondence should be addressed to Chenguang Yang; cyang@ieee.org

Received 30 December 2016; Accepted 7 February 2017; Published 12 March 2017

Academic Editor: Cheng S. Chin

Copyright © 2017 Weihaio Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, potential field method has been used to navigate a three omnidirectional wheels' mobile robot and to avoid obstacles. The potential field method is used to overcome the local minima problem and the goals nonreachable with obstacles nearby (GNRON) problem. For further consideration, model predictive control (MPC) has been used to incorporate motion constraints and make the velocity more realistic and flexible. The proposed method is employed based on the kinematic model and dynamics model of the mobile robot in this paper. To show the performance of proposed control scheme, simulation studies have been carried to perform the motion process of mobile robot in specific workplace.

1. Introduction

In recent decades, omnidirectional mobile robot (OMR) has attracted increasing attention and investigation from the research communities [1–3]. One of advantages of OMR using omnidirectional wheels is that it does not have non-holonomic constraint which exists in differentially driven mobile robot [4–6]. With the input of the rotating speed of each omnidirectional wheel, the mobile robot can easily move wherever the user wants. This simplifies the control law which can be achieved easily. As it is shown in Figure 1, omnidirectional wheel consists of wheel and rollers, which means that the speed of the whole omnidirectional wheel is the combination of wheel speed and roller speed. Robot's control is very complicated, and sometimes it is necessary to consider state constraint of the robot to complete the control design [7, 8].

Since the path planning problem has been put forward, it has been studied by numbers of researchers [9]. A large number of research results have been proposed. The path planning algorithm develops from the earliest grid method, artificial potential field method [10], visibility graph [11] to

C-space method [12], A* algorithm, and D* algorithm [13]. Now, it is also studied to combine fuzzy logic algorithm [14, 15], adaptive algorithm [16, 17], and neural network algorithm [18–21]. In recent years, potential field method is more and more mature and widely used in omnidirectional mobile robots, because of its logical simplicity and obstacle avoidance capability. Many researches have proved the excellent capability of navigation and obstacle avoidance [22–24]. Hence, potential field method is utilized in this paper for the motion planning of omnidirectional wheeled mobile robot. To overcome the local minima problem and the goals nonreachable with obstacles nearby (GNRON) problem, the repulsive potential functions for motion planning contain the distance between robot and obstacle.

A popular way to control a mobile robot is to design the kinematic control based only on the kinematics equation [25–27]. Since 1995, people have put forward an integral dynamics model of a mobile robot [4]. Using dynamics model to control robot's motion is a common way [28–31]. This paper combines the kinematics as well as dynamics equation of the omnidirectional wheel and potential field method, to

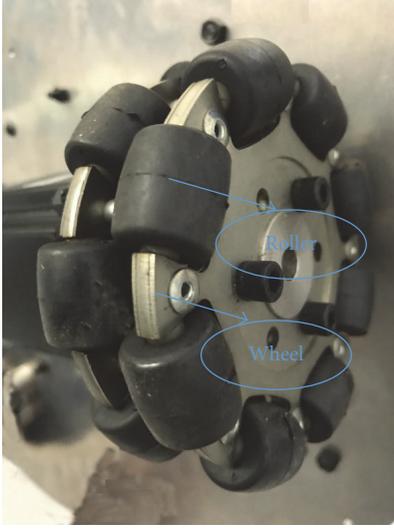


FIGURE 1: The structure of i omnidirectional wheel.

control and navigate the mobile robot. In addition to these contributions in this paper, model predictive control (MPC) is utilized in motion planning for robust controller performance [32–35]. This paper is organized as follows: in Section 2, the kinematics equation of omnidirectional wheel and how the mobile robot built with 3 omnidirectional wheels can achieve the omnidirectional motion are discussed. In Section 3, the dynamics model of 3-omnidirectional wheel mobile robot is explained. In Section 4, a novel potential field method, which can overcome the GNRON problem, is introduced. In Section 5, MPC has been introduced. Both kinematics model and dynamics model have been applied in MPC. In Section 6, the simulations illustrating the effectiveness of the proposed method are presented. Finally, conclusion is given in Section 7.

2. The Kinematics Equation of Omnidirectional Wheel

From Figure 2, the following equation can be obtained:

$$\begin{bmatrix} v'_{ix} \\ v'_{iy} \end{bmatrix} = \begin{bmatrix} 0 & \sin \alpha_i \\ r & \cos \alpha_i \end{bmatrix} \begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} = K_{i1} \begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix}, \quad (1)$$

where v_{ix} , v_{iy} , and ω_i are generalized velocity of point O'_i in Cartesian coordinate system and v'_{ix} , v'_{iy} , and ω'_i are generalized velocity of point O'_i in $x'O'y'$ coordinate system. v_{ir} is the i th roller's central velocity vector.

When the i th omnidirectional wheel's central speed is mapped to Cartesian coordinate system, then

$$\begin{bmatrix} v_{ix} \\ v_{iy} \end{bmatrix} = \begin{bmatrix} \cos \eta_i & -\sin \eta_i \\ \sin \eta_i & \cos \eta_i \end{bmatrix} \begin{bmatrix} v'_{ix} \\ v'_{iy} \end{bmatrix} = K_{i2} K_{i1} \begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix}. \quad (2)$$

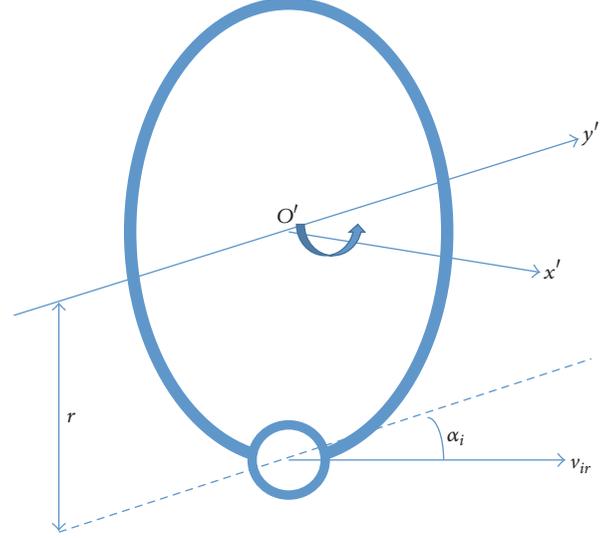


FIGURE 2: The motion relationship between system center and the i omnidirectional wheel.

The system moves in two-dimensional space, so, according to geometric relationship and [36], the wheels' speed can be represented by

$$\begin{bmatrix} v_{ix} \\ v_{iy} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -l_{iy} \\ 0 & 1 & l_{ix} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = K_{i3} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \omega \end{bmatrix}, \quad (3)$$

where l_{ix} and l_{iy} are the position of the i th wheel's mass point in xOy coordinate. According to (2)-(3), the system inverse kinematics equations can be defined as

$$K_{i2} K_{i1} \begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} = K_{i3} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \omega \end{bmatrix}, \quad i = 1, 2, 3, \quad (4)$$

where $\det(K_{i1})$ is not zero, so as the $\det(K_{i2})$. Define $K_i = [K_{i2}]^{-1} [K_{i1}]^{-1} K_{i3}$, $l_{ix} = l_i \cos \beta_i$, and $l_{iy} = l_i \sin \beta_i$. β_i is the angle between OO'_i and x -axis. Then the inverse kinematics equation of i th omnidirectional wheel is

$$\begin{bmatrix} \omega_i \\ v_{ir} \end{bmatrix} = [K_{i2}]^{-1} [K_{i1}]^{-1} K_{i3} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \omega \end{bmatrix}; \quad (5)$$

define $\gamma_i = \eta_i - \alpha_i$; then

$$K_i = \frac{1}{-r \sin \alpha_i} \cdot \begin{bmatrix} \cos(\gamma_i) & \sin(\gamma_i) & -l_{iy} \cos(\gamma_i) + l_{ix} \sin(\gamma_i) \\ r \cos \eta_i & -r \sin \eta_i & l_{iy} \cdot r \cos \eta_i - l_{ix} \cdot r \sin \eta_i \end{bmatrix}. \quad (6)$$

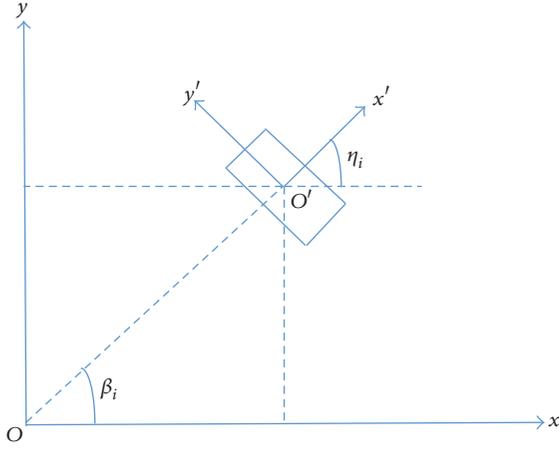
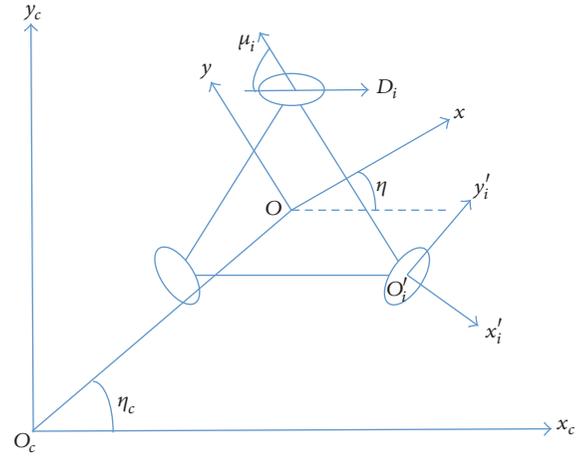
FIGURE 3: The structure of i th omnidirectional wheel.

FIGURE 4: All three coordinates in OMR system.

Define

$$A_1 = \begin{bmatrix} \cos(\gamma_1) \\ \sin \alpha_1 \\ \cos(\gamma_2) \\ \sin \alpha_2 \\ \cos(\gamma_3) \\ \sin \alpha_3 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \frac{\sin(\gamma_1)}{\sin \alpha_1} \\ \frac{\sin(\gamma_2)}{\sin \alpha_2} \\ \frac{\sin(\gamma_3)}{\sin \alpha_3} \end{bmatrix} \quad (7)$$

$$A_3 = \begin{bmatrix} \frac{l_1 \sin(\gamma_1 - \beta_1)}{\sin \alpha_1} \\ \frac{l_2 \sin(\gamma_2 - \beta_2)}{\sin \alpha_2} \\ \frac{l_3 \sin(\gamma_3 - \beta_3)}{\sin \alpha_3} \end{bmatrix}.$$

The inverse kinematics solution of wheel speed to system center is

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{-r} [A_1 \ A_2 \ A_3] \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \omega \end{bmatrix}. \quad (8)$$

The Jacobian matrix systems inverse kinematics equation is

$$R = \frac{1}{-r} [A_1 \ A_2 \ A_3]. \quad (9)$$

According to Figures 2–4, since this paper discusses a mobile robot built by three omnidirectional wheels, α_i , β_i , and η_i , are fixed. The actual values are shown in Table 1.

TABLE 1: A table with notes.

Parameter	β_1	β_2	β_3	η_1	η_2	η_3	α_1	α_2	α_3
Value	60°	180°	300°	60°	180°	300°	90°	90°	90°

From Table 1 and specification that $r = 50.67$ mm and $l = 118.18$ mm, the actual parameter of OMR can be substituted into (8); then

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} -0.0170 & 0.00987 & 2.3323 \\ 0 & -0.0197 & 2.3323 \\ 0.0171 & 0.00987 & 2.3323 \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \omega \end{bmatrix}. \quad (10)$$

From (10), $\text{rank}(R) = 3$, which means that this robot can achieve omnidirectional movement.

3. The Dynamics Equation of OMR

Three coordinates $x_c O_c y_c$, xOy , and $x'_i O'_i y'_i$ are constructed as in Figure 4. O_c is a specific point in the workspace, and O is the central point of the mobile robot while O'_i is the central point of each wheel. η is the angle between the front of OMR and X_c . The vector $S_c = [x_c, y_c]^T$ indicates the position of O . According to Newton's second law,

$$m\ddot{x}_c = F_{x_c} \quad (11)$$

$$m\ddot{y}_c = F_{y_c},$$

equivalent to

$$M\ddot{S}_c = F_c. \quad (12)$$

$F = [F_{x_c}, F_{y_c}]^T$ is a force vector on the central point of mobile robot in the Cartesian coordinate. $M = \text{diag}(m, m)$ is a symmetric positive definite matrix, and m is the mass of mobile robot.

The transfer matrix which transfers coordinate $x_c O_c y_c$ into coordinate xOy is

$$T_c = \begin{bmatrix} \cos \eta & -\sin \eta \\ \sin \eta & \cos \eta \end{bmatrix}. \quad (13)$$

Consequently,

$$\begin{aligned}\dot{S}_c &= T_c \dot{s} \\ \dot{F}_c &= T_c \dot{f}\end{aligned}\quad (14)$$

$s = [x, y]^T$ and $f = [f_x, f_y]$, respectively, mean the central point O 's displacement and force vector in mobile coordinate xOy . Therefore, (12) becomes

$$M(T_c^T \dot{T}_c \dot{s} + \ddot{s}) = f. \quad (15)$$

Finally, the dynamics of the omnidirection mobile robot can be described as

$$\begin{aligned}m(\ddot{x} - \dot{y}\dot{\eta}) &= f_x \\ m(\ddot{y} + \dot{x}\dot{\eta}) &= f_y \\ I_v \ddot{\eta} &= M_I,\end{aligned}\quad (16)$$

where I_v and M_I are the moment of inertia of mobile robot around its central axis and the corresponding torque, respectively. Among them, f_x , f_y , and M_I can be drawn by

$$\begin{aligned}f_x &= -(N_1^w + N_2^w) \sin \mu + N_3 \\ f_y &= (N_1^w - N_2^w) \cos \mu,\end{aligned}\quad (17)$$

where μ is the angle between wheel and y -axis, and $\mu_1 = 30^\circ$, $\mu_2 = 30^\circ$, and $\mu_3 = 90^\circ$.

$$M_I = I_1 N_1^w + I_2 N_2^w + I_3 N_3^w. \quad (18)$$

According to [37], the dynamics model of drive system of each wheel is assumed as

$$I_w \dot{\omega}_i + \xi \omega_i = h \tau_i - r N_i^w \quad (i = 1, 2, 3), \quad (19)$$

where N_i^w is the drive power of each wheel. I_w is the moment of inertia of wheel around its central axis. ξ is viscous friction constant between wheel and ground. $\dot{\omega}_i$ is angular acceleration of each wheel. r is radius of the wheel. h is drive factor. τ_i is the input torque of each wheel.

The speed of each omnidirectional mobile robot's wheel v_i can be described as $r\omega_i$. According to [38], the dynamics model of OMR can be described as the following equation:

$$\ddot{q} = A(q) \dot{q} + B(q) \tau, \quad (20)$$

where

$$\begin{aligned}A(q) &= \begin{bmatrix} \frac{-3\xi}{3I_w + 2Mr^2} & \frac{-3I_w \dot{\eta}}{3I_w + 2Mr^2} & 0 \\ \frac{3I_w \dot{\eta}}{3I_w + 2Mr^2} & \frac{3\xi}{3I_w + 2Mr^2} & 0 \\ 0 & 0 & \frac{-3\xi l^2}{3I_w l^2 + I_v r^2} \end{bmatrix} \\ B(q) &= \begin{bmatrix} \frac{-hr\kappa_1}{3I_w + 2Mr^2} & \frac{hr\kappa_1}{3I_w + 2Mr^2} & \frac{2hr \cos \eta}{3I_w + 2Mr^2} \\ \frac{hr\kappa_2}{3I_w + 2Mr^2} & \frac{hr\kappa_3}{3I_w + 2Mr^2} & \frac{2hr \sin \eta}{3I_w + 2Mr^2} \\ \frac{hr l}{3I_w l^2 + I_v r^2} & \frac{hr l}{3I_w l^2 + I_v r^2} & \frac{hr l}{3I_w l^2 + I_v r^2} \end{bmatrix}.\end{aligned}\quad (21)$$

Let us define $\kappa_1 = \sqrt{3} \sin \eta - \cos \eta$, $\kappa_2 = -\sqrt{3} \cos \eta - \sin \eta$, and $\kappa_3 = \sqrt{3} \cos \eta - \sin \eta$.

4. Potential Field for OMR's Motion Planning

Using the potential field algorithm for OMR's path planning will be modified to produce a virtual force to navigate mobile robot and obstacle avoidance. For simple theoretical analysis, mobile robot is considered as a mass point and moves in two-dimensional space whose position can be denoted by $s = [x, y]^T$. The distance as well as the angle between robot and goal, robot and obstacles can be detected by ultrasonic sensors. Inspired by [23], then the attractive potential function caused by goal can be calculated by the following equation:

$$U_{\text{att}}(s) = \frac{1}{2} a_{\text{att}} d^c(s, s_{\text{goal}}), \quad (22)$$

where a_{att} is a positive scaling factor, $d^c(s, s_{\text{goal}}) = \|(s_{\text{goal}} - s)\|$ is the distance between the OMB's mass point and the goal s_{goal} , and $c = 1$ or 2 . For $c = 2$, the attractive force is

$$F_{\text{att}}(s) = -\nabla U_{\text{att}}(s) = a_{\text{att}} (s_{\text{goal}} - s) = F_{\text{att}} n_{\text{RG}}. \quad (23)$$

The repulsive potential function is

$$\begin{aligned}U_{\text{rep}}(s) &= \begin{cases} \frac{1}{2} a_{\text{rep}} \left(\frac{1}{d(s, s_{\text{obs}})} - \frac{1}{d_0} \right)^2 d^m(s, s_{\text{goal}}), & \text{if } d(s, s_{\text{obs}}) \leq d_0 \\ 0, & \text{if } d(s, s_{\text{obs}}) > d_0. \end{cases}\end{aligned}\quad (24)$$

where a_{rep} is a positive scaling factor, $d(s, s_{\text{obs}})$ is the minimal distance between the OMB's mass point q and the hindrance, d_0 denotes the level of the influence of the hindrance to the robot and it is defined as a positive constant, and n is a positive constant. The repulsive force is

$$\begin{aligned}F_{\text{rep}}(s) &= -\nabla U_{\text{rep}}(s) \\ &= \begin{cases} F_{\text{rep1}} n_{\text{OR}} + F_{\text{rep2}} n_{\text{RG}}, & \text{if } d(s, s_{\text{obs}}) \leq d_0 \\ 0, & \text{if } d(s, s_{\text{obs}}) > d_0. \end{cases}\end{aligned}\quad (25)$$

$n_{OR} = \nabla d(s, s_{obs})$ and $n_{RG} = -\nabla d(s, s_{goal})$ are two unit vectors pointing from the obstacle to the OMR and from the OMR to the goal. According to (23) and (25), F_{total} is drawn by the following equation:

$$\begin{aligned} F_{total} &= F_{att}(s) + F_{rep}(s) \\ &= F_{att}n_{RG} + F_{rep1}n_{OR} + F_{rep2}n_{RG}. \end{aligned} \quad (26)$$

According to [23], for the robot with 3 omnidirectional wheels, the real input is the 3 angular velocities of the omnidirectional wheels, ω_1, ω_2 , and ω_3 , which satisfies (8), and v_{x_c} and v_{y_c} have the following relationship with v and ω :

$$\begin{aligned} v_{x_c} &= v \cos \eta \\ v_{y_c} &= v \sin \eta. \end{aligned} \quad (27)$$

The mobile robot needs to decelerate as soon as it nears the obstacle, while its velocity will be higher when it is far from the obstacle, so the robot's velocity is chosen by the distance between OMB and obstacle. Thus the velocity of the OMR in $x_c O_c y_c$ can be determined by

$$v = \begin{cases} v_o \\ \frac{d(s, s_{obs})}{d_0} v_o, \end{cases} \quad (28)$$

where v_o is the optimal velocity of the robot.

As the total force F_{total} can be calculated, its angle η_f is known. The difference angle between η_f and the orientation of the robot η_c is

$$\eta_e = \eta_f - \eta_c. \quad (29)$$

Thus the angular velocities ω can be ensured by

$$\omega = K\eta_e; \quad (30)$$

define K as a positive gain.

5. Model Predictive Control for Omnidirection Mobile Robot

In recent years, MPC has been widely used in motion control Internet of things applications [35, 39]. MPC has low requirements for model's accuracy and it is suitable for step response model and linear and nonlinear model. The control problem is described as a cost function's optimization problem. The input which is constrained by some specific conditions and minimizes the cost function is the optimal input. One of MPC's advantage is its rolling optimization [40] that means, according to its reference, it can optimize a cost function to get an optimal input vector at every sample time. According to the MPC method introduced by [41], because the reference is produced by potential field, what we need is a discrete-time model with constraints. In the following section, according to [35] two discrete-time controllers, kinematic controller, and dynamics controller are proposed.

5.1. Kinematic Controller. With (8) and $v = r\omega$, then the kinematics model of mobile robot can be transformed into

$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = -[A_1 \ A_2 \ A_3] \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \omega \end{bmatrix}. \quad (31)$$

Define $q = [x_c, y_c, \eta]^T$ as the state of mobile robot in $x_c O_c y_c$ and $S(q) = (rR)^{-1}$. Therefore

$$\dot{q} = -S(q)v. \quad (32)$$

With the help of zero-order hold (ZOH), a continuous-time system can be described as a discrete-time form

$$q(j+1) = q(j) + \dot{q}(j)T. \quad (33)$$

with a sampling period T . According to (31), (32), and (33) can be rewritten as

$$\begin{aligned} q(j+1) &= \begin{bmatrix} x_c(j+1) \\ y_c(j+1) \\ \eta(j+1) \end{bmatrix} \\ &= \begin{bmatrix} x_c(j) - 1.1547Tv_1 - 0.5785Tv_2 + 0.5777Tv_3 \\ y_c(j) - 1.002Tv_2 + 0.334Tv_3 \\ \eta(j) + 0.0028Tv_3 \end{bmatrix} \end{aligned} \quad (34)$$

The cost function for the MPC can be defined as

$$J(q, u) = \sum_{k=j}^{j+N-1} L_K(q(k), u(k) + F(q(j+N))), \quad (35)$$

where $L_K(q, u)$ is the stage cost.

$$\begin{aligned} L_K(q, u) &= \sum_{k=1}^N q^T(k+j|j) Q_K q(k+j|j) \\ &+ \sum_{k=0}^{N_u-1} \Delta u^T(k+j|j) R_K \Delta u(k+j|j), \end{aligned} \quad (36)$$

where N is prediction horizon where $N \geq 1$ and N_u is control horizon where $1 \leq N_u \leq N$. Q_K and R_K are appropriate weighting matrices. $q(k+j|j)$ means the predicted state of the OMR and $\Delta u(k+j|j)$ means the input increment of the controller. $L_K(q, u)$ is the most used standard quadratic form in practice. By way of solving the following finite-horizon optimal control problem (FHOCQ) online:

$$u^* = \arg \min_u \{J(q \cdot u)\}. \quad (37)$$

The current control $u(j) = [v(j), \omega(j)]^T$ can be ensured at the instant time j . Because the torques generated by motors are limited by the performance of the motors, $u(j)$ has upper bound and lower bound and the change of $u(j)$ is also constrained. Thus

$$\begin{aligned} u_{\min} &\leq u(j) \leq u_{\max} \\ \Delta u_{\min} &\leq \Delta u(j) \leq \Delta u_{\max} \\ q_{\min} &\leq q(j) \leq q_{\max}. \end{aligned} \quad (38)$$

According to [42], (38) can be transformed into

$$\begin{aligned} h_i^- - \hat{P}_i &\leq \hat{u}(j) \leq h_i^+ - \hat{P}_i \\ h_i^- &\leq q(j) \leq h_i^+. \end{aligned} \quad (39)$$

The kinematic equation (34) can be described into the following form:

$$q(j+1) = g_1(q(j)) + g_2(q(j))u(j), \quad (40)$$

where g_1 and g_2 are the continuous nonlinear function, $g_1(0) = 0$, $q = [q_1, q_2, q_3]^T = [x_c, y_c, \eta]^T$ is the state vector, $u = [u_1, u_2]^T = [v, \omega]^T$ is the input vector, and

$$\begin{aligned} g_1(q) &= \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \\ g_2 &= T \begin{bmatrix} -1.1547 & -0.5785 & 0.5777 \\ 0 & -1.002 & 0.334 \\ 0 & 0 & 0.0028 \end{bmatrix}. \end{aligned} \quad (41)$$

Define the following vectors:

$$\begin{aligned} \bar{q} &= [q(j+1|j), \dots, q(j+N|j)]^T \in R^{3N} \\ \bar{u}(j) &= [u(j+1|j), \dots, u(j+N|j)]^T \in R^{2N_u} \\ \Delta \bar{u}(j) &= [\Delta u(j+1|j), \dots, \Delta u(j+N|j)]^T \\ &\in R^{2N_u}. \end{aligned} \quad (42)$$

The predicated output can be determined by the following form:

$$\bar{x} = G\Delta \bar{u}(j) + \bar{g}_1 + \bar{g}_2, \quad (43)$$

where

$$\begin{aligned} G &= \begin{bmatrix} g_2(q(\bar{N}_1)) & \cdots & 0 \\ g_2(q(\bar{N}_2)) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ g_2(q(\bar{N}_N)) & \cdots & g_2(q(\bar{N}_N)) \end{bmatrix} \\ \bar{g}_1 &= \begin{bmatrix} g_1(q(\bar{N}_1)) \\ g_1(q(\bar{N}_2)) \\ \vdots \\ g_1(q(\bar{N}_N)) \end{bmatrix} \in R^{3N} \\ \bar{g}_2 &= \begin{bmatrix} g_2(q(\bar{N}_1))u(j-1) \\ g_2(q(\bar{N}_2))u(j-1) \\ \vdots \\ g_2(q(\bar{N}_N))u(j-1) \end{bmatrix} \in R^{3N}; \end{aligned} \quad (44)$$

define $\bar{N}_i = j-1+i | j-1$.

Hence, the original optimization problem (35) can be transformed into

$$\begin{aligned} \min & \|\bar{q}(j)\|_{Q_K}^2 + \|\Delta \bar{u}(j)\|_{R_K}^2 \\ &= \min \|G\Delta \bar{u}(j) + \bar{g}_1 + \bar{g}_2\|_{Q_K}^2 + \|\Delta \bar{u}(j)\|_{R_K}^2 \end{aligned} \quad (45)$$

subject to

$$\begin{aligned} \Delta \bar{u}_{\min} &\leq \Delta \bar{u}(j) \leq \Delta \bar{u}_{\max} \\ \bar{u}_{\min} &\leq \bar{u}(j-1) \leq \Delta \bar{u}_{\max} \\ \bar{u}_{\min} &\leq \bar{u}(j-1) + I\Delta \bar{u}(j) \leq \Delta \bar{u}_{\max} \\ \bar{x}_{\min} &\leq G\Delta \bar{u}(j) + \bar{g}_1 + \bar{g}_2 \leq \bar{x}_{\max}. \end{aligned} \quad (46)$$

Problem (45) can be rewritten as a QP problem

$$\min \frac{1}{2} \Delta \bar{u}^T \bar{W} \Delta \bar{u} + \bar{H}^T \Delta \bar{u} \quad (47)$$

subject to

$$\begin{aligned} E\Delta \bar{u} &\leq \bar{b} \\ \Delta \bar{u}_{\min} &\leq \Delta \bar{u} \leq \Delta \bar{u}_{\max}, \end{aligned} \quad (48)$$

where the coefficients are

$$\begin{aligned} \bar{W} &= 2(G^T Q_K G + R_K) \\ \bar{H} &= -2G^T Q_K (\bar{g}_1 + \bar{g}_2) \\ E &= [-I \ I \ -G \ G]^T. \end{aligned} \quad (49)$$

5.2. Dynamics Controller. According the dynamics model of OMR, then

$$\begin{aligned} \ddot{x}_c &= \frac{-3I_w \dot{y}_c \dot{\eta} - 3\xi \dot{x}_c - hr\kappa_1 \tau_1 + hr\kappa_1 \tau_2 + 2hr\tau_3}{2Mr^2 + 3I_w} \\ \ddot{y}_c &= \frac{3I_w \dot{x}_c \dot{\eta} + 3\xi \dot{y}_c - hr\kappa_2 \tau_1 + hr\kappa_3 \tau_2 + 2hr\tau_3}{2Mr^2 + 3I_w} \\ \ddot{\eta} &= \frac{-3\xi l^2 \eta^2 + hlr(\tau_1 + \tau_2 + \tau_3)}{I_w r^2 + 3I_w r^2}. \end{aligned} \quad (50)$$

Applying (33) into (50) and $[\ddot{x}, \ddot{y}, \ddot{\eta}] = [\dot{v}_{x_c}, \dot{v}_{y_c}, \dot{\omega}]$ the discrete-time dynamics model of the OMR can be described as

$$\begin{aligned} v_x(j+1) &= v_x(j) + \ddot{x}T \\ v_y(j+1) &= v_y(j) + \ddot{y}T \\ \omega(j+1) &= \omega(j) + \ddot{\eta}T. \end{aligned} \quad (51)$$

According to $u = [v_{x_c}, v_{y_c}, \omega]^T$ and $q = [x_c, y_c, \eta]^T$, then

$$u(j+1) = u(j) + \ddot{q}T \quad (52)$$

Applying (52) into (40), then, we can draw

$$\begin{aligned} q(j+2) &= g_1(q(j)) + g_2(q(j))u(j) \\ &+ g_2(g_1(q(j)) + g_2(q(j))u(j))u(j+1) \end{aligned} \quad (53)$$

subject to

$$\begin{aligned} u_{\min} &\leq u(j) \leq u_{\max} \\ \tau_{\min} &\leq \tau(j) \leq \tau_{\max} \\ \ddot{q}_{\min} &\leq \ddot{q}(j) \leq \ddot{q}_{\max}, \end{aligned} \quad (54)$$

where, respectively, min and max means the lower bounds and upper bounds. τ is the motor inputs. u is the velocity. \ddot{q} is the acceleration.

According to (53), the predictive model can be formulated as

$$\begin{aligned} q(k+j|j) &= g_1(q(k+j-2|j)) \\ &+ g_2(q(k+j-2|j))u(k+j-2|k) \\ &+ g_2(g_1(q(k+j-2|j)) \\ &+ g_2(q(k+j-2|j))u(k+j-2|k))u(k+j \\ &- 1|k), \end{aligned} \quad (55)$$

where $k \in [1, \bar{N}]$.

The quadratic objective function (QBF) of the robot's state and the motor input under a predictive horizon \bar{N} and a control horizon \bar{N}_u can be determined by the following equation:

$$\begin{aligned} L_D(k) &= \sum_{k=1}^{\bar{N}} q^T(k+j|j) \bar{Q}_D q(k+j|j) \\ &+ \sum_{k=1}^{\bar{N}_u} \tau^T(k+j-2|j) \bar{R}_D \tau(k+j-2|j), \end{aligned} \quad (56)$$

where \bar{Q}_D and \bar{R}_D are appropriate weighting matrices.

Hence, the dynamics predictive motor torque can be obtained by

$$\tau^* = \arg \min_{\tau} L_D(j) \quad (57)$$

subject to

$$\begin{aligned} \tau_{\min} &\leq \tau(k+j|j) \leq \tau_{\max}, \quad k \in [0, \bar{N}-2] \\ \ddot{q}_{\min} &\leq \ddot{q}(k+j|j) \leq \ddot{q}_{\max}, \quad k \in [1, \bar{N}]. \end{aligned} \quad (58)$$

6. Experimental Part

In this section, we design a 100 m \times 100 m workplace within 6 obstacles (O_i , $i = 1, 2, 3, 4, 5, 6$). We define the velocity of the OMR as 1 m/s and the start point of the OMR as (3, 90) while

the destination is (70, 28). We define the prediction horizon $N = 3$ and the control horizon $N_u = 2$. As the inputs of kinematic controller and dynamics controller are different, we set up a different constraint. The save distance between the OMR and obstacle is adjustable and the accurate value of the OMR's model is based on the OMR we built physically. The simulation is carried out in MATLAB.

6.1. Simulation of Kinematics Controller. In this section, simulation is carried out based on kinematics model. The OMR is navigated by potential field method combining MPC. In this simulation, we assume the goal position is already known, and the start position and start velocity are defined in advance.

The process of this simulation is shown by the following flow chart in Figure 5, where O_i means the i th obstacle. First, we define the originate position of the mobile robot and the originate velocity. Then, we sent the goal position to the mobile robot. With the help of potential field method, mobile robot can draw the next status, which is used as the reference for the MPC. After the MPC process, the predictive status is sent to mobile robot for navigation and to the previous process for the next reference.

The area of workplace is within 100 m \times 100 m and there are some obstacles randomly distributed in it. With the help of potential field and MPC, the OMR can adjust its velocity and finally reach the goal which is shown by Figures 6 and 7.

According to Figures 6 and 7, we can see that the OMR successfully reaches the goal and can smoothly self-avoid the obstacle. The velocity of the OMR is shown by Figures 8, 9, and 10.

At the beginning of simulation, we define the originate velocity of the robot as 1. When the mobile robot starts moving, the velocity (v_t) is modified by MPC as v_t on the left of simulation.

Connecting Figures 6, 8, and 9, we can see that the moment the change of v_t is rapid is the moment that mobile robot encounters an obstacle and it needs to change its direction. Figure 10 is the velocities of three omnidirectional wheels; the composition of all three velocities is v_t .

According to potential field method, we can get the total force angle, and then we can determine η . The change of η is shown by Figure 11.

The result of potential field method in navigation is outstanding. The OMR can smoothly avoid obstacles and successfully guide itself to the goal. With supplement of MPC, the motion of mobile robot is constrained and the velocity becomes more realistic and flexible. The robustness of system is enhanced.

6.2. Simulation of Dynamics Controller. In this section, similar to the last section, we change kinematics model into dynamics model and simulate a similar process which OMR is navigated by potential field and MPC. In this simulation, we assume the goal position is already known, and the start position and start velocity are defined in advance too.

According Figures 12 and 13, it shows that, with the control of dynamics controller and navigation of potential field, the OMR achieves its goal and successfully avoids the

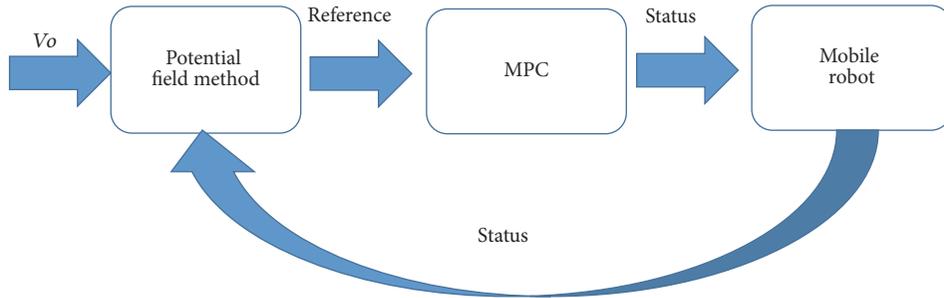


FIGURE 5: The velocity of the OMR during the simulation.

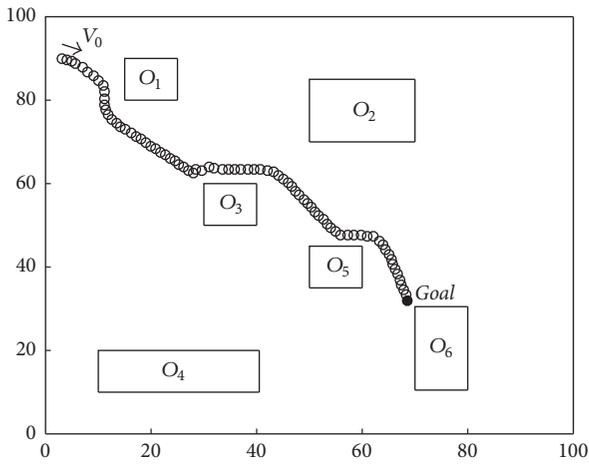


FIGURE 6: The simulation of OMR controlled by kinematic controller moving in the workplace.

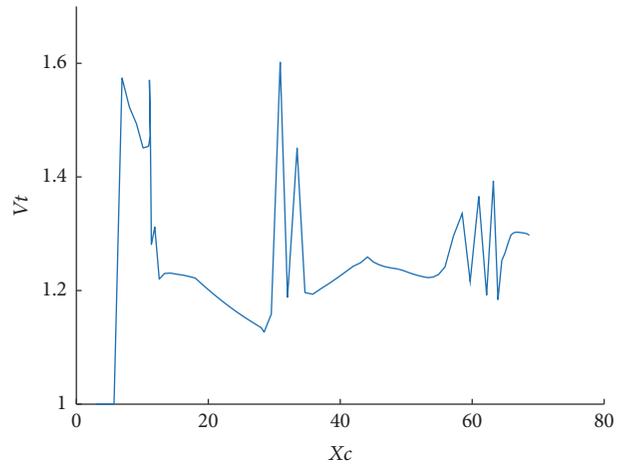


FIGURE 8: The velocity of the OMR controlled by kinematic controller during the simulation.

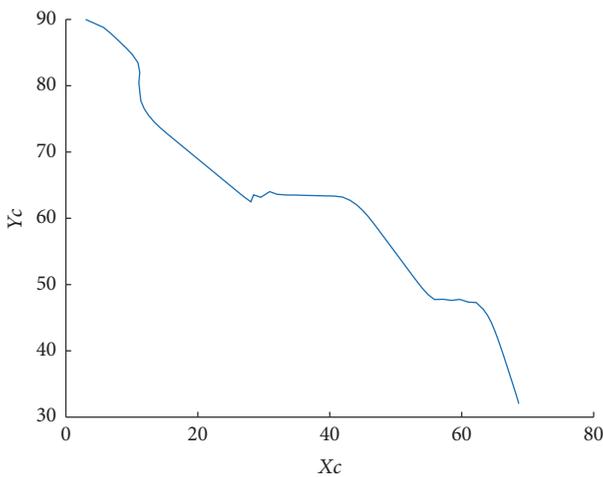


FIGURE 7: The trajectory of the OMR controlled by kinematic controller in $X_c O_c Y_c$ coordinate.

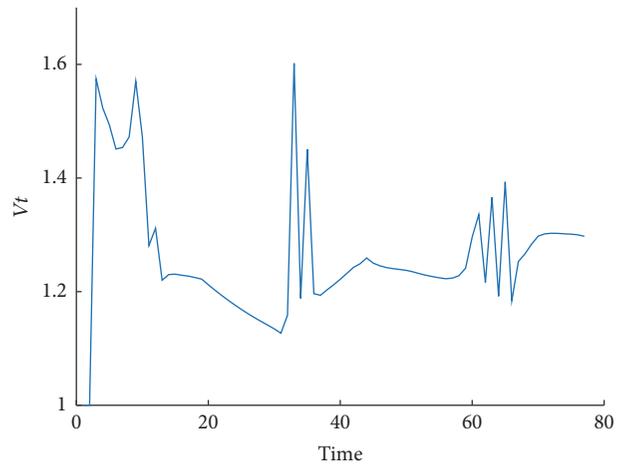


FIGURE 9: The change of velocity of the OMR controlled by kinematic controller.

obstacles. When the OMR approaches an obstacle, potential field algorithm generates a repulsive force which makes the OMR turn around and prevent itself from hitting the obstacle. With the help of MPC, the velocity is constrained, which can

prevent some value that is beyond the real produced by the potential field method from navigating the OMR.

Comparing Figures 14 and 15 to Figure 12, it is easy to see that when the velocity changes sharply is when the OMR is too close to the obstacle and it needs to slow down to avoid possible collision.

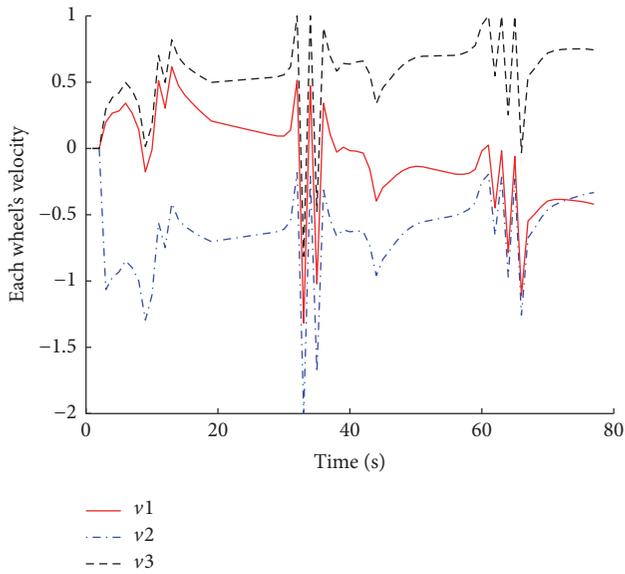


FIGURE 10: Three omnidirectional wheels' velocity.

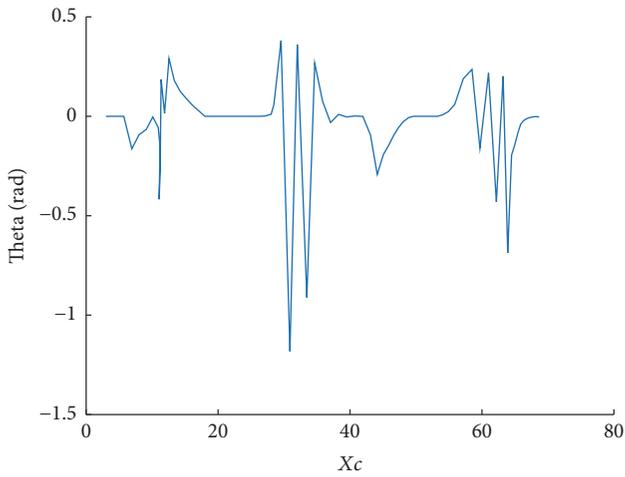


FIGURE 11: The angle with the change of robot's position is determined by potential field.

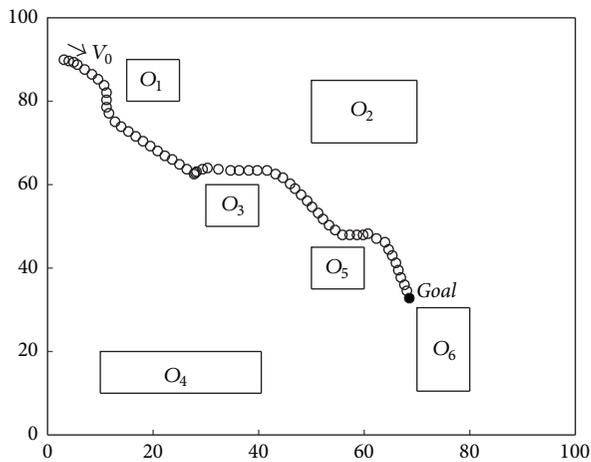


FIGURE 12: The simulation of the OMR controlled by dynamics controller moving in the workplace.

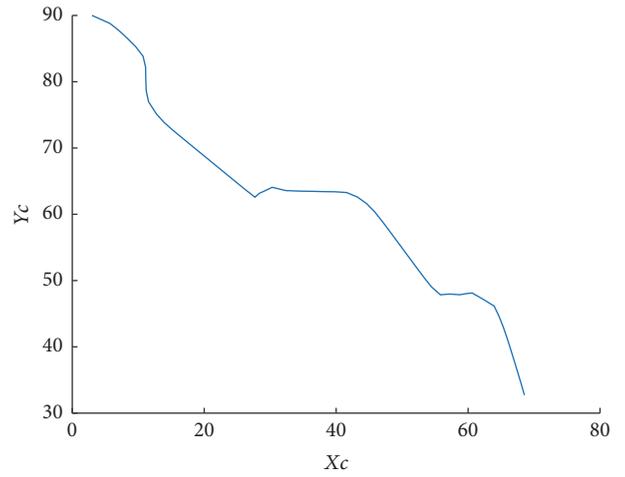


FIGURE 13: The trajectory of the OMR controlled by dynamics controller in $X_cO_eY_c$ coordinate.

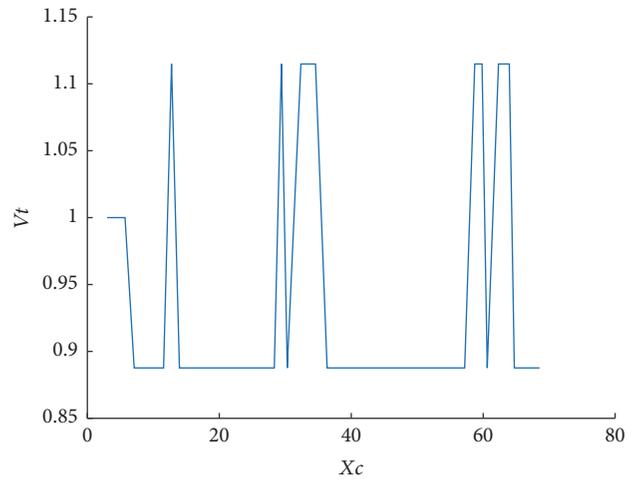


FIGURE 14: The velocity of the OMR controlled by dynamics controller during the simulation.

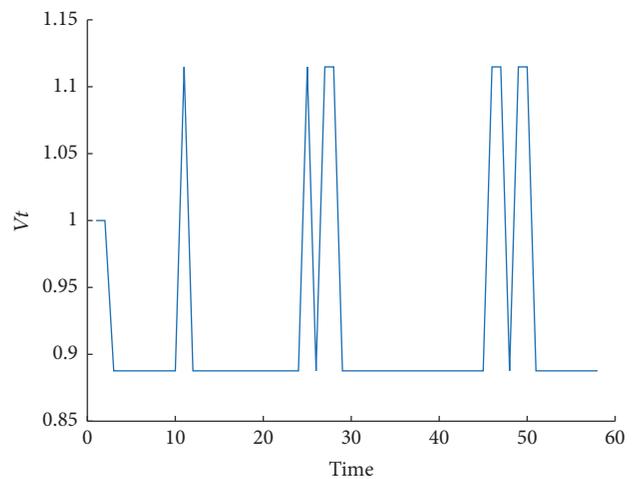


FIGURE 15: The change of velocity of the OMR controlled by dynamics controller.

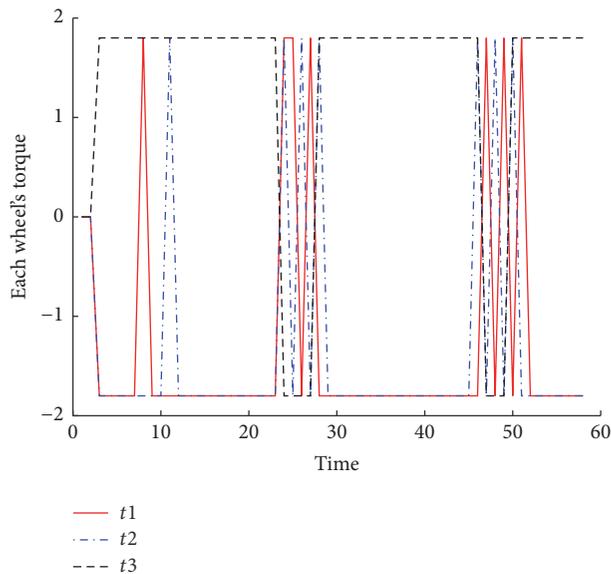


FIGURE 16: The change of torque of the OMR.

According to Figure 16, when the OMR needs to change its direction, it changes three torques, respectively, to generate a combined effort to steer its motion.

7. Conclusion

In this paper, a novel potential field method has been used to navigate a class of OMR. With the consideration of distance between the robot and the goal, robot, and obstacle, the GNRON problem is solved. In addition, it discusses the kinematic as well as dynamics model of mobile robot. For improving system's robustness, it combines potential field and MPC, so the motion planning is more complete. Finally, simulation results show that the proposed control scheme is more appropriate for omnidirectional mobile robot's navigation.

Disclosure

Chun-Yi Su is on leave from Concordia University, Canada.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was partially supported by National Nature Science Foundation (NSFC) under Grant 61473120, Guangdong Provincial Natural Science Foundation 2014A030313266 and International Science and Technology Collaboration Grant 2015A050502017, Science and Technology Planning Project of Guangzhou 201607010006, and State Key Laboratory of Robotics and System (HIT) Grant SKLRS-2017-KF-13.

References

- [1] R. L. Williams II, B. E. Carter, P. Gallina, and G. Rosati, "Dynamic model with slip for wheeled omnidirectional robots," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, pp. 285–293, 2002.
- [2] J. E. Mohd Salih, M. Rizon, S. Yaacob, A. H. Adom, and M. R. Mamat, "Designing omni-directional mobile robot with mecanum wheel," *American Journal of Applied Sciences*, vol. 3, no. 5, pp. 1831–1835, 2006.
- [3] Y. Liu, X. Wu, J. J. Zhu, and J. Lew, "Omni-directional mobile robot controller design by trajectory linearization," *Dynamics*, vol. 2, pp. 2–2, 2003.
- [4] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics," in *Proceedings of the 34th IEEE Conference on Decision and Control*, pp. 3805–3810, IEEE, December 1995.
- [5] C. Canudas de Wit and O. J. Sordalen, "Exponential stabilization of mobile robots with nonholonomic constraints," *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1791–1797, 1992.
- [6] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 53–64, 2003.
- [7] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 620–629, 2016.
- [8] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 3, pp. 334–344, 2016.
- [9] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT*," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2016.
- [10] M. C. Lee and M. G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '03)*, pp. 735–740, July 2003.
- [11] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [12] B. K. Choi, D. H. Kim, and R. B. Jerard, "C-space approach to tool-path generation for die and mould machining," *CAD Computer Aided Design*, vol. 29, no. 9, pp. 657–669, 1997.
- [13] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: the field D* algorithm," *Journal of Field Robotics*, vol. 23, no. 2, pp. 79–101, 2006.
- [14] A. M. C. Smith, C. Yang, H. Ma, P. Culverhouse, A. Cangelosi, and E. Burdet, "Novel hybrid adaptive controller for manipulation in complex perturbation environments," *PLoS ONE*, vol. 10, no. 6, Article ID e0129281, 2015.
- [15] Y. Jiang, C. Yang, and H. Ma, "A review of fuzzy logic and neural network based intelligent control design for discrete-time systems," *Discrete Dynamics in Nature and Society*, vol. 2016, Article ID 7217364, 11 pages, 2016.
- [16] C. Yang, K. Huang, H. Cheng, Y. Li, and C.-Y. Su, "Haptic identification by ELM controlled uncertain manipulator," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, In press.

- [17] C. Yang, Z. Li, and J. Li, "Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models," *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 24–36, 2013.
- [18] C. Yang, Y. Jiang, Z. Li, W. He, and C.-Y. Su, "Neural control of bimanual robots with guaranteed global stability and motion precision," *IEEE Transactions on Industrial Informatics*, 2017.
- [19] C. Yang, X. Wang, Z. Li, Y. Li, and C. Su, "Teleoperation control based on combination of wave variable and neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.
- [20] C. Yang, X. Wang, L. Cheng, and H. Ma, "Neural-learning-based telerobot control with guaranteed performance," *IEEE Transactions on Cybernetics*, 2017.
- [21] L. Zhang, Z. Li, and C. Yang, "Adaptive neural network based variable stiffness control of uncertain robotic systems using disturbance observer," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2236–2245, 2017.
- [22] Z. Li, S. Zhao, J. Duan, C. Su, C. Yang, and X. Zhao, "Human cooperative wheelchair with brain machine interaction based on shared control strategy," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 185–195, 2017.
- [23] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.
- [24] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1398–1404, Sacramento, Calif, USA, April 1991.
- [25] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, MIT Press, 2011.
- [26] C.-C. Tsai, F.-C. Tai, and Y.-R. Lee, "Motion controller design and embedded realization for Mecanum wheeled omnidirectional robots," in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA '11)*, pp. 546–551, Taipei, Taiwan, June 2011.
- [27] R. Cui, X. Zhang, and D. Cui, "Adaptive sliding-mode attitude control for autonomous underwater vehicles with input nonlinearities," *Ocean Engineering*, vol. 123, pp. 45–54, 2016.
- [28] R. Cui, J. Guo, and Z. Mao, "Adaptive backstepping control of wheeled inverted pendulums models," *Nonlinear Dynamics*, vol. 79, no. 1, pp. 501–511, 2015.
- [29] C. S. Chin and S. H. Lum, "Rapid modeling and control systems prototyping of a marine robotic vehicle with model uncertainties using xPC Target system," *Ocean Engineering*, vol. 38, no. 17–18, pp. 2128–2141, 2011.
- [30] C. S. Chin, M. W. Lau, E. Low, and G. G. Seet, "A robust controller design method and stability analysis of an underactuated underwater vehicle," *International Journal of Applied Mathematics and Computer Science*, vol. 16, no. 3, pp. 345–356, 2006.
- [31] C. S. Chin, M. W. S. Lau, and E. Low, "Supervisory cascaded controller design: experiment test on a remotely operated vehicle," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 225, no. 3, pp. 584–603, 2011.
- [32] Z. Li, H. Xiao, C. Yang, and Y. Zhao, "Model predictive control of nonholonomic chained systems using general projection neural networks optimization," *IEEE Transactions on Systems Man and Cybernetics: Systems*, vol. 45, no. 10, pp. 1313–1321, 2015.
- [33] A. S. K. Annamalai, R. Sutton, C. Yang, P. Culverhouse, and S. Sharma, "Robust adaptive control of an uninhabited surface vehicle," *Journal of Intelligent and Robotic Systems*, vol. 78, no. 2, pp. 319–338, 2015.
- [34] H. Xiao, Z. Li, C. Yang et al., "Robust stabilization of a wheeled mobile robot using model predictive control based on neurodynamics optimization," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 505–516, 2017.
- [35] Z. Li, C. Yang, C.-Y. Su, J. Deng, and W. Zhang, "Vision-based model predictive control for steering of a nonholonomic mobile robot," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 553–564, 2016.
- [36] G. Campion, G. Bastin, and B. D'Andréa-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics & Automation*, vol. 12, no. 1, pp. 47–62, 1996.
- [37] D. Xu, D. Zhao, J. Yi, and X. Tan, "Trajectory tracking control of omnidirectional wheeled mobile manipulators: robust neural network-based sliding mode approach," *IEEE Transactions on Systems, Man & Cybernetics, Part B: Cybernetics*, vol. 39, no. 3, pp. 788–799, 2009.
- [38] K. Watanabe, "Control of an omnidirectional mobile robot," in *Proceedings of the 2nd International Conference on Knowledge-Based Intelligent Electronic Systems (KES '98)*, vol. 3, pp. 51–60, Adelaide, Australia, April 1998.
- [39] Z. Li and C. Yang, "Neural-adaptive output feedback control of a class of transportation vehicles based on wheeled inverted pendulum models," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 6, pp. 1583–1591, 2012.
- [40] J. M. Maciejowski, *Predictive Control with Constraints*, Prentice-Hall, 2002.
- [41] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*, Springer, London, UK, 2009.
- [42] M. Li, Y. Li, S. S. Ge, and T. H. Lee, "Adaptive control of robotic manipulators with unified motion constraints," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 184–194, 2017.