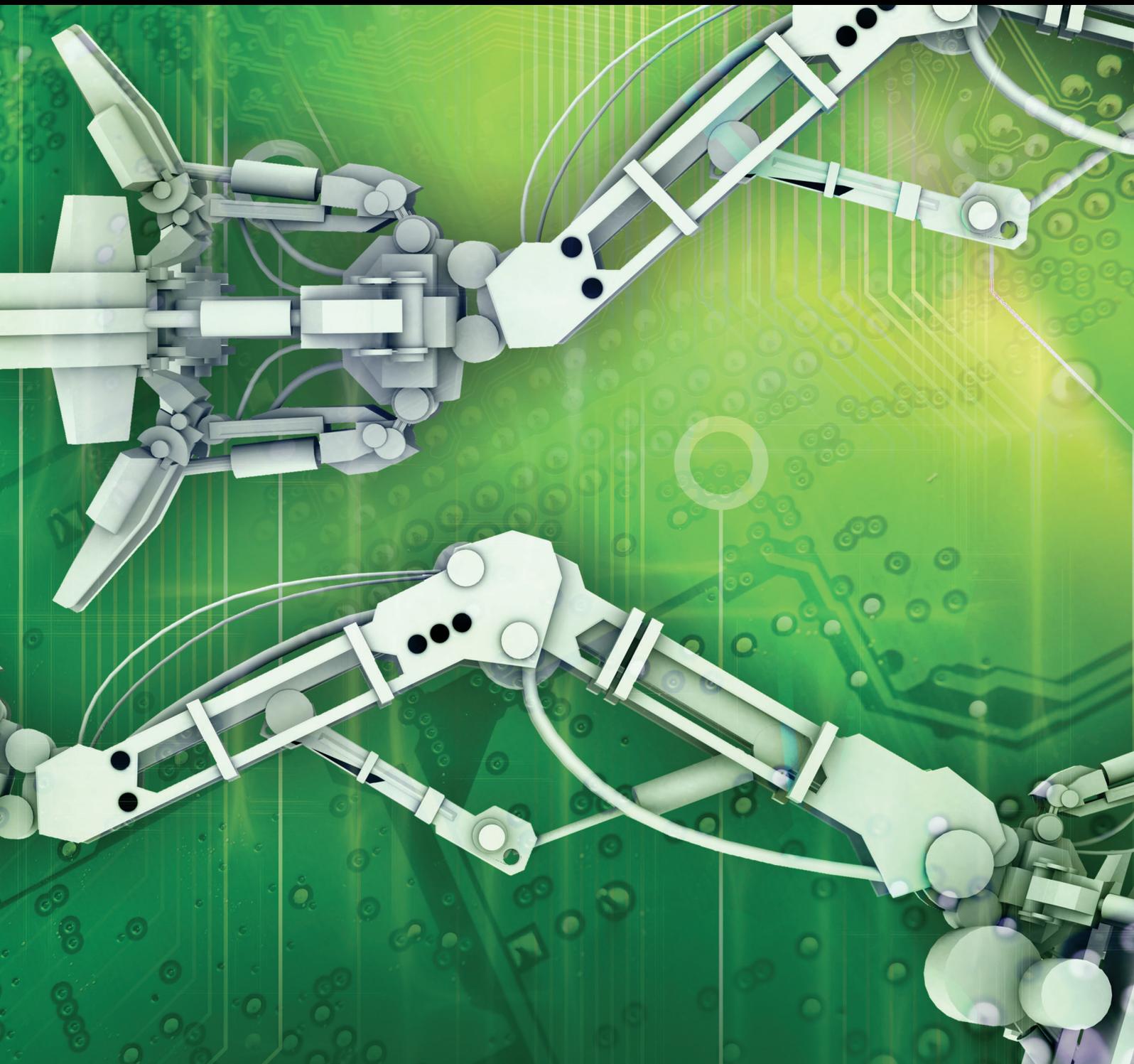


Biologically Inspired Robotics

Guest Editors: Liwei Shi, Maki K. Habib, Nan Xiao, and Huosheng Hu





Biologically Inspired Robotics

Journal of Robotics

Biologically Inspired Robotics

Guest Editors: Liwei Shi, Maki K. Habib, Nan Xiao,
and Huosheng Hu



Copyright © 2015 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in "Journal of Robotics." 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

Paolo Arena, Italy
Suguru Arimoto, Japan
Jorge Dias, Portugal
Meng J. Er, Singapore
L. Fortuna, Italy
Andrew A. Goldenberg, Canada
Madan M. Gupta, Canada
Huosheng Hu, UK
Farrokh Janabi-Sharifi, Canada
Seul Jung, Republic of Korea

Yangmin Li, Macau
Lyle N. Long, USA
Sauro Longhi, Italy
Zhiwei Luo, Japan
Shugen Ma, Japan
Fumitoshi Matsuno, Japan
Rene V. Mayorga, Canada
Ali Meghdari, Iran
R. C. Michelini, Italy
Giovanni Muscato, Italy

Shahram Payandeh, Canada
Gordon R. Pennock, USA
Lakmal Seneviratne, UK
Tarek M. Sobh, USA
T. Tarn, USA
Toshio Tsuji, Japan
Keigo Watanabe, Japan
Simon X. Yang, Canada
Yuan F. Zheng, USA

Contents

Biologically Inspired Robotics, Liwei Shi, Maki K. Habib, Nan Xiao, and Huosheng Hu
Volume 2015, Article ID 894394, 2 pages

Rolling Locomotion Control of a Biologically Inspired Quadruped Robot Based on Energy Compensation, Takuma Nemoto, Rajesh Elara Mohan, and Masami Iwase
Volume 2015, Article ID 649819, 10 pages

Development and Motion Testing of a Robotic Ray, Jianhui He and Yonghua Zhang
Volume 2015, Article ID 791865, 13 pages

Action Selection and Operant Conditioning: A Neurobotic Implementation,
André Cyr and Frédéric Thériault
Volume 2015, Article ID 643869, 10 pages

Bioinspired Tracking Control of High Speed Nonholonomic Ground Vehicles,
Adam Shoemaker and Alexander Leonessa
Volume 2015, Article ID 919073, 15 pages

Unified Switching between Active Flying and Perching of a Bioinspired Robot Using Impedance Control, Shanshan Du, Heping Chen, Yong Liu, and Runtong Hu
Volume 2015, Article ID 763710, 11 pages

Development and Evaluation of Compact Robot Imitating a Hermit Crab for Inspecting the Outer Surface of Pipes, Naoto Imajo, Yogo Takada, and Mikiji Kashinoki
Volume 2015, Article ID 312780, 7 pages

Sparse Approximation for Nonrigid Structure from Motion, Yaming Wang, Xiaomeng Yan, Junbao Zheng, and Mingfeng Jiang
Volume 2015, Article ID 435385, 8 pages

Unmanned Aerial Vehicle Navigation Using Wide-Field Optical Flow and Inertial Sensors,
Matthew B. Rhudy, Yu Gu, Haiyang Chao, and Jason N. Gross
Volume 2015, Article ID 251379, 12 pages

Inverse Kinematic Analysis and Evaluation of a Robot for Nondestructive Testing Application,
Zongxing Lu, Chunguang Xu, Qinxue Pan, Xinyu Zhao, and Xinliang Li
Volume 2015, Article ID 596327, 7 pages

Simulation of Octopus Arm Based on Coupled CPGs, Juan Tian and Qiang Lu
Volume 2015, Article ID 529380, 9 pages

Editorial

Biologically Inspired Robotics

Liwei Shi,¹ Maki K. Habib,² Nan Xiao,^{1,3} and Huosheng Hu⁴

¹*The Institute of Advanced Biomedical Engineering System, School of Life Science, Beijing Institute of Technology, No. 5, Zhongguancun South Street, Haidian District, Beijing 100081, China*

²*Mechanical Engineering Department, School of Sciences and Engineering, The American University in Cairo, Cairo 11511, Egypt*

³*Faculty of Engineering, Kagawa University, 2217-20 Hayashi-cho, Takamatsu 761-0396, Japan*

⁴*School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK*

Correspondence should be addressed to Liwei Shi; shiliwei@bit.edu.cn

Received 13 September 2015; Accepted 14 September 2015

Copyright © 2015 Liwei Shi 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.

Biologically inspired robotics has enabled today's robots to operate in a variety of unstructured and dynamically changing environments in addition to traditional structured environments. As a result, intelligent robots will soon be ready to serve in our home, hospital, office, and outdoors. It is clear that bioinspired methods are becoming increasingly important in the face of the complexity of today's demanding applications. Biological inspiration in robotics is leading to complex structures with sensory-motor coordination, in which learning often plays an important role in achieving adaptation. This special issue is focused on the theoretical and technological challenges of evolutionary transformation from biological systems to intelligent robots.

The paper "Development and Evaluation of Compact Robot Imitating a Hermit Crab for Inspecting the Outer Surface of Pipes" by N. Imajo et al. concentrates on an outer-pipe inspection robot. The robot's locomotion mechanism is developed to imitate a terrestrial hermit crab's claws. It is equipped with two rimless wheels. Each of the spokes is tipped with a neodymium magnet, which allows the robot to remain attached to even a vertical steel pipe. Moreover, the robot has a mechanism for adjusting the camber angle of the right and left wheels, allowing it to tightly grip pipes with different diameters.

The paper "Bioinspired Tracking Control of High Speed Nonholonomic Ground Vehicles" by A. Shoemaker and A. Leonessa presents a control algorithm based on a unicycle model. This algorithm focuses on driving the current vehicle to within some variable following distance of a user defined reference system by commanding longitudinal and angular

rates. This algorithm is proven stable through Lyapunov criteria. The results show that the algorithm demonstrates significant promise in high speed trajectory tracking with obstacle avoidance.

The paper "Unified Switching between Flying and Perching of a Bioinspired Robot Using Impedance Control" by S. Du et al. introduces a unified impedance control approach for bioinspired flying and perching robots. The impedance force control method is developed to enable the robot to approach a desired position and handle the contact force with the environment smoothly until the adsorption device perches on the object surface steadily.

The paper "Rolling Locomotion Control of a Biologically Inspired Quadruped Robot Based on Energy Compensation" by T. Nemoto et al. proposes an approach to control rolling locomotion with the quadruped robot. A dynamic model of the quadruped robot with rolling locomotion is developed by applying a constraint force due to collision and contact with the ground to a base quadruped robot model. The proposed control approach is effective in achieving periodic rolling locomotion.

The paper "Development and Motion Testing of a Robotic Ray" by J. He and Y. Zhang develops a robotic ray based on the simplified pectoral structure of living bluespotted ray. The mechanical structure and control circuit are presented, and kinematic analysis on the fin ray and the full fin is discussed. The average propulsion velocity generally increases with the increase of kinematic parameters such as frequency, amplitude, and wavelength, respectively.

The paper “Unmanned Aerial Vehicle Navigation Using Wide-Field Optical Flow and Inertial Sensors” by M. B. Rhudy et al. investigates vision-aided inertial navigation techniques which do not rely upon GPS using UAV (unmanned aerial vehicle) flight data. Two different formulations are presented, a full state estimation formulation which captures the aircraft ground velocity vector and attitude and a simplified formulation which assumes that all of the aircraft velocity is in the forward direction. In addition, a state is included in each formulation in order to estimate the distance between the image center and the aircraft.

The paper “Sparse Approximation for Nonrigid Structure from Motion” by Y. Wang et al. proposes a novel sparse approximation method to resolve the nonrigid structure from motion problem in trajectory space. Instead of generating a truncated traditional trajectory basis, this method uses an atom dictionary which includes a set of overcomplete bases to estimate the real shape of the deformable object.

The paper “Simulation of Octopus Arm Based on Coupled CPGs” by J. Tian and Q. Lu presents three coupled central pattern generators (CPGs) and a 2-dimensional dynamic model of the octopus arm to explore possible strategies of the octopus movement control. The CPGs’ signals treated as activation are added on the ventral, dorsal, and transversal sides, respectively. The effects of the octopus arm are discussed when the parameters of the CPGs are changed. The results are beneficial for researchers to understand the octopus movement further.

The paper “Action Selection and Operant Conditioning: A Neurobotic Implementation” by A. Cyr and F. Thériault shows an AS (action selection) process made from simple cellular elements. It is based on a CPG and sensory neurons which influence decision neurons in their choice to generate a behavior from the action neurons. The experiments are conducted under a biologically inspired paradigm, specifically with a SNN (spiking neural network) acting as brain controller for virtual and physical robots. The simplicity and the generic aspect of above AS model may provide a fast track solution to build more complex SNN, including multiple actions in different dynamic scenarios.

The paper “Inverse Kinematic Analysis and Evaluation of a Robot for Nondestructive Testing Application” by Z. Lu et al. provides a simple and effective method for the inverse kinematics problem of an industrial robot system. The arm-wrist separateness method is adopted to solve the inverse kinematics for robots with 6 DOF. The proposed method will greatly benefit the development of nondestructive industrial testing.

Liwei Shi
Maki K. Habib
Nan Xiao
Huosheng Hu

Research Article

Rolling Locomotion Control of a Biologically Inspired Quadruped Robot Based on Energy Compensation

Takuma Nemoto,¹ Rajesh Elara Mohan,² and Masami Iwase¹

¹Department of Robotics and Mechatronics, Tokyo Denki University, 5 Senjuasahi-cho, Adachi-ku, Tokyo 120-8551, Japan

²Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372

Correspondence should be addressed to Takuma Nemoto; nemoto@ctrl.fr.dendai.ac.jp

Received 27 March 2015; Accepted 12 July 2015

Academic Editor: Huosheng Hu

Copyright © 2015 Takuma Nemoto 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 have developed a biologically inspired reconfigurable quadruped robot which can perform walking and rolling locomotion and transform between walking and rolling by reconfiguring its legs. This paper presents an approach to control rolling locomotion with the biologically inspired quadruped robot. For controlling rolling locomotion, a controller which can compensate robot's energy loss during rolling locomotion is designed based on a dynamic model of the quadruped robot. The dynamic model describes planar rolling locomotion based on an assumption that the quadruped robot does not fall down while rolling and the influences of collision and contact with the ground, and it is applied for computing the mechanical energy and a plant in a numerical simulation. The numerical simulation of rolling locomotion on the flat ground verifies the effectiveness of the proposed controller. The simulation results show that the quadruped robot can perform periodic rolling locomotion with the proposed energy-based controller. In conclusion, it is shown that the proposed control approach is effective in achieving the periodic rolling locomotion on the flat ground.

1. Introduction

Reconfigurable robots have the capability to adapt to different tasks and environments. Various designs of reconfigurable robots have been investigated, for instance, modular robots [1–3], cooperative robots [4–6], and transformable multi-legged or parallel robots [7–12].

Designs of reconfigurable robots can be inspired by creatures in nature [13–16]. Reconfigurable robots inspired by a creature performing walking and rolling locomotion particularly provide the capability to attain the fast and energy-efficient movement on the flat ground with rolling locomotion and high stability and mobility on the uneven ground with walking locomotion. Such reconfigurable robots have been discussed in literatures [17, 18]. Lin et al. have focused on a caterpillar that can escape rapidly from predators by reconfiguring its body structure like a wheel and have developed a caterpillar-inspired soft robot, which has attempted rolling locomotion [17]. King has focused on somersault rolling locomotion performed by a spider called

“huntsman spider (*Cebrennus villosus*)” and has developed a quadruped robot capable of somersaulting, which has performed somersault rolling locomotion [18].

We have optimized the design mechanism presented in [18] and have developed reconfigurable robotic platforms which can perform walking and rolling locomotion and transform between walking and rolling by reconfiguring their legs. For the platforms, we have built a feedforward controller and a terrain perception system in our previous work [19]. Applying the feedforward controller and the terrain perception system has provided walking and rolling locomotion on the platforms; however, we have not achieved stable and periodic rolling locomotion.

In this paper, we discuss periodic rolling locomotion control of one of our platforms called the huntsman-spider-inspired quadruped robot. For periodic rolling locomotion, we focus on robot's energy during rolling locomotion, though previous studies of rolling locomotion on reconfigurable robots [8, 11, 20, 21] have focused on the movement of robot's center of gravity (COG). We show that the quadruped

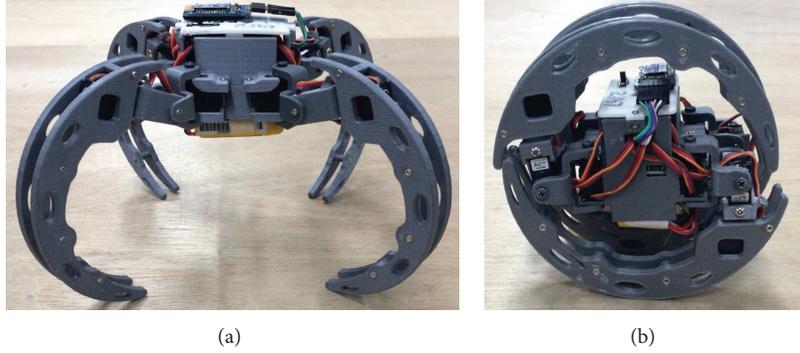


FIGURE 1: Huntsman-spider-inspired quadruped robot: (a) walking form and (b) rolling form. This robot has four legs and three motors on each leg, namely, twelve motors. Three motors on each leg are mounted as their axes are at right angles to each other.

robot performs periodic rolling locomotion with energy-based control.

The quadruped robot loses some of robot's energy due to collision and contact with the ground while rolling, and therefore it cannot perform periodic rolling locomotion without energy supply. It hence requires energy compensation, which is provided by applying control methods based on object's energy discussed in literatures [22–26].

In this paper, a controller which can compensate the robot's energy loss during rolling locomotion is designed based on a dynamic model of the quadruped robot. Its dynamic model describes planar rolling locomotion based on an assumption that it does not fall down while rolling and the influences of collision and contact with the ground. Applying the dynamic model, the controller computes robot's energy and input for energy compensation. The effectiveness of the proposed controller is verified through a numerical simulation of its rolling locomotion.

2. Modeling the Huntsman-Spider-Inspired Quadruped Robot with Rolling Locomotion

This section presents the dynamic model of the huntsman-spider-inspired quadruped robot with rolling locomotion.

Figure 1 shows the developed platform. This robot has four legs and three motors on each leg, namely, twelve motors. Three motors on each leg are mounted as their axes are at right angles to each other.

The robot has the walking form and rolling one shown in Figure 1 and can transform between walking and rolling by reconfiguring their legs as shown in Figure 2. By utilizing these forms, it can adapt to different tasks and environments.

The robot can roll on the flat ground by moving the front legs or the rear legs simultaneously as shown in Figure 3, which shows rolling locomotion with feedforward control. In this paper, the rolling locomotion is modeled.

The quadruped robot model is developed based on the following assumptions.

Assumption 1. The quadruped robot can swing the legs while rolling.

TABLE 1: Physical parameters of the rolling huntsman-spider-inspired quadruped robot.

Parameter	Character	Value
Mass of body (kg)	m_b	1.70×10^{-1}
Mass of leg (kg)	m_l	5.50×10^{-2}
Inertia moment of body (kgm^2)	I_b	7.65×10^{-5}
Inertia moment of leg (kgm^2)	I_l	6.10×10^{-6}
Viscosity of body (Nms/rad)	c_b	1.70×10^{-9}
Viscosity of leg (Nms/rad)	c_l	5.50×10^{-10}
Length from COG of body to joint (m)	l_b	5.50×10^{-2}
Length from joints to COG of leg (m)	l_{gl}	5.83×10^{-2}
Angle from leg to COG of leg (rad)	α_{gl}	5.40×10^{-1}
Radius of ground contact area (m)	r_l	8.00×10^{-2}
Gravity acceleration (m/s^2)	g	9.81

TABLE 2: Variables of the rolling huntsman-spider-inspired quadruped robot ($j = 1, 2$).

COG coordinates of body (m)	(x_b, y_b)
COG coordinates of leg j (m)	(x_{lj}, y_{lj})
Rotational angle of body (rad)	θ_b
Rotational angle of leg j (rad)	θ_{lj}
Joint torque of leg j (Nm)	τ_{lj}

Assumption 2. The quadruped robot does not fall down while rolling.

Assumption 3. The quadruped robot rolls across the high frictional flat ground without slipping.

Assumption 4. Collision with the ground is assumed as completely inelastic in nature.

According to the assumptions, the model diagram of the rolling quadruped robot is shown in Figure 4, the physical parameters are shown in Table 1, and the variables are shown in Table 2. The subscripts b and lj denote the body and the legs, respectively, and $j = 1, 2$ denotes number of

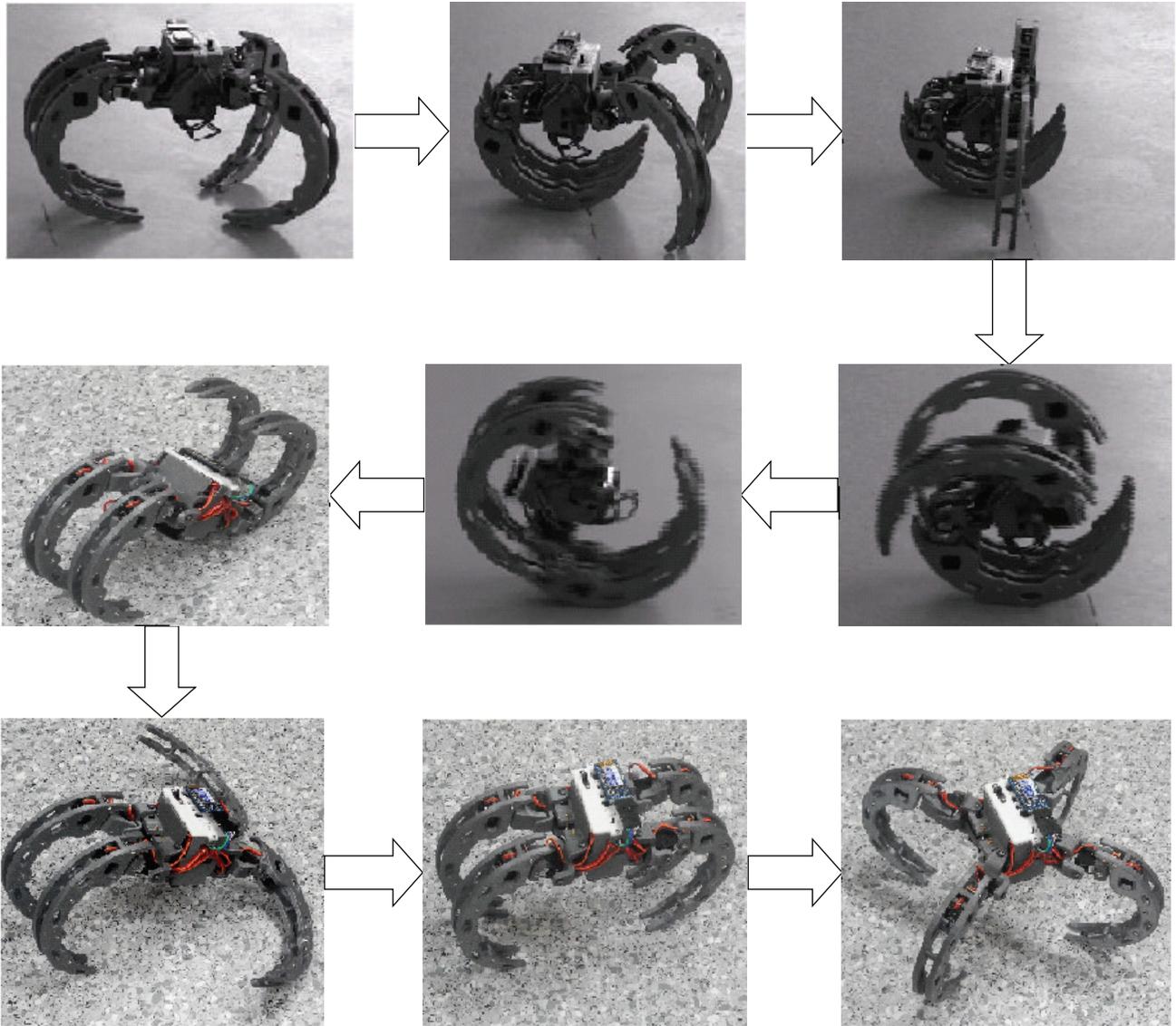


FIGURE 2: Transformation between walking and rolling. The quadruped robot has the walking form and rolling one and can transform between walking and rolling by reconfiguring their legs.

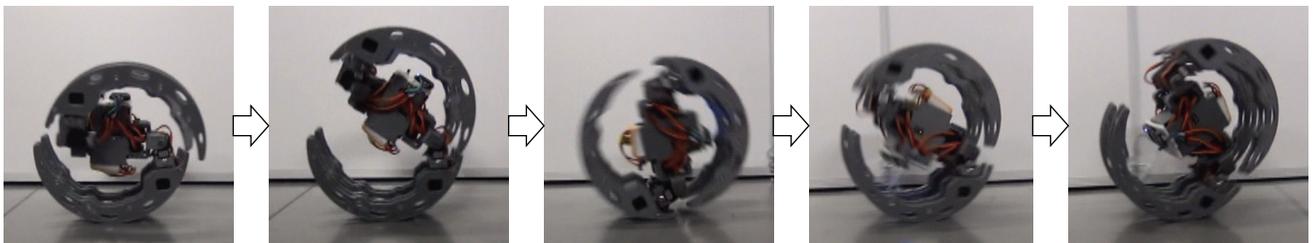


FIGURE 3: Rolling locomotion with feedforward control. The quadruped robot can roll on the flat ground by moving the front legs or the rear legs simultaneously.

the legs. The rotational angles of the legs are a relative angle against the body. The model diagram describes planar rolling locomotion on the quadruped robot in the vertical two-dimensional surface, and its X -axis describes the flat ground.

A motion equation of the rolling quadruped robot model is derived by applying the projection method [27–29]. To derive it, the projection method yields a base motion equation of the quadruped robot, which does not contain the influences of collision and contact with the ground on the

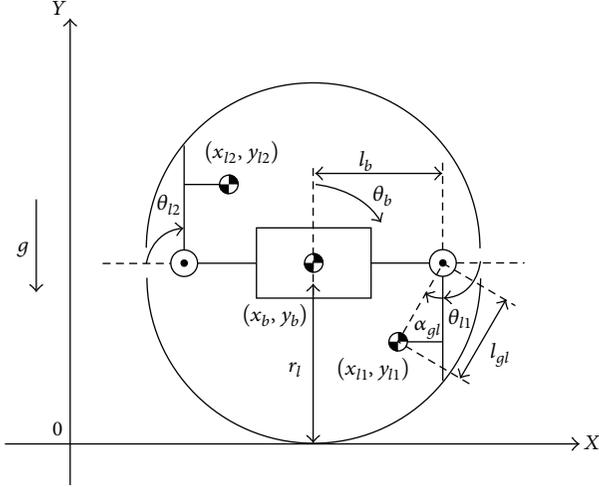


FIGURE 4: Model diagram of the rolling huntsman-spider-inspired quadruped robot. The model diagram describes planar rolling locomotion on the quadruped robot in the vertical two-dimensional surface, and its X -axis describes the flat ground.

robot, from independent motion equations of components constituting the robot and constraint conditions between each component. It is then obtained by giving a constraint force due to collision and contact with the ground, which is given by constraint conditions posed to the quadruped robot while the collision or contact occurs, to the base motion equation [30, 31]. The constraint force also gives discontinuous changes of velocity due to collision according to Assumption 4.

2.1. The Base Model of the Huntsman-Spider-Inspired Quadruped Robot. An unconstrained motion equation constituted of the motion equations of the independent components is written to derive the base motion equation of the quadruped robot.

Generalized coordinates \mathbf{x}_q are defined as

$$\mathbf{x}_q = [\theta_b, \theta_{l1}, \theta_{l2}, x_b, y_b, x_{l1}, y_{l1}, x_{l2}, y_{l2}]^T. \quad (1)$$

A generalized mass matrix \mathbf{M}_q and a generalized force vector \mathbf{h}_q are given by

$$\begin{aligned} \mathbf{M}_q &= \text{diag}(\mathbf{M}_{11}, \mathbf{M}_{22}), \\ \mathbf{M}_{11} &:= \begin{bmatrix} I_b + 2I_l & I_l & I_l \\ I_l & I_l & 0 \\ I_l & 0 & I_l \end{bmatrix}, \\ \mathbf{M}_{22} &:= \text{diag}(m_b, m_b, m_l, m_l, m_l, m_l), \end{aligned} \quad (2)$$

$$\mathbf{h}_q = [-c_b \dot{\theta}_b, \tau_{l1} - c_l \dot{\theta}_{l1}, \tau_{l2} - c_l \dot{\theta}_{l2}, 0, -m_b g, 0, -m_l g, 0, -m_l g]^T.$$

The unconstrained motion equation is represented by $\mathbf{M}_q \ddot{\mathbf{x}}_q = \mathbf{h}_q$.

The projection method leads to a constrained motion equation by considering conditions to constrain system behavior including definitions of positional relationships between each component. The constraint conditions of the quadruped robot are definitions of positional relationships between the body and each leg. They are given by

$$\begin{aligned} x_{l1} &= x_b + l_b \cos \theta_b + l_{gl} \cos(\theta_b + \theta_{l1} + \alpha_{gl}), \\ y_{l1} &= y_b - l_b \sin \theta_b - l_{gl} \sin(\theta_b + \theta_{l1} + \alpha_{gl}), \\ x_{l2} &= x_b - l_b \cos \theta_b - l_{gl} \cos(\theta_b + \theta_{l2} + \alpha_{gl}), \\ y_{l2} &= y_b + l_b \sin \theta_b + l_{gl} \sin(\theta_b + \theta_{l2} + \alpha_{gl}). \end{aligned} \quad (3)$$

The constraint conditions give a constraint matrix \mathbf{C}_q which should satisfy $\mathbf{C}_q \dot{\mathbf{x}}_q = \mathbf{0}$. They are combined into a constraint equation $\Phi_q = \mathbf{0}$ by moving the right member of each equation in (3) to the other side. The constraint matrix \mathbf{C}_q is thus represented by

$$\mathbf{C}_q = \frac{\partial \Phi_q}{\partial \mathbf{x}_q}. \quad (4)$$

Applying the constraint matrix \mathbf{C}_q and Lagrange's undetermined multipliers λ_q yields a constrained system

$$\mathbf{M}_q \ddot{\mathbf{x}}_q = \mathbf{h}_q + \mathbf{C}_q^T \lambda_q. \quad (5)$$

Since (5) has redundant degrees of freedom, they are reduced.

An independent velocity vector under constrained state $\dot{\mathbf{q}}_q$ which is selected from $\dot{\mathbf{x}}_q$ is defined as

$$\dot{\mathbf{q}}_q = [\dot{\theta}_b, \dot{\theta}_{l1}, \dot{\theta}_{l2}, \dot{x}_b, \dot{y}_b]^T. \quad (6)$$

Since applying the independent velocity vector $\dot{\mathbf{q}}_q$ yields $\dot{\mathbf{x}}_q = [\dot{\mathbf{q}}_q, \mathbf{v}_q^T]^T$, we can represent the constraint matrix by $\mathbf{C}_q = [\mathbf{C}_{q1}, \mathbf{C}_{q2}]$ so that $\mathbf{C}_q \dot{\mathbf{x}}_q = \mathbf{C}_{q1} \dot{\mathbf{q}}_q + \mathbf{C}_{q2} \mathbf{v}_q$. From this relationship, an orthogonal matrix \mathbf{D}_q can be obtained so as to be $\mathbf{C}_q \mathbf{D}_q = \mathbf{0}$ and $\dot{\mathbf{x}}_q = \mathbf{D}_q \dot{\mathbf{q}}_q$. Since $\mathbf{C}_q \dot{\mathbf{x}}_q = \mathbf{C}_{q1} \dot{\mathbf{q}}_q + \mathbf{C}_{q2} \mathbf{v}_q = \mathbf{0}$ gives $\mathbf{v}_q = -\mathbf{C}_{q2}^{-1} \mathbf{C}_{q1} \dot{\mathbf{q}}_q$, the orthogonal matrix \mathbf{D}_q is obtained from $\dot{\mathbf{x}}_q = [\dot{\mathbf{q}}_q, \mathbf{v}_q^T]^T = \mathbf{D}_q \dot{\mathbf{q}}_q$ as

$$\mathbf{D}_q = \begin{bmatrix} \mathbf{I}_5 \\ -\mathbf{C}_{q2}^{-1} \mathbf{C}_{q1} \end{bmatrix}, \quad (7)$$

where \mathbf{I} denotes an identity matrix and an index of \mathbf{I} denotes a dimensions of an identity matrix. Besides (7) satisfies $\mathbf{C}_q \mathbf{D}_q = \mathbf{C}_{q1} - \mathbf{C}_{q2} \mathbf{C}_{q2}^{-1} \mathbf{C}_{q1} = \mathbf{0}$.

The constrained motion equation is derived by projecting the constrained system (5) on the space constrained by \mathbf{D}_q^T and transforming the coordinates of the component vectors. The base motion equation of the quadruped robot is thereby derived as

$$\mathbf{D}_q^T \mathbf{M}_q \mathbf{D}_q \ddot{\mathbf{q}}_q + \mathbf{D}_q^T \mathbf{M}_q \dot{\mathbf{D}}_q \dot{\mathbf{q}}_q = \mathbf{D}_q^T \mathbf{h}_q. \quad (8)$$

2.2. *Consideration of the Constraint Force due to Collision and Contact with the Ground.* The motion equation of the quadruped robot with rolling locomotion is derived by applying the constraint force due to collision and contact with the ground to the base motion equation of the quadruped robot.

Applying the constraint force due to collision and contact with the ground τ_I to the base motion equation (5) yields the motion equation of the quadruped robot with rolling locomotion

$$\mathbf{M}_q \ddot{\mathbf{x}}_q = \mathbf{h}_q + \mathbf{C}_q^T \boldsymbol{\lambda}_q + \boldsymbol{\tau}_I. \quad (9)$$

The constraint force $\boldsymbol{\tau}_I$ is represented by

$$\boldsymbol{\tau}_I = \mathbf{C}_I^T \boldsymbol{\lambda}_I, \quad (10)$$

where \mathbf{C}_I is the constraint matrix due to collision and contact with the ground, $\boldsymbol{\lambda}_I$ is the Lagrange's undetermined multipliers, and \mathbf{C}_I should satisfy $\mathbf{C}_I \dot{\mathbf{x}}_q = \mathbf{0}$.

When the height of a grounding point of the quadruped robot h_j is less than or equal to 0 ($h_j \leq 0$) and the ground reaction force λ_j is greater than 0 ($\lambda_j > 0$), consider the following constraint conditions:

- (1) grounding legs roll without slipping;
- (2) a height of a grounding leg does not change.

Here the coordinates of ground basing point (x_{cj}, y_{cj}) are given by

$$\begin{aligned} x_{c1} &= x_b + l_b \cos \theta_b + l_c \cos(\theta_b + \theta_{l1} + \alpha_c), \\ y_{c1} &= y_b - l_b \sin \theta_b - l_c \sin(\theta_b + \theta_{l1} + \alpha_c), \\ x_{c2} &= x_b - l_b \cos \theta_b - l_c \cos(\theta_b + \theta_{l2} + \alpha_c), \\ y_{c2} &= y_b + l_b \sin \theta_b + l_c \sin(\theta_b + \theta_{l2} + \alpha_c), \end{aligned} \quad (11)$$

and h_j can be represented by $h_j = y_{cj} - r_l$, where $l_c = 5.70 \times 10^{-2}$ m and $\alpha_c = 1.30$ rad are the length from the joint to the ground basing point and angle from the leg to the ground basing point, respectively. The following expressions are thus derived from the above:

$$\begin{aligned} x_{cj} &= x_{cj0} + r_l (\theta_b + \theta_{lj} - (\theta_{b0} + \theta_{lj0})), \\ y_{cj} &= r_l, \\ h_j &\leq 0 \cap \lambda_j > 0, \end{aligned} \quad (12)$$

where x_{cj0} , θ_{b0} , and θ_{lj0} are the x -coordinate of each leg and the angle of the body and the legs when constraints occur, respectively.

The constraint matrix \mathbf{C}_I is consequently represented by

$$\mathbf{C}_I = \frac{\partial \Phi_I}{\partial \mathbf{x}_q}, \quad h_j \leq 0 \cap \lambda_j > 0, \quad (13)$$

where the constraint equation $\Phi_I = \mathbf{0}$ is obtained from (12).

When (13) holds, projecting (9) on the space constrained by \mathbf{D}_q^T and transforming the coordinates of the component vectors can transform (9) into

$$\mathbf{D}_q^T \mathbf{M}_q \mathbf{D}_q \ddot{\mathbf{q}}_q + \mathbf{D}_q^T \mathbf{M}_q \dot{\mathbf{D}}_q \dot{\mathbf{q}}_q = \mathbf{D}_q^T \mathbf{h}_q + \mathbf{D}_q^T \mathbf{C}_I^T \boldsymbol{\lambda}_I. \quad (14)$$

Besides substituting (10) into (9) can also represent (9) as

$$\begin{aligned} \mathbf{M}_q \ddot{\mathbf{x}}_q &= \mathbf{h}_q + \mathbf{C}_{qq}^T \boldsymbol{\lambda}_{qq}, \\ \mathbf{C}_{qq} &:= [\mathbf{C}_q^T, \mathbf{C}_I^T]^T, \\ \boldsymbol{\lambda}_{qq} &:= [\boldsymbol{\lambda}_q^T, \boldsymbol{\lambda}_I^T]^T. \end{aligned} \quad (15)$$

Since $\mathbf{C}_{qq} \dot{\mathbf{x}}_q = \mathbf{0}$ and $\mathbf{C}_{qq} \ddot{\mathbf{x}}_q = -\dot{\mathbf{C}}_{qq} \dot{\mathbf{x}}_q$, (15) can be transformed into

$$\begin{aligned} \mathbf{C}_{qq} \ddot{\mathbf{x}}_q &= \mathbf{C}_{qq} \mathbf{M}_q^{-1} \mathbf{h}_q + \mathbf{C}_{qq} \mathbf{M}_q^{-1} \mathbf{C}_{qq}^T \boldsymbol{\lambda}_{qq}, \\ \boldsymbol{\lambda}_{qq} &= (\mathbf{C}_{qq} \mathbf{M}_q^{-1} \mathbf{C}_{qq}^T)^{-1} (\mathbf{C}_{qq} \mathbf{M}_q^{-1} \mathbf{h}_q + \dot{\mathbf{C}}_{qq} \dot{\mathbf{x}}_q). \end{aligned} \quad (16)$$

$\boldsymbol{\lambda}_I$ included in (14) is produced from (16).

2.3. *Velocity Transformation.* In the case of touching each leg of the quadruped robot to the ground, a collision occurs and the velocities of the components change discontinuously. The velocities after the collision can be determined by the constraint conditions (12) posed to the robot when the collision with the ground occurs and the velocities before the collision. According to Assumption 4, the collision is assumed as completely inelastic collision, and the velocities after the collision are obtained from the velocities before the collision as follows.

Transforming (9) gives $\boldsymbol{\lambda}_q$ as

$$\begin{aligned} \boldsymbol{\lambda}_q &= -\mathbf{X}_q^{-1} (\mathbf{C}_q \mathbf{M}_q^{-1} \mathbf{h}_q + \dot{\mathbf{C}}_q \dot{\mathbf{x}}_q + \mathbf{C}_q \mathbf{M}_q^{-1} \boldsymbol{\tau}_I), \\ \mathbf{X}_q &:= \mathbf{C}_q \mathbf{M}_q^{-1} \mathbf{C}_q^T. \end{aligned} \quad (17)$$

Substituting (10) and (17) into (9) yields

$$\begin{aligned} \mathbf{M}_q \ddot{\mathbf{x}}_q &= \mathbf{Y}_q \mathbf{h}_q - \mathbf{C}_q^T \mathbf{X}_q^{-1} \dot{\mathbf{C}}_q \dot{\mathbf{x}}_q + \mathbf{Y}_q \mathbf{C}_I^T \boldsymbol{\lambda}_I, \\ \mathbf{Y}_q &:= \mathbf{I}_9 - \mathbf{C}_q^T \mathbf{X}_q^{-1} \mathbf{C}_q \mathbf{M}_q^{-1}. \end{aligned} \quad (18)$$

Let $\dot{\mathbf{x}}_q^-$ denote the velocities before the collision and let $\dot{\mathbf{x}}_q^+$ denote the velocities after the collision. From (18), we obtain the following velocity relationship between the velocities before the collision and after that:

$$\mathbf{M}_q \dot{\mathbf{x}}_q^+ - \mathbf{M}_q \dot{\mathbf{x}}_q^- = \mathbf{Y}_q \mathbf{C}_I^T \boldsymbol{\lambda}_I. \quad (19)$$

Since $\dot{\mathbf{x}}_q^+$ should satisfy $\mathbf{C}_I \dot{\mathbf{x}}_q^+ = \mathbf{0}$, $\boldsymbol{\lambda}_I$ is given by

$$\begin{aligned} \boldsymbol{\lambda}_I &= -\mathbf{Z}_q^{-1} \mathbf{C}_I \dot{\mathbf{x}}_q^-, \\ \mathbf{Z}_q &:= \mathbf{C}_I \mathbf{M}_q^{-1} \mathbf{Y}_q \mathbf{C}_I^T. \end{aligned} \quad (20)$$

The velocities after the collision $\dot{\mathbf{x}}_q^+$ are thus obtained by substituting (20) into (19) as

$$\dot{\mathbf{x}}_q^+ = (\mathbf{I}_9 - \mathbf{M}_q^{-1} \mathbf{Y}_q \mathbf{C}_I^T \mathbf{Z}_q^{-1} \mathbf{C}_I) \dot{\mathbf{x}}_q^-. \quad (21)$$

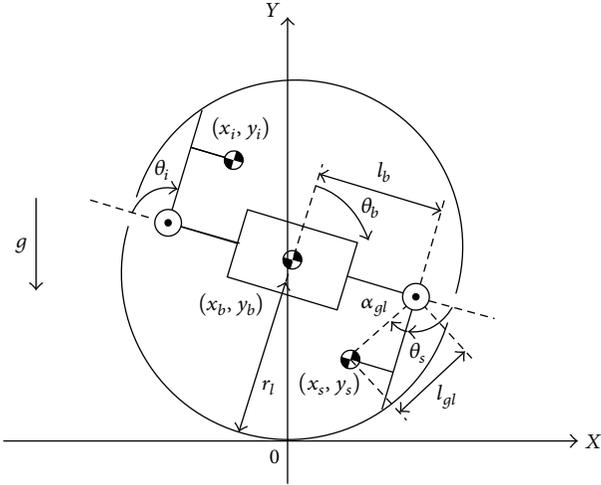


FIGURE 5: Transformed model diagram of the rolling huntsman-spider-inspired quadruped robot. A grounding leg is defined as the supporting leg and another leg is defined as the idling leg. The grounding point of the supporting leg is defined as the origin 0, which is switched to the point after switching each supporting leg.

3. Design of the Energy-Based Controller

This section presents the controller based on the robot's energy during rolling locomotion. The quadruped robot loses some of the robot's energy due to collision and contact with the ground while rolling, and therefore it cannot perform periodic rolling locomotion without energy supply. It hence requires a controller which can compensate the robot's energy loss [25].

We derive a motion equation and mechanical energy of the quadruped robot based on its grounding point on a supporting leg. Applying them, the controller generates joint torque for its supporting leg that allows the robot's kinetic energy to reach target energy at the moment when it switches the supporting leg.

The quadruped robot performs an energy recovery action immediately before switching the supporting leg. Otherwise it returns and keeps the legs to initial positions.

3.1. Transforming the Motion Equation. The transformed motion equation and the mechanical energy of the quadruped robot are derived by defining a grounding leg as the supporting leg and another leg as the idling leg and shifting the origin 0 to the grounding point of the supporting leg as shown in Figure 5. Here the subscripts s and i denote the supporting leg and the idling leg, respectively. The origin 0 is switched to the grounding point of the supporting leg after switching each supporting leg.

The transformed motion equation is derived by modifying the constraint conditions and the independent velocity vector. The constraint conditions after shifting the origin are given by

$$\begin{aligned} x_b &= -l_c \cos(\theta_b + \theta_s + \alpha_c) - l_b \cos \theta_b, \\ y_b &= l_c \sin(\theta_b + \theta_s + \alpha_c) + l_b \sin \theta_b + r_l, \end{aligned}$$

$$\begin{aligned} x_s &= -l_c \cos(\theta_b + \theta_s + \alpha_c) + l_{gl} \cos(\theta_b + \theta_s + \alpha_{gl}), \\ y_s &= l_c \sin(\theta_b + \theta_s + \alpha_c) - l_{gl} \sin(\theta_b + \theta_s + \alpha_{gl}) + r_l, \\ x_i &= -l_c \cos(\theta_b + \theta_s + \alpha_c) - 2l_b \cos \theta_b \\ &\quad - l_{gl} \cos(\theta_b + \theta_i + \alpha_{gl}), \\ y_i &= l_c \sin(\theta_b + \theta_s + \alpha_c) + 2l_b \sin \theta_b \\ &\quad + l_{gl} \sin(\theta_b + \theta_i + \alpha_{gl}) + r_l. \end{aligned} \quad (22)$$

The independent velocity vector $\dot{\mathbf{q}}_q$ is also modified as

$$\dot{\mathbf{q}}_q = [\dot{\theta}_b, \dot{\theta}_s, \dot{\theta}_i]^T, \quad (23)$$

where $\dot{\theta}_s$ and $\dot{\theta}_i$ are the controllable variables and $\dot{\theta}_b$ is uncontrollable one; however, $\dot{\theta}_i$ is not utilized for energy compensation control. Applying the modified constraint conditions (22) and the independent velocity vector (23) yields the constraint matrix \mathbf{C}_q and the orthogonal matrix \mathbf{D}_q in the manner described in the previous section, and thereby we can derive the quadruped robot motion equation with the origin at the grounding point of the supporting leg.

The transformed motion equation is represented by

$$\mathbf{M}_c(\mathbf{q}_q) \ddot{\mathbf{q}}_q + \mathbf{C}_c(\mathbf{q}_q, \dot{\mathbf{q}}_q) \dot{\mathbf{q}}_q + \frac{\partial U}{\partial \mathbf{q}_q} = \mathbf{T}_c, \quad (24)$$

where $\mathbf{M}_c(\mathbf{q}_q)$ is the inertia matrix, $\mathbf{C}_c(\mathbf{q}_q, \dot{\mathbf{q}}_q)$ is the damping matrix, U is the potential energy, and \mathbf{T}_c is the input torque and also $\mathbf{M}_c(\mathbf{q}_q) = \mathbf{D}_q^T \mathbf{M}_q \mathbf{D}_q$ and $\mathbf{T}_c = [0, \tau_s, \tau_i]^T$.

The mechanical energy E , the kinetic energy K , and the potential energy U are defined as

$$\begin{aligned} E &= K + U, \\ K &= \frac{1}{2} \dot{\mathbf{q}}_q^T \mathbf{M}_c(\mathbf{q}_q) \dot{\mathbf{q}}_q, \\ U &= (m_b y_b + m_l (y_s + y_i)) g. \end{aligned} \quad (25)$$

Additionally the relationship between the time derivative mechanical energy and the input torque is represented by

$$\dot{E} = \dot{K} + \dot{U} = \dot{\mathbf{q}}_q^T \mathbf{T}_c. \quad (26)$$

3.2. Energy Compensation Control. Some of the kinetic energy K is lost due to collision and contact with the ground. We hence set the kinetic energy at the moment when the quadruped robot switches the supporting leg for the case of completing rolling locomotion as the target energy E_d for control and assume that $K = E_d$ is satisfied at the moment while it is completing that. On the basis of the above, the energy state function V is defined as

$$\begin{aligned} V &= \frac{1}{2} (K - E_d)^2, \\ \dot{V} &= (K - E_d) \dot{K}. \end{aligned} \quad (27)$$

Equation (27) is $V > 0$ obviously and satisfies $V = 0$ for $K = E_d$. When $\dot{V} < 0$, $V \rightarrow 0$ and $K \rightarrow E_d$ is satisfied. Rolling locomotion is thus achieved by satisfying $K \rightarrow E_d$ when the joint torque which satisfies $\dot{V} < 0$ is given to the supporting leg.

The time derivative of the energy state function \dot{V} is written by applying (26) and (27) as

$$\begin{aligned}\dot{V} &= (K - E_d)(\dot{\mathbf{q}}_q^T \mathbf{T}_c - \dot{U}), \\ &= (K - E_d)(\dot{\theta}_s \tau_s + \dot{\theta}_i \tau_i - \dot{U}) < 0.\end{aligned}\quad (28)$$

Since the kinetic energy is recovered only by movement of the supporting leg, let $\dot{\theta}_i \tau_i = 0$. Equation (28) is definitely transformed into

$$\dot{V} = (K - E_d)(\dot{\theta}_s \tau_s - \dot{U}) < 0. \quad (29)$$

This paper supposes that the quadruped robot should roll only in the positive direction on the X -coordinate. In order for the robot to roll in only one direction, we restrict the angular velocity of the supporting leg during an energy recovery action to $\dot{\theta}_s < 0$. Under this restriction, if $K - E_d < 0$, that is, $K < E_d$, then

$$\dot{\theta}_s \tau_s - \dot{U} > 0. \quad (30)$$

Since $\dot{\theta}_s < 0$, let $\dot{\theta}_s = -|\dot{\theta}_s|$. Equation (30) is calculated as

$$\tau_s < \begin{cases} -\left|\frac{\dot{U}}{\dot{\theta}_s}\right|, & (\dot{U} \geq 0 \cap \dot{\theta}_s < 0), \\ \left|\frac{\dot{U}}{\dot{\theta}_s}\right|, & (\dot{U} < 0 \cap \dot{\theta}_s < 0). \end{cases} \quad (31)$$

If $K - E_d \geq 0$, that is, $K \geq E_d$, then

$$\tau_s > \begin{cases} -\left|\frac{\dot{U}}{\dot{\theta}_s}\right|, & (\dot{U} \geq 0 \cap \dot{\theta}_s < 0), \\ \left|\frac{\dot{U}}{\dot{\theta}_s}\right|, & (\dot{U} < 0 \cap \dot{\theta}_s < 0). \end{cases} \quad (32)$$

The input is determined to satisfy (31) and (32). The input τ_s is defined as

$$\tau_s = \begin{cases} -\zeta \left|\frac{\dot{U}}{\dot{\theta}_s}\right|, & (K - E_d < 0 \cap \dot{\theta}_s < 0), \\ \zeta \left|\frac{\dot{U}}{\dot{\theta}_s}\right|, & (K - E_d \geq 0 \cap \dot{\theta}_s < 0), \\ 0, & (\dot{\theta}_s \geq 0), \end{cases} \quad (33)$$

where ζ is the gain to adjust the input and $1 < \zeta < \infty$. Besides an initial angular velocity is provided by another input since $\dot{\theta}_s$ has the singular point in (33).

To determine ζ , values of the energy state function V at the moment when the quadruped robot switches the supporting leg are set as the Poincare mapping $\xi = \{\xi_1, \xi_2, \dots, \xi_k, \dots\}$.

TABLE 3: Controller parameters.

E_d	5.00×10^{-2}
α_0	10.0
K_p	1.00×10^2
K_i	20.0
K_d	1.00×10^{-1}

Here ξ_k denotes the value of the energy state function V for the time t_k at the moment. The control target is achieved when the Poincare mapping is settled to 0. The gain ζ settling the Poincare mapping to 0 is defined as

$$\zeta = \zeta_0 + \sum_{k=1}^N \left(\kappa_p \Delta \xi_k + \kappa_i \sum_{n=1}^k \Delta \xi_n + \kappa_d \frac{\Delta \xi_k - \Delta \xi_{k-1}}{\Delta t_k} \right), \quad (34)$$

where $\Delta \xi_k = \xi_k - \xi_*$ is the difference between the value ξ_k and the target value $\xi_* = 0$ of the energy state function V , $\Delta t_k = t_k - t_{k-1}$ is the difference between the time t_k and the previous time t_{k-1} at the moment, and ζ_0 , κ_p , κ_i , and κ_d are the adjustable parameters and should be positive constant.

4. Simulation of Rolling Locomotion on the Flat Ground

Rolling locomotion on the quadruped robot on the flat ground is simulated to verify the effectiveness of the proposed controller. The initial state of the rotational angle of the body and the legs is set at $\theta_b = 8.73 \times 10^{-2}$ rad and $\theta_{ij} = 1.57$ rad, respectively. The quadruped robot starts rolling locomotion when the first leg contacts the ground. The robot performs the energy recovery action immediately before switching the supporting leg. At this time the horizontal COG of the robot $X_g = (m_b x_b + m_l(x_s + x_i))/(m_b + 2m_l) \geq 1.25 \times 10^{-2}$.

The model described in the section about modeling is applied as a plant. A PID controller is applied with the proposed controller in order to return and keep the legs to the initial positions and provide an initial angular velocity for the supporting leg. The PID controller works when the proposed controller is not active.

The controller parameters are shown in Table 3.

Simulation results are shown in Figures 6–11. Figure 6 shows the rotational angle of the body, Figures 7 and 8 show the positions of the body and the legs, Figure 9 shows the rotational angles of the legs, Figure 10 shows the joint torque values with energy compensation control, and Figure 11 shows the error between the kinetic energy and the target energy.

Figures 6 and 7 show that the quadruped robot moves in the positive direction with rolling continuously. It rotates 5.57 times and moves 2.91 m in 10 s. Figure 8 shows that Y direction positions repeatedly increase and decrease. The positional relationship between the legs in Figure 8 shows that it rolls with switching the supporting leg.

Figure 9 shows that the quadruped robot performs the energy recovery action immediately before switching the supporting leg and returns it to the initial position immediately

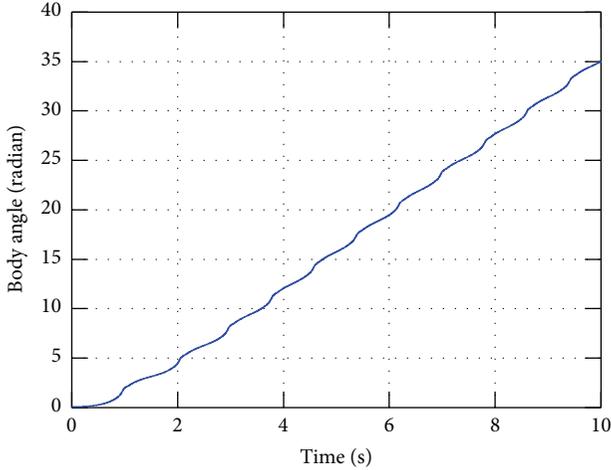


FIGURE 6: Rotational angle of the body. The rotational angle of the body increases over time. It means that the quadruped robot rolls continuously, and the robot rotates 5.57 times in 10 s.

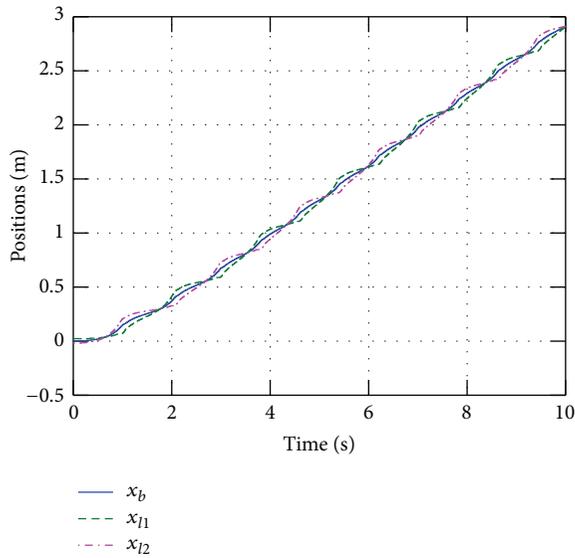


FIGURE 7: X direction positions of the body and the legs (x_b : body, x_{l_j} : leg j). The X direction positions of the body and the legs increase over time. It means that the quadruped robot moves in the positive direction, and the robot moves 2.91 m in 10 s.

after that. The legs rotate in the negative direction when it is the supporting leg. Figure 9 also shows that the angular variations of the legs are converged to periodic trajectories. The joint torque values for the energy recovery action shown in Figure 10 allow the error of energy to be more than 0 as shown in Figure 11. It means that the robot's energy loss is compensated by the proposed controller. The above results show that the proposed controller is effective in achieving periodic rolling locomotion on the flat ground with the quadruped robot.

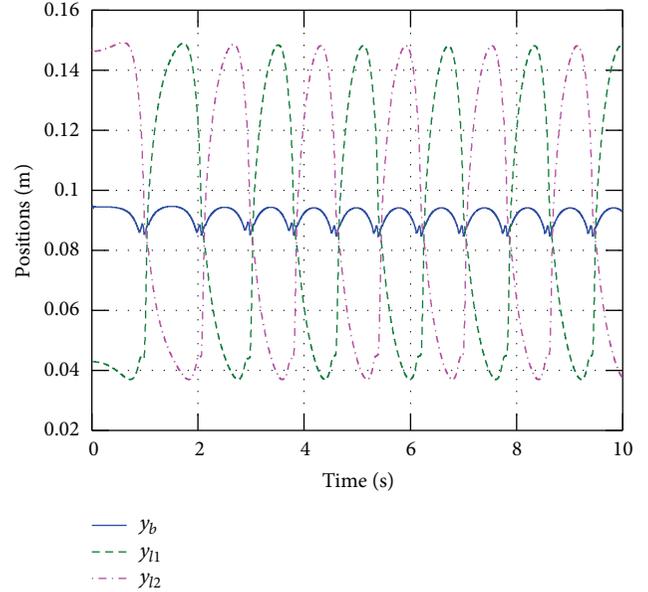


FIGURE 8: Y direction positions of the body and the legs (y_b : body, y_{l_j} : leg j). The Y direction positions of the body and the legs repeatedly increase and decrease. The positional relationship between the legs shows that the quadruped robot rolls with switching the supporting leg.

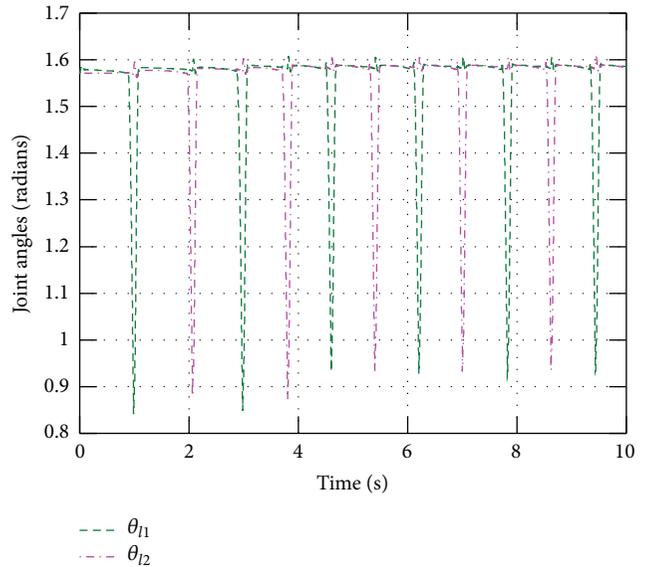


FIGURE 9: Rotational angles of the legs (θ_{l_j} : leg j). The quadruped robot moves the supporting leg in the negative direction immediately before switching the supporting leg to increase the robot's kinetic energy. The legs return to its initial position immediately after that.

5. Conclusion

This paper has presented an approach to control rolling locomotion with a huntsman-spider-inspired quadruped robot. A dynamic model of the quadruped robot with rolling locomotion has been developed by applying a constraint

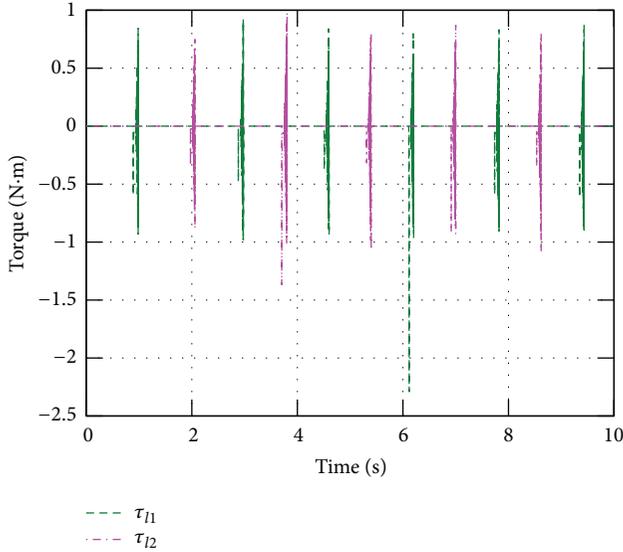


FIGURE 10: Joint torque values with energy compensation control (τ_{1j} : leg j). The negative joint torque values are supplied to the legs immediately before the quadruped robot switches the supporting leg.

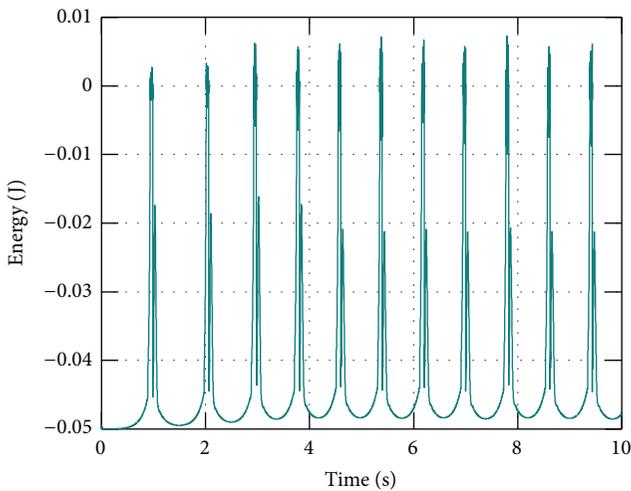


FIGURE 11: Error of energy. The joint torque values allow the error of energy to be more than 0, which means that the robot's energy loss is compensated by the proposed controller.

force due to collision and contact with the ground to a base quadruped robot model. The rolling locomotion is limited to planar one by an assumption that the quadruped robot does not fall down while rolling. A transformed model and mechanical energy of the quadruped robot have been derived based on its grounding point on a supporting leg. Employing these, a controller which can compensate the robot's energy loss during rolling locomotion has been designed. The effectiveness of the proposed controller has been verified through a numerical simulation of rolling locomotion on the flat ground. The simulation results have shown that the quadruped robot can perform periodic rolling locomotion

with the proposed energy-based controller. The proposed control approach is effective in achieving periodic rolling locomotion in conclusion. The proposed controller will be implemented in our platforms and its effectiveness will be tested in future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] W.-M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein, and J. Venkatesh, "Multimode locomotion via SuperBot reconfigurable robots," *Autonomous Robots*, vol. 20, no. 2, pp. 165–177, 2006.
- [2] S. Murata and H. Kurokawa, "Self-reconfigurable robots," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 71–78, 2007.
- [3] M. Yim, W.-M. Shen, B. Salemi et al., "Modular self-reconfigurable robot systems [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 43–52, 2007.
- [4] P. Pirjanian, C. Leger, E. Mumm et al., "Distributed control for a modular, reconfigurable cliff robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '02)*, pp. 4083–4088, May 2002.
- [5] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo, "The cooperation of swarm-bots: physical interactions in collective robotics," *IEEE Robotics & Automation Magazine*, vol. 12, no. 2, pp. 21–28, 2005.
- [6] V. Trianni, S. Nolfi, and M. Dorigo, "Cooperative hole avoidance in a swarm-bot," *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 97–103, 2006.
- [7] O. Mori and T. Omata, "Coupling of two 2-link robots with a passive joint for reconfigurable planar parallel robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '02)*, vol. 4, pp. 4120–4125, May 2002.
- [8] T. Yamawaki, O. Mori, and T. Omata, "Nonholonomic dynamic rolling control of reconfigurable 5R closed kinematic chain robot with passive joints," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '03)*, vol. 3, pp. 4054–4059, September 2003.
- [9] C. C. Phipps and M. A. Minor, "Introducing the hex-a-ball, a hybrid locomotion terrain adaptive walking and rolling Robot," in *Proceedings of the 8th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR '05)*, pp. 525–532, September 2005.
- [10] M. Ohira, R. Chatterjee, T. Kamegawa, and F. Matsuno, "Development of three-legged modular robots and demonstration of collaborative task execution," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 3895–3900, April 2007.
- [11] C. C. Phipps, B. E. Shores, and M. A. Minor, "Design and quasi-static locomotion analysis of the rolling disk biped hybrid robot," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1302–1314, 2008.
- [12] S. Nansai, N. Rojas, M. R. Elara, and R. Sosa, "Exploration of adaptive gait patterns with a reconfigurable linkage mechanism," in *Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 4661–4668, November 2013.

- [13] A. Castano, W.-M. Shen, and P. Will, "Conro: towards deployable robots with inter-robots metamorphic capabilities," *Autonomous Robots*, vol. 8, no. 3, pp. 309–324, 2000.
- [14] W.-M. Shen, B. Salemi, and P. Will, "Hormone-inspired adaptive communication and distributed control for conro self-reconfigurable robots," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 700–712, 2002.
- [15] R. Pfeifer, M. Lungarella, and F. Iida, "Self-organization, embodiment, and biologically inspired robotics," *Science*, vol. 318, no. 5853, pp. 1088–1093, 2007.
- [16] J.-A. Meyer and A. Guillot, "Biologically inspired robots," in *Springer Handbook of Robotics*, pp. 1395–1422, Springer, Berlin, Germany, 2008.
- [17] H.-T. Lin, G. G. Leisk, and B. Trimmer, "Goqbot: a caterpillar-inspired soft-bodied rolling robot," *Bioinspiration & Biomimetics*, vol. 6, no. 2, Article ID 026007, 2011.
- [18] R. S. King, *BiLBIO: A Biologically Inspired Robot with Walking and Rolling Locomotion*, Springer, 2013.
- [19] A. Sinha, N. Tan, and R. E. Mohan, "Terrain perception for a reconfigurable biomimetic robot using monocular vision," *Robotics and Biomimetics*, vol. 1, no. 1, article 23, 11 pages, 2014.
- [20] W. H. Lee and A. C. Sanderson, "Dynamic rolling locomotion and control of modular robots," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 32–41, 2002.
- [21] J. Sastra, S. Chitta, and M. Yim, "Dynamic rolling for a modular loop robot," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 758–773, 2009.
- [22] K. J. Åström and K. Furuta, "Swinging up a pendulum by energy control," *Automatica*, vol. 36, no. 2, pp. 287–295, 2000.
- [23] K. J. Åström, J. Aracil, and F. Gordillo, "A new family of smooth strategies for swinging up a pendulum," in *Proceedings of the 16th Triennial World Congress of International Federation of Automatic Control (IFAC '05)*, pp. 1118–1123, Elsevier, July 2005.
- [24] F. Asano, Z.-W. Luo, and M. Yamakita, "Biped gait generation and control based on a unified property of passive dynamic walking," *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 754–762, 2005.
- [25] S. Kojima and M. Iwase, "Yo-yo motion control based on impulsive luenberger observer," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC '11)*, pp. 5323–5328, IEEE, December 2011.
- [26] S. Nansai, M. Iwase, and M. R. Elara, "Energy based position control of jansen walking robot," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '13)*, pp. 1241–1246, October 2013.
- [27] K. Arczewski and W. Blajer, "A unified approach to the modelling of holonomic and nonholonomic mechanical systems," *Mathematical Modelling of Systems*, vol. 2, no. 3, pp. 157–174, 1996.
- [28] W. Blajer, "A geometrical interpretation and uniform matrix formulation of multibody system dynamics," *Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 81, no. 4, pp. 247–259, 2001.
- [29] H. Ohsaki, M. Iwase, and S. Hatakeyama, "A consideration of nonlinear system modeling using the projection method," in *Proceedings of the 46th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE '07)*, pp. 1915–1920, IEEE, 2007.
- [30] H. Ohta, M. Yamakita, and K. Furuta, "From passive to active dynamic walking," *International Journal of Robust and Nonlinear Control*, vol. 11, no. 3, pp. 287–303, 2001.
- [31] A. Ohata, A. Sugiki, Y. Ohta, S. Suzuki, and K. Furuta, "Engine modeling based on projection method and conservation laws," in *Proceedings of the IEEE International Conference on Control Applications*, vol. 2, pp. 1420–1424, IEEE, September 2004.

Research Article

Development and Motion Testing of a Robotic Ray

Jianhui He and Yonghua Zhang

Department of Mechatronics Engineering, Taizhou Vocational & Technical College, Taizhou 318000, China

Correspondence should be addressed to Jianhui He; hjh1103@mail.ustc.edu.cn

Received 16 October 2014; Revised 26 December 2014; Accepted 9 January 2015

Academic Editor: Nan Xiao

Copyright © 2015 J. He and Y. Zhang. 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.

Biomimetics takes nature as a model for inspiration to immensely help abstract new principles and ideas to develop various devices for real applications. In order to improve the stability and maneuvering of biomimetic fish like underwater propulsors, we selected bluespotted ray that propel themselves by taking advantage of their pectoral fins as target. First, a biomimetic robotic undulating fin driven propulsor was built based on the simplified pectoral structure of living bluespotted ray. The mechanical structure and control circuit were then presented. The fin undulating motion patterns, fin ray angle, and fin shape to be investigated are briefly introduced. Later, the kinematic analysis of fin ray and the whole fin is discussed. The influence of various kinematic parameters and morphological parameters on the average propulsion velocity of the propulsor was analyzed. Finally, we conclude that the average propulsion velocity generally increases with the increase of kinematic parameters such as frequency, amplitude, and wavelength, respectively. Moreover, it also has a certain relationship with fin undulating motion patterns, fin ray angle, fin shape, and fin aspect ratio.

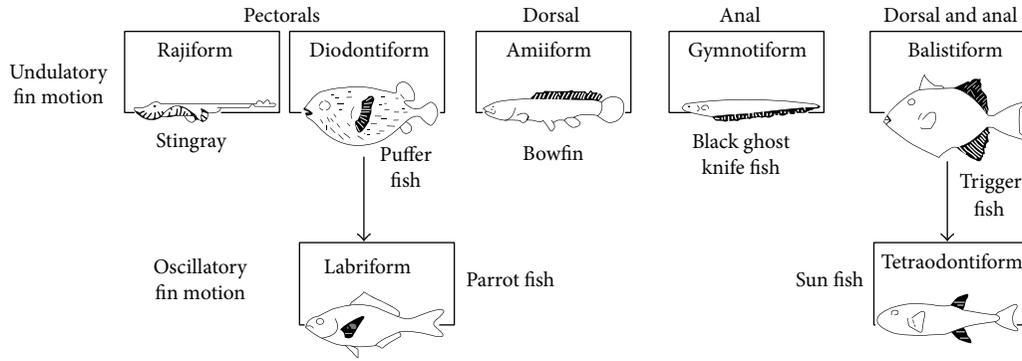
1. Introduction

Each fish species has its own unique way of interacting with different environments, which then dictates the species' shape and size, as well as the way it propels itself, through a process of natural selection. Therefore, fish provide useful illustrations of propulsor design, swimming modes, and body/fin shape (morphology). An estimated 15% of the fish families in the world use median and pectoral fins, termed median and/or paired fin (MPF) locomotion [1]. Obviously, batoids are very famous for their MPF locomotion who exhibit two modes of pectoral swimming behavior: (1) undulatory locomotion, termed "rajiform" and (2) oscillatory locomotion, termed "mobuliform." Rajiform locomotion is performed by skates and most stingrays and involves undulatory waves that are propagated down the fins from anterior to posterior [2, 3]. Compared with body and/or caudal fin (BCF) locomotion fishes, batoids have remarkable manoeuvrability and can efficiently stabilize themselves in currents and surges and are more hydromechanically efficient at low-speeds than BCF periodic swimmers [4]. They leave a less noticeable wake than BCF locomotion fishes and are capable of turning on their own axis with little or no lateral translation of

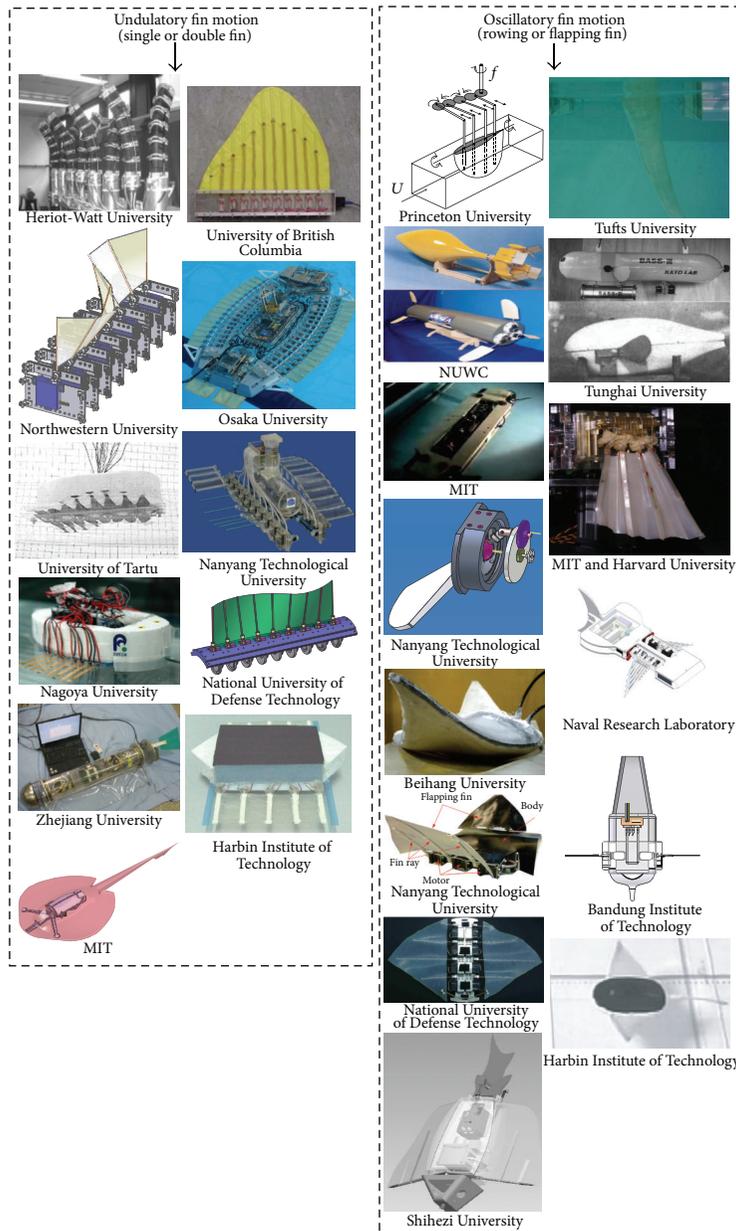
the body [5–7]. These characteristics make batoid fishes an ideal platform to emulate in the design of a bioinspired autonomous underwater vehicle (AUV).

On the other hand, the demands for general aquatic locomotion research, environmental protection, ocean exploration, military reconnaissance, data gathering, search and rescue devices, video exploration, and remote maintenance operations have become more urgent and desirable than ever before. But the high cost of performing such operations with ship based platforms prevents researchers from probing the oceans extensively [8]. Thus, with the development of low-cost robotic technologies, several teams have developed the robotics prototype resembling real fish with MPF modes. Figure 1 illustrates some relevant robotic fishes developed and tested [1, 9–36]. The biomimetic undulating fin mechanisms were developed by using various actuators, such as motor [9–30], parallel bellows actuator (PBA) [1], IPMC [31–34], and SMA [35, 36].

An important design consideration for swimming machines is the design of propulsors: their shape, location on the machine, pattern of movement, and mechanical and material properties (e.g., inertia and stiffness). The overall shape of the robot is another important consideration. As



(a)



(b)

FIGURE 1: Swimming modes of fishes using MPF propulsion. Hatched areas in Figure 2(a) shows the propulsive segment that contributes to thrust generation. Figure 2(b) shows the robotic fishes that use MPF modes [1, 9–36].

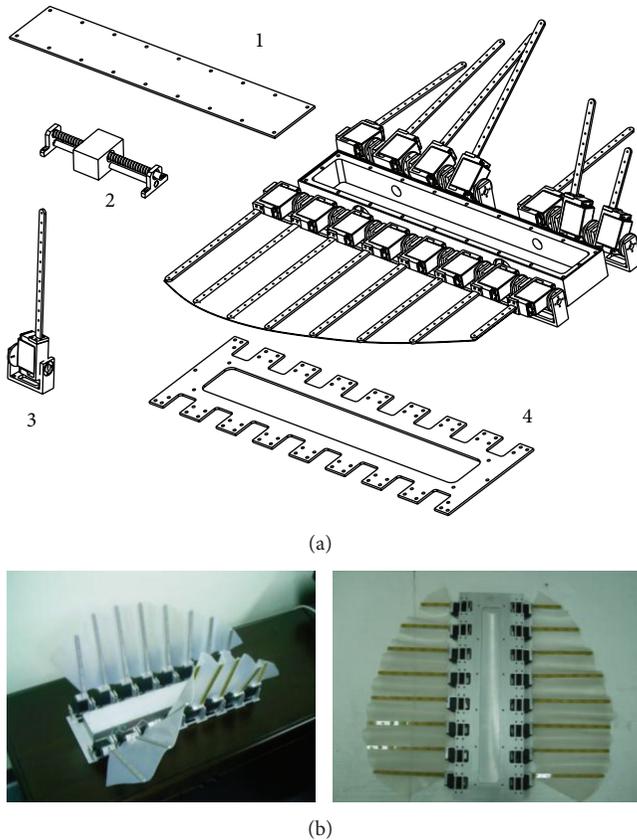


FIGURE 2: Layout of prototype. 1. Lid; 2. center of gravity adjustment mechanism; 3. oscillating module; 4. baseboard.

fish are impressive swimmers in many ways, it is hoped that submersible robots that swim like fish might be superior to submersibles using propellers [37]. Our long term goal is to develop an AUV that is capable of civil or military application. However, PBA actuators are too clumsy and require complex gas supply system. SMA actuators are highly nonlinear and possibly difficult to achieve precise deformation control. Meanwhile, the output power of IPMC actuators is too small. Thus, from the perspective of practical application, motor actuators may be not optimal but much feasible.

Several motor driven robotic undulating fins based AUVs have already been built. Despite the fact that all of these robots aim to provide alternatives to propellers, their propulsion mechanisms differ substantially. Accordingly, we subdivide it into three types: (1) a mechanical implementation relying on camshafts to generate the propagating wave which comes at the price of limited flexibility of parameter variation [18]. (2) the oscillating motion of fin rays is independently controlled by servomotors, and alternate oscillation of fin rays can produce undulating motion on fin membrane [16, 17]. (3) The undulating motion of the whole robotic fin is produced by oscillating motion of a single anteriorly equipped fin ray, and the vibration subsequently spreads to the rear end [12]. The abovementioned second type could be categorized into two types depending on the way the fin rays are connected. One is the fin rays that are parallel-arranged and connected on a

fixed baseline [16, 17]. The other one is the fin rays that are connected on a free baseline through cranks. Each crank is directly driven by a motor and baseline moves as all cranks oscillate [11]. Unlike the prototype built by BOILEAU [16], we simplify the oscillating mechanism and make the whole structure much more compact and reliable.

The present work is concerned with the design of bluespotted ray inspired underwater propulsor possessing two large lateral expanded undulating fins and the parametric study of propulsion performance. The rest of paper is organized as follows. In Section 2, we present detailed mechanism design. The undulating patterns, fin ray angle, and fin shape to be investigated are also briefly introduced. In Section 3, kinematic analysis of the proposed mechanism is proposed. In Section 4, the influence of various kinematic parameters and morphological parameters on the average propulsion velocity of the propulsor was analyzed. Finally, the work is concluded with some remarks.

2. Mechanism and Control

Kier and Thompson suggested that the fins of a stingray are supported by three dimensional arrays of muscle [38]. Existing actuators, both linear and rotary, are unable to model the complex musculature of these fins. Despite the complexity of the actual musculature, the fins of a stingray exhibit much the same undulations as those displayed by the fins of ray-finned fish using an undulatory swimming mode. As a possible simplification, the fin of a stingray is divided into many segments such that the fin looks similar to that of a ray-finned fish.

2.1. Mechanical Structure. Figure 2 shows the mechanical structure of bluespotted ray like biorobotic underwater propulsor with a modular undulating fin consisting of eight equally spaced servomotors attached to a lightweight structure on both sides. The prototype design of the developed robot comprises three individual modules: two pectoral fin modules, electronics housing module, and center of gravity adjustment module. Tail is not considered for simplicity. The propulsor is an exact copy of the bluespotted ray anatomy, with a length of 0.71 m and width of 0.64 m, which fully mimics the anatomical dimensions of a mature ray.

Thereinto, the fin ray element consists of shaft sleeve, fixed nut, steering-gear bracket, steering-gear seat, and steering gear, shown in Figure 3. The housing is used to install battery, control electronics, and center of gravity adjustment module. All of them are installed on the baseboard. The prototype owns built-in energy sources. It is a self-sufficient prototype and gives an autonomy of 30 minutes of operation at moderate velocities. Meanwhile, two batteries provide the electrical energy required by electronics and control system components. The total batteries capacity is 3200 mAh. The mechanical design of the actuator system and the other sections provide robotic ray with the maximum interior capacity compared to the overall volume. This feature enables further modifications and makes the system suitable for additional accessory placements. The microcontroller (Altera

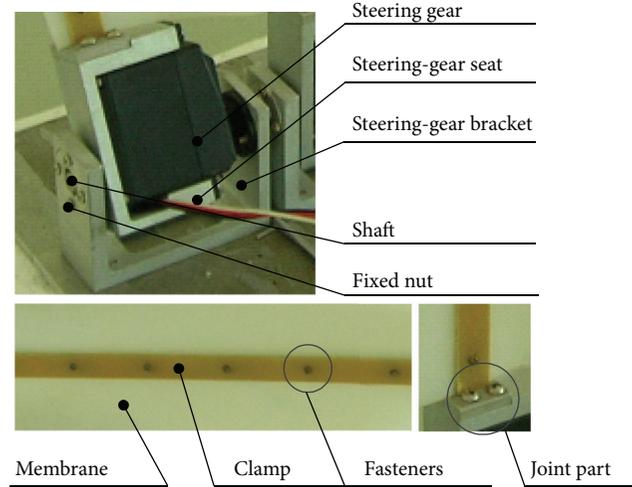


FIGURE 3: Oscillating module structure.

TABLE 1: Parameters of robotic ray.

Parameter	Unit	Values of robotic ray	Values of real bluespotted ray [3]
Total length	mm	710	710
Total width	mm	640	640
Total height	mm	42	42
The weight of the entire robot	kg	4.7	—
Fin ray number	—	8	>200
Fin ray length	mm	Adjustable	10~220
Fin ray width	mm	10	0.3~1.0
Fin ray height	mm	3	0.3~1.0
Fin ray space	mm	64.5 (equal distance)	<2.0 (nonequidistance)
Fin ray obliquity	°	30~90 (adjustable)	-100~85
Surface width	mm	462.5	710
Membrane modulus of elasticity	MPa	2.7	0.14
Frequency	Hz	≤3	≤3
Amplitude	°	≤60	≤45
Wavelength	mm	>475	355~890
Drive mode	—	Motor	Muscle

Max II, CPLD) based built-in control system controls the fin motions. Backed up with optical sensors which are used for environment exploration, this stand-alone system is the first step to the full autonomy of the prototype. Fin rays are made of elastic material (polypropylene) in order to add compliance of motion. This design is simpler than those using smart materials [39, 40] or multijoint linkage mechanisms [41–43]. Fin rays are designed as uniform strength cantilever beams in order to reduce the mass without causing structure failure in oscillations. Fin membrane is made of silicon rubber. The physical specifications of robotic ray are listed in Table 1.

2.2. Control. The fin of the real fish consists of fin rays that have varying span and stiffness with a flexible membrane connecting them together. Analogously, we develop a modular robotic fin which enables us to experiment with different materials for the rays and flexible membrane, each

fin ray is directly connected to its corresponding driver motor. The advantages of this direct-connection mode are that the amplitude, frequency, and phase of each fin ray can be independently controlled. Each motor drives a radial insert, the angle of which could be changed from 30° to 90° from the robotic ray centerline. The motor up/down moves the radial inserts generating waves that propagate along the circumference of the body. The control electronics provide a pulse width modulated (PWM) signal to control the amplitude and frequency of rotation of each motor. The motors operate at a standard voltage of 4.8 volts. At these volts, each motor delivers 8.0 kg-cm of torque and rotates at a maximum speed equivalent to 0.10 sec/60°. STR-36 series micropower wireless module is used as the wireless data transceiver in short-ranges. The outward appearance and modules of the control electronics board are shown in Figure 4.

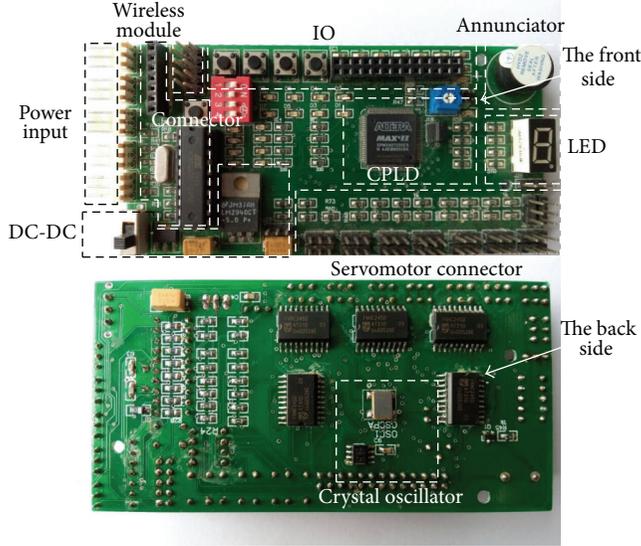


FIGURE 4: The control electronics.

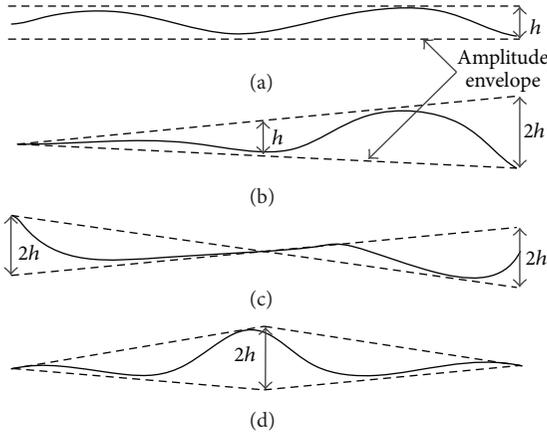


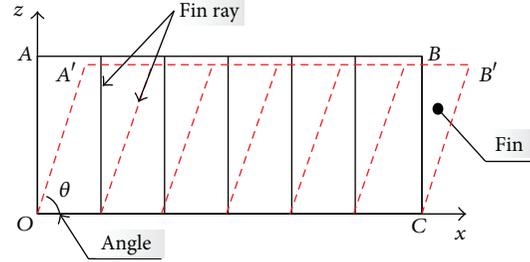
FIGURE 5: Four typical amplitude envelopes of modular undulating fin. (a) The amplitude envelope is fairly constant along the fins (Mode 1). (b) The amplitude envelope gradually increased from the anterior part to the posterior (Mode 2). (c) The amplitude envelope decreases from the anterior part to the mid part and increases toward the posterior (Mode 3). (d) The amplitude envelope increases from the anterior part to the mid part and decreases toward the posterior (Mode 4).

2.3. Undulating Patterns of the Robotic Ray. Some qualitative observations predicted that the propulsion velocity produced by the robotic ray is different among various fin undulating patterns. To reveal the regularity of this initial finding, four typical undulating fin patterns are selected and compared, with the same amplitude envelope area as well as some other kinematic parameters, such as frequency, wavelength, and wave propulsion velocity (Figure 5).

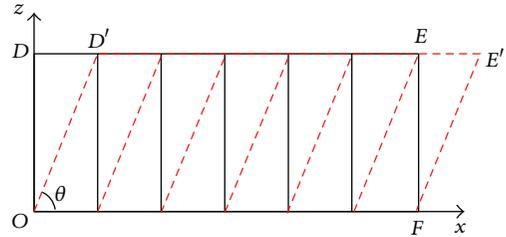
2.4. Fin Shape of Robotic Ray. In order to investigate the influence of fin morphology on propulsion performance, three morphologies of fins (triangle, rectangle, and trapezoid)

TABLE 2: List of three morphologies of fins.

Name	Fin shape	Aspect ratio	Fin area
Rectangle		0.8	
Trapezoid		1	0.45 m ²
Triangle		0.8, 1.0, 1.2, 1.4, 1.6	



(a) Constant fin ray length



(b) Constant fin area

FIGURE 6: Two cases of the change of fin ray angle (a) constant fin ray length and (b) constant fin area.

are performed with the same undulating patterns and surface area (Table 2).

2.5. Fin Ray Angle of Robotic Ray. The influence of fin ray angle on propulsion velocity is another consideration in our experimental investigation. Two cases are tested here. One is that the fin ray length keeps constant when its angle changes and thus the fin surface accordingly varies (Figure 6(a)). The other case is that the fin ray length varies when its angle changes to keep the fin surface constant (Figure 6(b)).

3. Modeling

3.1. Definition of Coordinate System. Figure 7 shows the definition of reference coordinate system. There are four main coordinate systems: earth coordinate system $O_w X_w Y_w Z_w$, body coordinate system $o_t x_t y_t z_t$, fin coordinate system $o_q^{(n)} x_q^{(n)} y_q^{(n)} z_q^{(n)}$, and fin ray coordinate system $o_r^{(n)} x_r^{(n)} y_r^{(n)} z_r^{(n)}$. Thereinto, body coordinate system is used for swimming kinematics description, meanwhile reflecting the motion state of biology relative to earth coordinate

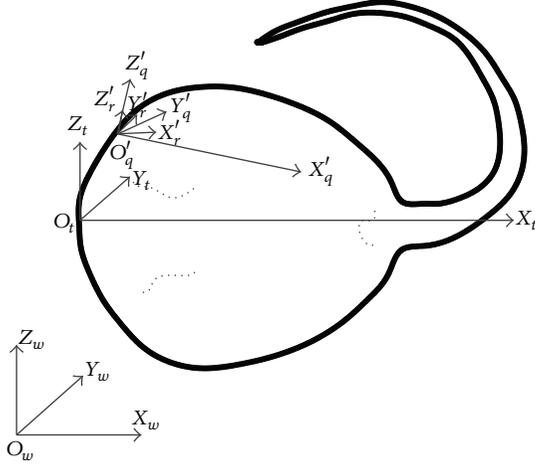


FIGURE 7: Definition of reference coordinate system.

system; fin coordinate system describes the undulatory movement characteristics of pectoral fins while fin ray coordinate system describes the up/down movement characteristics of fin rays.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{bmatrix} \cos \theta \cos \varphi & \sin \varphi \cos \theta & -\sin \theta \\ -\cos \gamma \sin \varphi + \sin \gamma \sin \theta \cos \varphi & \cos \varphi \cos \gamma + \sin \gamma \sin \theta \sin \varphi & \sin \gamma \cos \theta \\ \sin \gamma \sin \varphi + \cos \gamma \sin \theta \cos \varphi & \cos \gamma \sin \theta \sin \varphi - \sin \gamma \cos \varphi & \cos \gamma \cos \theta \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} x_{t0} \\ y_{t0} \\ z_{t0} \end{pmatrix}, \quad (3)$$

where θ , φ , and γ are the rotation angles around the three axes.

3.2. Kinematic Analysis on Fin Ray. Similar to Shirgaonkar et al. [13], Epstein et al. [17], and Hu et al. [18], undulating fin can be modeled as a ruled surface in 3D space. The fin baseline is the directrix of the ruled surface, while the fin ray is the generatrix. The undulation can then be generated through a sequential up/down motion of generatrix on the ruled surface. If we assume that all the fin rays' up/down motion are under the same sinusoidal waveforms. In the fin ray coordinate system, the space coordinate of a point $P(x_r, y_r, z_r)$ on the i th fin ray can be described as follows:

$$\begin{aligned} x_r &= 0, \\ y_r &= L_{ip} \cos(\theta_{fi}(i, t)), \\ z_r &= L_{ip} \sin(\theta_{fi}(i, t)), \end{aligned} \quad (4)$$

where L_{ip} is the distance from point P to fin ray base; $\theta_{fi}(i, t)$ is the angular position of the i th fin ray at time t given by

$$\theta_{fi}(i, t) = \theta_{fi \max}(x_i) \sin\left(\frac{2\pi t}{T_i} - \theta_{0i}\right), \quad (5)$$

The relationship among these coordinate systems is as follows.

(1) The relationship between fin coordinate system and fin ray coordinate system is given by

$$\begin{pmatrix} x_q \\ y_q \\ z_q \end{pmatrix} = \begin{bmatrix} \sin \alpha & \cos \alpha & 0 \\ -\cos \alpha & \sin \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix} + \begin{pmatrix} x_{r0} \\ y_{r0} \\ z_{r0} \end{pmatrix}, \quad (1)$$

where α is the angle between fin ray and fin baseline. In our prototype, $\alpha = \pi/2$. $[x_{r0} \ y_{r0} \ z_{r0}]$ means the origin coordinates translation.

(2) The relationship between fin coordinate system and body coordinate system is given by

$$\begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} = \begin{bmatrix} \cos \beta & \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_q \\ y_q \\ z_q \end{pmatrix} + \begin{pmatrix} X_{q0} \\ Y_{q0} \\ Z_{q0} \end{pmatrix}, \quad (2)$$

where β is the angle between fin baseline and body centerline. In our prototype, $\beta = 0$.

(3) The relationship between body coordinate system and earth coordinate system is given by

where

$$\begin{aligned} \theta_{fi \max}(x_i) &= \arcsin\left(\frac{f(x_i)}{L_{ir}}\right), \\ f(x_i) &= f(x)|_{x=x_i}, \\ \theta_{0i} &= 2\pi \frac{(i-1)L}{N\lambda}. \end{aligned} \quad (6)$$

$\theta_{fi \max}(x_i)$ is the amplitude of the wave at the i th fin ray; T_i is the up/down motion cycle, $T_i = T$; θ_{0i} is the initial phase of the i th fin ray; L_{ir} is the length of the i th fin ray; N is the total number of rays; λ is the wavelength; and L is the fin length. Thus, (4) can be rewritten as

$$\begin{aligned} x_r &= 0, \\ y_r &= L_{ip} \cos\left(\arcsin\left(\frac{f(x_i)}{L_{ir}}\right) \sin\left(\frac{2\pi t}{T_i} - 2\pi \frac{(i-1)L}{N\lambda}\right)\right), \\ z_r &= L_{ip} \sin\left(\arcsin\left(\frac{f(x_i)}{L_{ir}}\right) \sin\left(\frac{2\pi t}{T_i} - 2\pi \frac{(i-1)L}{N\lambda}\right)\right). \end{aligned} \quad (7)$$

Observation of ray swimming strongly suggests that the amplitude along the fin ray is not linear. The assumption of

a linear variation along the fin ray in this study is to simplify the analysis. We further get fin ray up/down motion angular velocity and angular acceleration through the first derivation and the second derivation of (5):

$$\begin{aligned}\omega_{fi}(i, t) &= \frac{d\theta_{fi}(i, t)}{dt} = \frac{2\pi\theta_{fi\max}}{T} \cos\left(\frac{2\pi t}{T} - \theta_{oi}\right), \\ a_{fi}(i, t) &= \frac{d^2\theta_{fi}(i, t)}{dt^2} = -\frac{4\pi^2\theta_{fi\max}}{T^2} \sin\left(\frac{2\pi t}{T} - \theta_{oi}\right).\end{aligned}\quad (8)$$

As indicated in (8), at the maximum up/down motion angle, the fin ray angular velocity is zero, while the acceleration reaches its maximum and vice versa.

3.3. Kinematic Analysis on Undulatory Fin. For present study, we idealize fin kinematics as a travelling sinusoid on an otherwise stationary (i.e., nontranslating and nonrotating) membrane. As a consequence, the baseline of the fin remains fixed at all times, and all points on the fin from baseline to distal edge move in a sinusoidal manner.

In the fin coordinate system (Figure 8), the coordinate of a point $S(x_q, y_q, z_q)$ on fin surface can be described as follows:

$$\begin{bmatrix} x_q \\ y_q \\ z_q \end{bmatrix} = \mathbf{I} \begin{bmatrix} 0 \\ L_s \cos(\theta_f) \\ L_s \sin(\theta_f) \end{bmatrix} + \begin{bmatrix} o_q o_r \\ 0 \\ 0 \end{bmatrix}.\quad (9)$$

$\begin{bmatrix} 0 \\ L_s \cos(\theta_f) \\ L_s \sin(\theta_f) \end{bmatrix}$ is the coordinate of point S in the fin ray coordinate system. $\begin{bmatrix} o_q o_r \\ 0 \\ 0 \end{bmatrix}$ is the translation matrix, where

$$\begin{aligned}\theta_f &= \theta_{f\max}(x_q) \sin\left[2\pi\left(\frac{t}{T} - \frac{x_q}{\lambda}\right)\right], \\ \theta_{f\max}(x_q) &= \arcsin\left(\frac{f(x_q)}{L_{rq}}\right).\end{aligned}\quad (10)$$

\mathbf{I} is the rotary matrix; in this case, it is an identity matrix. L_s is the distance from point S to baseline. θ_f is the angle between $O_r S$ and y_r axis. $f(x_q)$ is value of amplitude envelope function at x_q . Equation (11) is further expressed as

$$\begin{aligned}\begin{bmatrix} x_q \\ y_q \\ z_q \end{bmatrix} &= \begin{bmatrix} x_q \\ L_s \cos\left(\arcsin\left(\frac{f(x_q)}{L_{rq}}\right) \sin\left[2\pi\left(\frac{t}{T} - \frac{x_q}{\lambda}\right)\right]\right) \\ L_s \sin\left(\arcsin\left(\frac{f(x_q)}{L_{rq}}\right) \sin\left[2\pi\left(\frac{t}{T} - \frac{x_q}{\lambda}\right)\right]\right) \end{bmatrix}.\end{aligned}\quad (11)$$

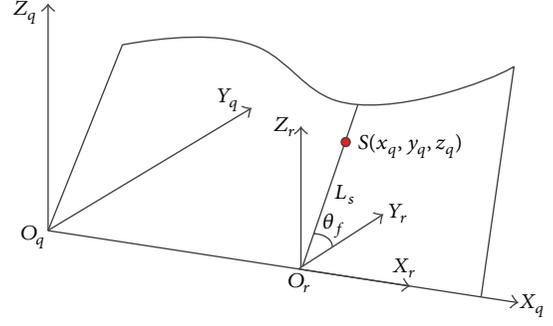


FIGURE 8: Kinematics modeling of undulatory fin.

The kinematics equation of biomimetic fin is then expressed as follows:

$$y_q(x_q, z_q, t) = A(x_q, z_q) \sin(\omega t - kx_q),\quad (12)$$

where $\omega = 2\pi f$ is undulatory angular frequency, $k = 2\pi/\lambda$ is wave number, and $A(x_q, z_q)$ is amplitude.

Here, if the fin ray is not perpendicular to fin base. That is, the angle between fin ray and x_q axis is θ ($\theta \neq \pi/2$); a more general form of (13) could be expressed below:

$$\begin{aligned}y_q(x_q, z_q, \theta, t) &= \frac{z_q \sqrt{1 + \tan^2(\theta)}}{\gamma} \sin\left[\omega t - 2\pi\left(\frac{x_q - z_q \tan(\theta)}{\lambda}\right)\right],\end{aligned}\quad (13)$$

where γ is the fin ray slope.

From (13), we notice that the kinematics equation of biomimetic fin is relative to kinematic parameters such as frequency (f), amplitude (A), wavelength (λ), and morphological parameters such as the angle between fin ray and x_q axis (θ). By adjusting them, various locomotion patterns produced by real ray could also be achieved by biorobotic underwater propulsor. The kinematics analysis conducted above helps a lot in the following experiments investigation.

3.4. Propulsion Velocity Analysis of Robotic Ray. According to (13), we use the law of conservation of momentum to integrate water quality that is enveloped by the undulating fin.

If we consider $\theta = \pi/2$, at the beginning ($t \approx 0$), (13) can be simplified to

$$y_q(x_q, z_q, 0) = A(x_q, z_q) \sin(kx_q).\quad (14)$$

It is reasonable to ignore the bending deformation of fin rays during motion since their material is aluminum alloy. We safely assume that the fin amplitude is linear change. Thus the fluid mass between ΔABC and $\Delta A'B'C'$ (Figure 9) is expressed as

$$dM_{\text{water}} \frac{1}{2} |y_q| \sqrt{Lr^2 - y_q^2} dx_q,\quad (15)$$

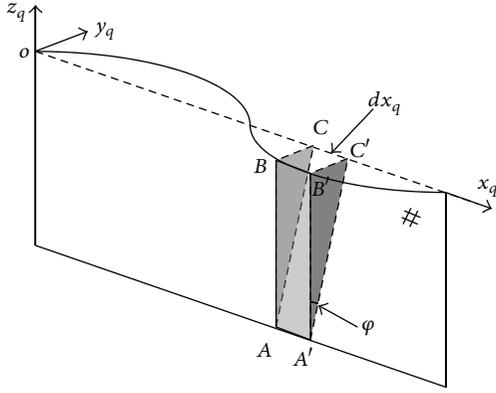


FIGURE 9: Propulsion velocity analysis of robotic ray.

where Lr is fin ray length, and the total fluid mass is calculated as below:

$$M_{\text{water}} = \frac{1}{2} \rho_{\text{water}} \int_0^{\tau\lambda} |y_q| \sqrt{Lr^2 - y_q^2} dx_q, \quad (16)$$

where ρ_{water} is density of fluid (e.g., water) and τ is natural number. Consider one full wavelength ($\tau = 1$), add (14) to (16), and then

$$\begin{aligned} M_{\text{water}} &= \frac{1}{2} \rho_{\text{water}} \int_0^{\lambda} |A \sin(kx_q)| \sqrt{Lr^2 - A^2 \sin^2(kx_q)} dx_q \\ &= \rho_{\text{water}} \int_0^{\lambda/2} A \sin(kx_q) \sqrt{Lr^2 - A^2 \sin^2(kx_q)} dx_q. \end{aligned} \quad (17)$$

From Figure 9 we have $\sqrt{Lr^2 - A^2 \sin^2(kx_q)} = Lr \cos(\varphi)$; then

$$M_{\text{water}} = \frac{2\rho_{\text{water}} ALr \cos(\varphi)}{k} \sin^2\left(\frac{\lambda k}{4}\right). \quad (18)$$

On the other hand, we have the following relations:

$$\mathbf{V} = \mathbf{V}_{\text{water}} + \mathbf{V}_{\text{fin}}. \quad (19)$$

Here \mathbf{V} represents the velocity of fluid (pushed by fin surface) relative to fin; $\mathbf{V}_{\text{water}}$ represents the velocity of fluid relative to earth coordinate system; \mathbf{V}_{fin} represents the velocity of fin relative to earth coordinate system. At the beginning, $\mathbf{V} \approx 0$, so the fluid drag on the fin surface can be ignored, using the law of conservation of momentum:

$$M_{\text{water}} \times \mathbf{V}_{\text{water}} = M_{\text{fin}} \times \mathbf{V}_{\text{fin}}. \quad (20)$$

Thus

$$\mathbf{V}_{\text{fin}} = \mathbf{V} - \mathbf{V}_{\text{water}} = \frac{M_{\text{water}}}{M_{\text{water}} + M_{\text{fin}}} \mathbf{V}, \quad (21)$$

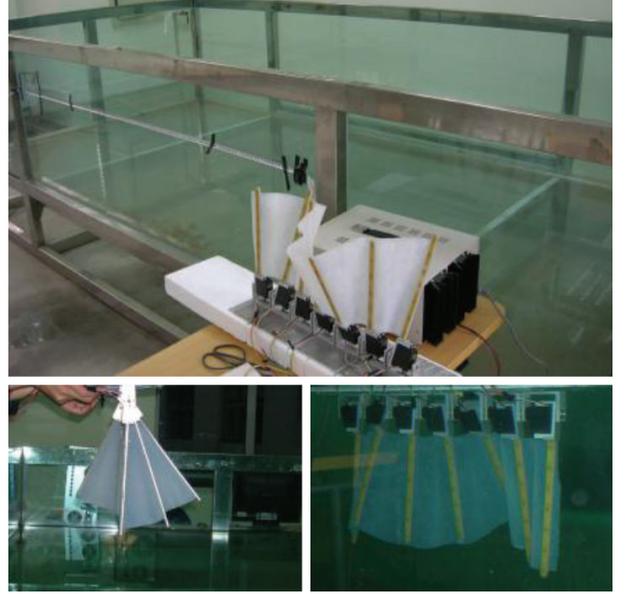


FIGURE 10: Water tank for free swimming test.

where $\mathbf{V} = \lambda f = \omega\lambda/2\pi$, and we finally have

$$\begin{aligned} \mathbf{V}_{\text{fin}} &= \frac{(2\rho_{\text{water}} ALr \cos(\varphi) / k) \sin^2(\lambda k/4)}{(2\rho_{\text{water}} ALr \cos(\varphi) / k) \sin^2(\lambda k/4) + M_{\text{fin}}} \left(\frac{\omega\lambda}{2\pi}\right) \\ &= \zeta \left(\frac{\omega\lambda}{2\pi}\right), \end{aligned} \quad (22)$$

where

$$\zeta = \frac{(2\rho_{\text{water}} ALr \cos(\varphi) / k) \sin^2(\lambda k/4)}{(2\rho_{\text{water}} ALr \cos(\varphi) / k) \sin^2(\lambda k/4) + M_{\text{fin}}}, \quad (23)$$

$(0 < \zeta < 1)$

where ζ means the influence of geometrical characteristic and amplitude on the propulsion velocity of robotic ray. Theoretically speaking, we conclude that \mathbf{V}_{fin} is directly proportional to ω and λ . What is more, \mathbf{V}_{fin} is in a certain proportion to A , Lr , and ρ_{water} and increases with them.

4. Experiments

The experimental systems for both free swimming and restricted forward straight swimming test of robotic fin are depicted below.

4.1. Experimental System for Free Swimming Test. Experimental system for free swimming test is shown in Figure 10. A high-speed video system (out of sight) (SpeedCAM) operates at 100 images per second with maximum resolution ratio of 512×512 pixels. The camera was placed above the flow tank to capture images in dorsal view. Images were obtained by the multichannel monochrome image acquisition card. Then, a computer operating system with human-computer interaction interface processed and analyzed images.

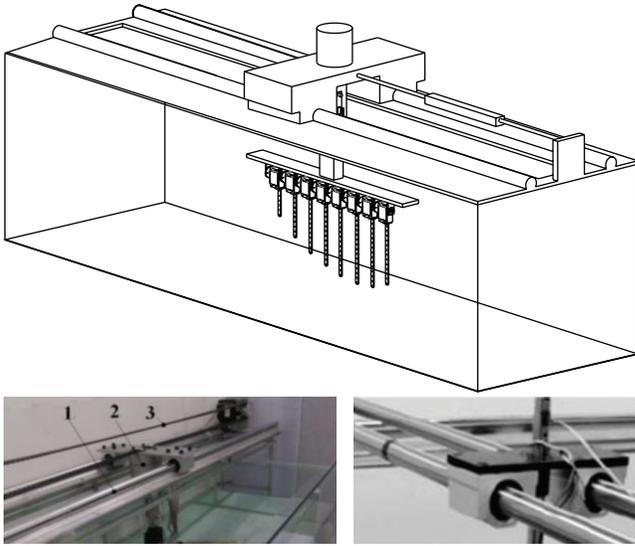


FIGURE 11: Experimental system for restricted forward straight swimming test. 1. Linear motion guide. 2. Fin clamp bracket. 3. Velocity sensor.

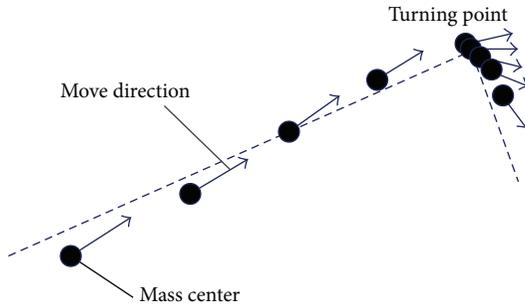


FIGURE 12: Trajectory tracking.

4.2. Experimental System for Restricted Forward Straight Swimming Test. Figure 11 is the experimental system for restricted forward straight swimming test. Most of current tests are conducted using this experimental system, such as the tests to investigate the influence of kinematic parameters and morphological parameters on averaged propulsion velocity. It consists of linear motion guide, fin clamp bracket, and velocity sensor. The robotic fin is fixed to the clamp bracket which can move along linear motion guide due to the undulating motion of robotic fin, and its motion velocity is detected and recorded by velocity sensor.

5. Results

5.1. Free Swimming Test. A single undulating fin was adopted to conduct the free swimming test in the water tank, including forward/backward and turning motion. By taking advantage of high speed camera, we tracked the motion trajectory of fin shown in Figure 12. The black spot represents mass center of fin, and the arrows mean its moving direction. When symmetrically oscillate fin ray, the fin surface receives

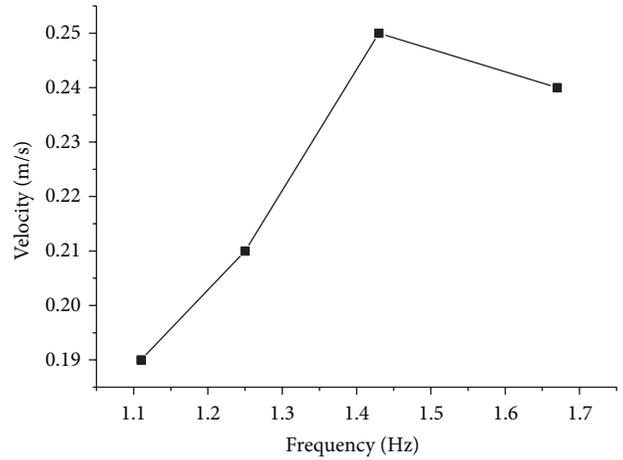


FIGURE 13: Relationship between averaged propulsion velocity and frequency.

symmetrical force from surrounding fluid and thus achieves straight moving. The turning motion of a single fin could be realized by bias-oscillating of fin ray.

5.2. Influence of Kinematic Parameters on Propulsion Velocity. We adopt four different frequencies (1.11 Hz, 1.25 Hz, 1.43 Hz, and 1.67 Hz), amplitudes (60 mm, 80 mm, 100 mm, and 120 mm) and wavelengths (256 mm, 322.5 mm, 387 mm, and 451.5 mm). The control variables method is used here and the results are shown below.

Figure 13 shows the relationship between averaged propulsion velocity and frequency. The increase of averaged propulsion velocity is almost directly proportional to that of frequency, which is consistent with the above theoretical calculation results. However, when frequency reaches a certain value (around 1.45 Hz in our test), the velocity decreases with the increase of frequency. The main reason may be that a higher frequency of the robotic ray may cause a more drastic disturbance to the surrounding fluid. Due to the limitation of experimental conditions, the disturbance cannot be fully extended and disappeared in the experimental tank. Furthermore, the reflected wave from the tank wall may further interfere with the performance of robotic ray, which causes the decrease of its velocity.

Figure 14 shows the relationship between averaged propulsion velocity and amplitudes. Similarly, the increase of averaged propulsion velocity is almost directly proportional to that of amplitudes, which is also consistent with the above theoretical calculation results.

Figure 15 shows the relationship between averaged propulsion velocity and wavelengths. At the beginning, the averaged propulsion velocity increases with the increase of wavelength, which is also consistent with the above theoretical calculation results. But a longer wavelength may flatten undulating curve. Due to the fluid viscosity, the velocity of fluid enveloped by the fin cannot reach the wave propagate velocity. Therefore, when wavelength reaches

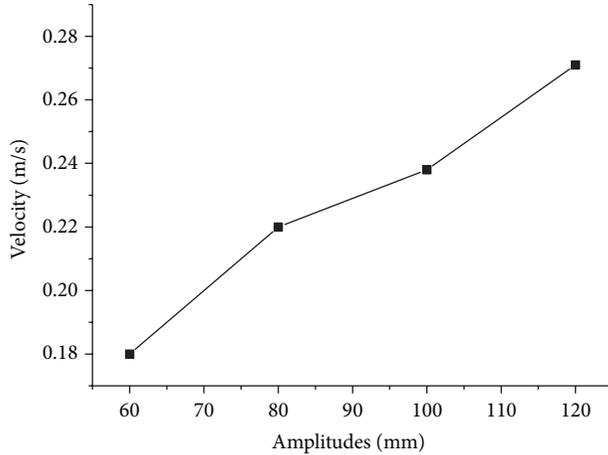


FIGURE 14: Relationship between averaged propulsion velocity and amplitudes.

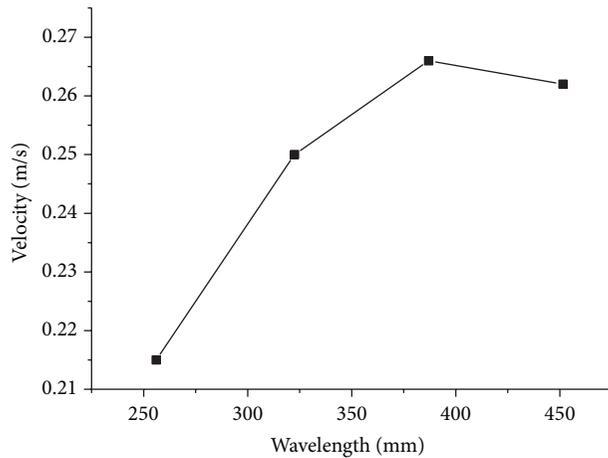


FIGURE 15: Relationship between averaged propulsion velocity and wavelengths.

a certain value (around 387 mm in our test), the velocity decreases with the further increase of wavelength.

5.3. Influence of Fin Undulating Patterns on Propulsion Velocity. Figure 16 shows the influence of different undulating patterns on averaged propulsion velocity of robotic fin. It is observed from Figure 16 that the amplitude envelope increases from the anterior part to the mid part and decreases toward the posterior (similar to real ray fin motion pattern) producing the highest velocity than the other three patterns. The averaged propulsion velocity of Mode 4 is about 0.271 m/s, while for Mode 1, Mode 2, and Mode 3 it is about 0.248 m/s, 0.238 m/s, and 0.182 m/s, respectively. Therefore, the fin in Mode 3 swims most slowly. It is also very interesting to highlight that seldom fishes in nature adopt this mode to propel themselves.

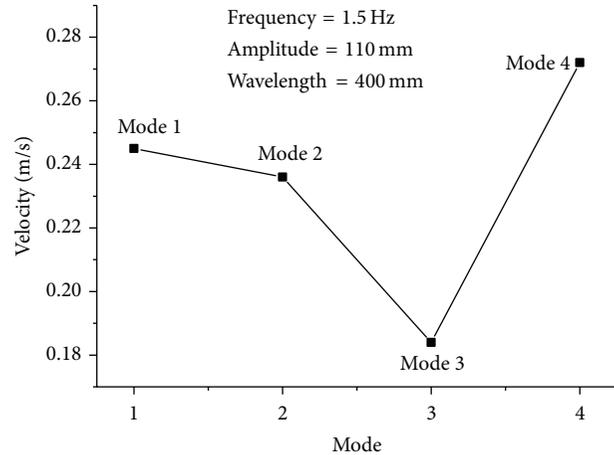


FIGURE 16: Influence of different undulating modes on averaged propulsion velocity. (a) The amplitude envelope is fairly constant along the fins (Mode 1). (b) The amplitude envelope gradually increased from the anterior part to the posterior (Mode 2). (c) The amplitude envelope decreases from the anterior part to the mid part and increases toward the posterior (Mode 3). (d) The amplitude envelope increases from the anterior part to the mid part and decreases toward the posterior (Mode 4).

5.4. Influence of Morphology Parameters on Propulsion Velocity

5.4.1. Fin Ray Angle. Three morphology parameters are considered: fin ray angle, fin shape, and fin aspect ratio. Figure 17 shows relationship between averaged propulsion velocity and fin ray angle. In the case of constant fin ray length (Figure 6(a)), the propulsion velocity increases with the increase of fin ray angle. It is not difficult to understand why it happens. When the fin ray angle increased, the fin surface area is growing accordingly in this case, the propulsion thrust may be subsequently increased which results in acceleration of propulsion velocity. In the case of constant fin area (Figure 6(b)), the propulsion velocity decreases with the increase of angle of fin ray; however, the tendency is reduced. The reason of which still confuses us and remains to be further investigated.

5.4.2. Fin Shape. Figure 18 is the relationship between averaged propulsion velocity and fin shape. As mentioned before, three typical fin shapes are discussed here: rectangle, triangle, and trapezoid. We find that the triangular fin is able to produce the highest propulsion velocity at 0.266 m/s. This fin morphology is observed in numerous batoid fishes.

5.4.3. Fin Aspect Ratio. We further select triangular fin to investigate influence of its aspect ratio on propulsion velocity at the same kinematic parameters. The aspect ratio is defined as the ratio of stretched length to chord length. Five values (0.8, 1.0, 1.2, 1.4, and 1.6) are easily achieved by change the robotic fin ray length as well as their spacing. The results are illustrated in Figure 19. With the increase of aspect ratio, the propulsion velocity is observably increased. We wonder

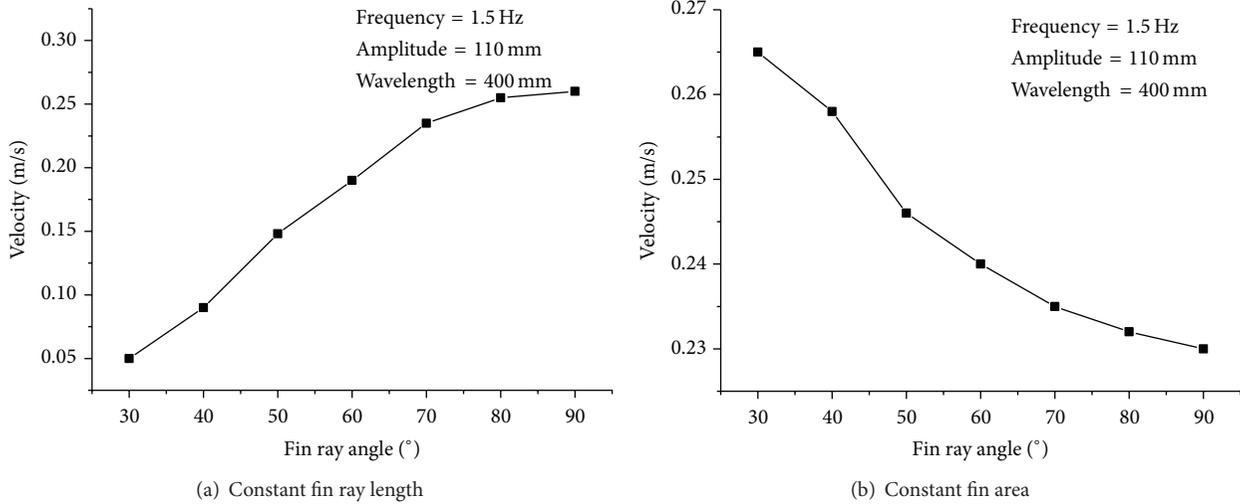


FIGURE 17: Relationship between averaged propulsion velocity and fin ray angle.

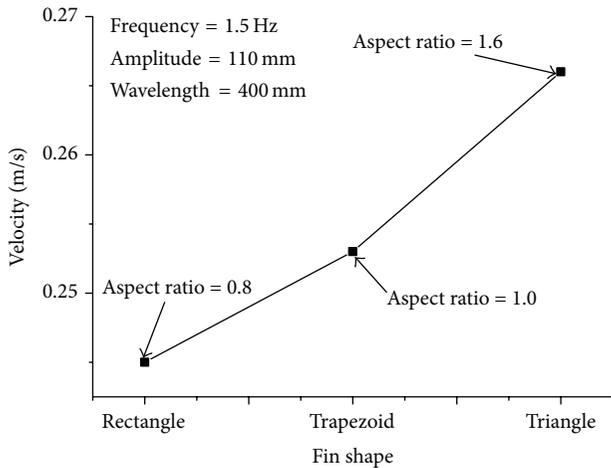


FIGURE 18: Relationship between averaged propulsion velocity and fin shape.

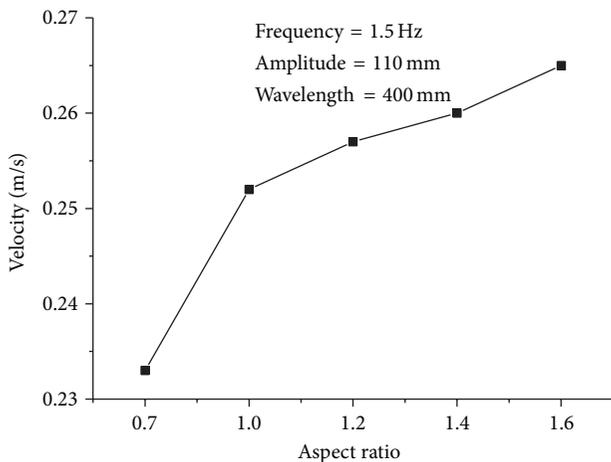


FIGURE 19: Relationship between averaged propulsion velocity and fin aspect ratio.

whether the change of propulsion velocity among different fin shapes may relate to the difference in aspect ratio.

6. Conclusions

In this paper, a robotic ray has been built based on the simplified pectoral structure of living bluespotted ray. The mechanical structure and control circuit are presented. Kinematic analysis on the fin ray and the full fin is discussed. The influence of various kinematic parameters and morphology parameters on the propulsion velocity of the robotic ray is analyzed using both theoretical and experimental methods. We finally conclude the following.

- (1) The averaged propulsion velocity of robotic ray is almost directly proportional to increase of frequency, amplitude, and wavelength.
- (2) The averaged propulsion velocity of robotic ray has a certain relationship with its morphological parameters such as fin shape. Meanwhile, in the case of constant fin ray length, the propulsion velocity increases with the increase of fin ray angle. In the case of constant fin area, the propulsion velocity decreases with the increase of angle of fin ray. As for the aspect ratio, the propulsion velocity is observably increased with the increase of it.
- (3) The amplitude envelope increases from the anterior part to the mid part and decreases toward the posterior (similar to real ray fin motion pattern) producing the highest velocity than the other three patterns. Combining with the fin shape, we may make a bold statement that the undulating pattern has a certain relationship with fin shape.

Our future work will focus on design, implement, and test control strategies involving two fins. Additionally, to analyze the efficiency and compare it quantitatively (a) with other undulation robots and (b) with other methods of propulsion

such as propellers is considered. Finally, the reason why the propulsion velocity decreases with the increase of angle of fin ray at the constant fin area remains to be further discussed. The relationship between motion pattern and fin shape will also be investigated later.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would first like to thank Professor JIA Laibing of University of Science and Technology of China for his support on the works over these years. The funds (2014YB01) and (2015YB02) supported by TZVTC (Taizhou Vocational and Technical College) are also acknowledged.

References

- [1] M. Sfakiotakis, D. M. Lane, and B. C. Davies, "An experimental undulating-fin device using the Parallel Bellows Actuator," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2356–2362, Seoul, Republic of Korea, May 2001.
- [2] L. J. Rosenberger and M. W. Westneat, "Functional morphology of undulatory pectoral fin locomotion in the stingray *Taeniura Lymma* (Chondrichthyes: Dasyatidae)," *The Journal of Experimental Biology*, vol. 202, no. 24, pp. 3523–3539, 1999.
- [3] L. J. Rosenberger, "Pectoral fin locomotion in batoid fishes: undulation versus oscillation," *The Journal of Experimental Biology*, vol. 204, no. 2, pp. 379–394, 2001.
- [4] M. Listak, G. Martin, D. Pugal, A. Aabloo, and M. Kruusmaa, "Design of a semiautonomous biomimetic underwater vehicle for environmental monitoring," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '05)*, pp. 9–14, Espoo, Finland, June 2005.
- [5] R. W. Blake, "On seahorse locomotion," *Journal of the Marine Biological Association of the United Kingdom*, vol. 56, no. 4, pp. 939–949, 1976.
- [6] R. W. Blake, "On ostraciiform locomotion," *Journal of the Marine Biological Association of the United Kingdom*, vol. 57, no. 4, pp. 1047–1055, 1977.
- [7] R. W. Blake, "On balistiform locomotion," *Journal of the Marine Biological Association of the United Kingdom*, vol. 58, no. 1, pp. 73–80, 1978.
- [8] 2013, <http://www.qing-yun.com>.
- [9] E. Blevins and G. V. Lauder, "Swimming near the substrate: a simple robotic model of stingray locomotion," *Bioinspiration and Biomimetics*, vol. 8, no. 1, Article ID 016005, 2013.
- [10] P. E. Sitorus, Y. Y. Nazaruddin, E. Leksono, and A. Budiyo, "Design and implementation of paired pectoral fins locomotion of *Labriiform* fish applied to a fish robot," *Journal of Bionic Engineering*, vol. 6, no. 1, pp. 37–45, 2009.
- [11] K. H. Low and A. Willy, "Biomimetic motion planning of an undulating robotic fish fin," *Journal of Vibration and Control*, vol. 12, no. 12, pp. 1337–1359, 2006.
- [12] P. V. Alvarado, S. Chin, W. Larson, A. Mazumdar, and K. Youcef-Toumi, "A soft body under-actuated approach to multi degree of freedom biomimetic robots: a stingray example," in *Proceedings of the 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechanics*, pp. 473–478, Tokyo, Japan, September 2010.
- [13] A. A. Shirgaonkar, O. M. Curet, N. A. Patankarm, and M. A. MacIver, "The hydrodynamics of ribbon-fin propulsion during impulsive motion," *The Journal of Experimental Biology*, vol. 211, no. 21, pp. 3490–3503, 2008.
- [14] F. Liu, C. Yang, Q. Su, D. Wang, and Y. Zhang, "Simulation analysis and experimental research on the movements of biomimetic fin," *Journal of Mechanical Engineering*, vol. 46, no. 19, pp. 24–29, 2010.
- [15] P. R. Bandyopadhyay, "Trends in biorobotic autonomous undersea vehicles," *IEEE Journal of Oceanic Engineering*, vol. 30, no. 1, pp. 109–139, 2005.
- [16] R. Boileau, L. Fan, and T. Moore, *Mechanization of Rajiform Swimming Motion: The Making of Robo-Ray*, University of British Columbia, 2002.
- [17] M. Epstein, J. E. Colgate, and M. A. MacIver, "A biologically inspired robotic ribbon fin," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Morphology, Control, and Passive Dynamics*, pp. 2412–2417, Edmonton, Canada, 2005.
- [18] T. Hu, L. Shen, L. Lin, and H. Xu, "Biological inspirations, kinematics modeling, mechanism design and experiments on an undulating robotic fin inspired by *Gymnarchus niloticus*," *Mechanism and Machine Theory*, vol. 44, no. 3, pp. 633–645, 2009.
- [19] R. P. Clark and A. J. Smits, "Thrust production and wake structure of a batoid-inspired oscillating fin," *Journal of Fluid Mechanics*, vol. 562, pp. 415–429, 2006.
- [20] P. L. Brower, *Design of a manta ray inspired underwater propulsive mechanism for long range, low power operation [Master of Science]*, Tufts University in Mechanical Engineering, 2006.
- [21] N. Kato, "Locomotion by mechanical pectoral fins," *Journal of Marine Science and Technology*, vol. 3, no. 3, pp. 113–121, 1998.
- [22] N. Kato, B. W. Wicaksono, and Y. Suzuki, "Development of biology-inspired autonomous underwater vehicle 'Bass III' with high maneuverability," in *Proceedings of the International Symposium on Underwater Technology (UT '00)*, pp. 84–89, Tokyo, Japan, May 2000.
- [23] K. H. Low, S. Prabu, J. Yang, S. W. Zhang, and Y. H. Zhang, "Design and initial testing of a single-motor-driven spatial pectoral fin mechanism," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '07)*, pp. 503–508, Harbin, China, August 2007.
- [24] P. R. Bandyopadhyay, *Biomimetics Research at NUWC Video (Edited)*, 14 Mts, Naval Undersea Warfare Center, Newport, RI, USA, 2000.
- [25] J. D. Geder, J. Palmisano, R. Ramamurti, W. C. Sandberg, and B. Ratna, "Fuzzy logic PID based control design and performance for a pectoral fin propelled unmanned underwater vehicle," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '08)*, pp. 40–46, Seoul, Republic of Korea, October 2008.
- [26] Y. R. Cai, S. S. Bi, and L. C. Zheng, "Design and experiments of a robotic fish imitating cow-nosed ray," *Journal of Bionic Engineering*, vol. 7, no. 2, pp. 120–126, 2010.

- [27] C. L. Zhou and K. H. Low, "Better endurance and load capacity: an improved design of manta ray robot (RoMan-II)," *Journal of Bionic Engineering*, vol. 7, pp. S137–S144, 2010.
- [28] S.-B. Yang, J. Qiu, and X.-Y. Han, "Kinematics modeling and experiments of pectoral oscillation propulsion robotic fish," *Journal of Bionic Engineering*, vol. 6, no. 2, pp. 174–179, 2009.
- [29] R. Hu, W. B. Wang, Q. L. Guo, H. Wang, and F. Cao, "Design of bionic robofish with flapping-win," *Journal of Shihezi University*, vol. 28, no. 1, pp. 109–112, 2010.
- [30] K. H. Low, "Modelling and parametric study of modular undulating fin rays for fish robots," *Mechanism and Machine Theory*, vol. 44, no. 3, pp. 615–632, 2009.
- [31] A. Punning, M. Anton, M. Kruusmaa, and A. Aabloo, "A biologically inspired ray-like underwater robot with electroactive polymer pectoral fins," in *IEEE Mechatronics and Robotics (MechRob '04)*, vol. 2, pp. 241–245, Aachen, Germany, 2004.
- [32] N. Kato, "Application of study on aqua bio-mechanisms to ocean engineering," *Kaiyo Kogaku Shinpojiumu*, vol. 16, pp. 131–138, 2001.
- [33] K. Takagi, M. Yamamura, Z.-W. Luo et al., "Development of a rajiform swimming robot using ionic polymer artificial muscles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 1861–1866, San Diego, Calif, USA, October 2006.
- [34] J. Tangorra, P. Anquetil, T. Fofonoff, A. Chen, M. Del Zio, and I. Hunter, "The application of conducting polymers to a biorobotic fin propulsor," *Bioinspiration and Biomimetics*, vol. 2, no. 2, pp. S6–S17, 2007.
- [35] Z. L. Wang, G. R. Hang, J. Li, Y. W. Wang, and K. Xiao, "A micro-robot fish with embedded SMA wire actuated flexible biomimetic fin," *Sensors and Actuators A: Physical*, vol. 144, no. 2, pp. 354–360, 2008.
- [36] Y. Wang, Z. Wang, J. Li, and G. Hang, "Development of a biomimetic manta ray robot fish actuated by shape memory alloy," *Robot*, vol. 32, no. 2, pp. 256–261, 2010.
- [37] K. H. Low and A. Willy, "Biomimetic motion planning of an undulating robotic fish fin," *Journal of Vibration and Control*, vol. 12, no. 12, pp. 1337–1359, 2006.
- [38] W. M. Kier and J. T. Thompson, "Muscle arrangement, function and specialization in recent coleoids," in *Proceedings of the 1st International Symposium: Coleoid Cephalopods through Time*, Berlin, Germany, September 2002.
- [39] Z. Chen, S. Shatara, and X. Tan, "Modeling of biomimetic robotic fish propelled by an ionic polymermetal composite caudal fin," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 3, pp. 448–459, 2010.
- [40] S. Guo, T. Fukuda, and K. Asaka, "A new type of fish-like underwater microrobot," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 1, pp. 136–141, 2003.
- [41] A. Azuma, *The Biokinetics of Flying and Swimming*, The American Institute of Aeronautics and Astronautics, Reston, Va, USA, 2nd edition, 2006.
- [42] G. V. Lauder, E. J. Anderson, J. Tangorra, and P. G. A. Madden, "Fish biorobotics: kinematics and hydrodynamics of self-propulsion," *The Journal of Experimental Biology*, vol. 210, no. 16, pp. 2767–2780, 2007.
- [43] K. H. Low, C. Zhou, and Y. Zhong, "Gait planning for steady swimming control of biomimetic fish robots," *Advanced Robotics*, vol. 23, no. 7-8, pp. 805–829, 2009.

Research Article

Action Selection and Operant Conditioning: A Neurobotic Implementation

André Cyr and Frédéric Thériault

Département d'Informatique, Université du Québec à Montréal (UQAM), Succursale Centre-Ville, Case Postale 8888, Montreal, QC, Canada H3C 3P8

Correspondence should be addressed to André Cyr; andre.cyr1@videotron.ca

Received 23 January 2015; Revised 7 April 2015; Accepted 14 May 2015

Academic Editor: Maki K. Habib

Copyright © 2015 A. Cyr and F. Thériault. 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.

Action selection (AS) is thought to represent the mechanism involved by natural agents when deciding what should be the next move or action. Is there a functional elementary core sustaining this cognitive process? Could we reproduce the mechanism with an artificial agent and more specifically in a neurobotic paradigm? Unsupervised autonomous robots may require a decision-making skill to evolve in the real world and the bioinspired approach is the avenue explored through this paper. We propose simulating an AS process by using a small spiking neural network (SNN) as the lower neural organisms, in order to control virtual and physical robots. We base our AS process on a simple central pattern generator (CPG), decision neurons, sensory neurons, and motor neurons as the main circuit components. As novelty, this study targets a specific operant conditioning (OC) context which is relevant in an AS process; choices do influence future sensory feedback. Using a simple adaptive scenario, we show the complementarity interaction of both phenomena. We also suggest that this AS kernel could be a fast track model to efficiently design complex SNN which include a growing number of input stimuli and motor outputs. Our results demonstrate that merging AS and OC brings flexibility to the behavior in generic dynamical situations.

1. Introduction

The vast topic of action selection (AS), including decision-making, behavioral choice, and behavior-switch as nomenclatures, is thoroughly explored from different perspectives of comprehension, levels of resolution, and scientific communities [1]. The AS biological phenomenon results from a neural process that leads to the observation of an agent doing one action over several others. The precise neural substrate underpinning this mechanism is not yet discovered [2, 3]. Even though many insights [4, 5] point toward how to simulate the AS natural process in artificial agents, there is no consensus to approach this cognitive phenomenon. In this view, the neurobotic domain aims to study AS from bioinspirations but applied for artificial intelligence (AI) and robotics purposes [6]. As a premise, building controllers for unsupervised autonomous robots necessarily requires a dedicated mechanism to operate behavioral transitions. Moreover, in real world, these actions should be adaptive instead of being ruled by fixed-patterns. Thus, a simulated

AS process should be flexible enough to cope with changing environments. These adaptive behaviors could come from learning functions which act as modulators of the AS process.

Artificial spiking neural networks (SNN) [7] have been successfully used as brain-controllers for robots, and several researches have proposed different computational models implementing AS through this specific experimental paradigm [8, 9]. A major aspect of SNN is to understand the information process at the level of a single spike [10]. Therefore, timing of spikes can be used for temporal event correlations and associative learning. As such, it could be interesting to study an AS mechanism in combination with an operant conditioning (OC) process, since we anticipate that these processes add more flexibility to switch behavior from their interactions, sharing both the ability of specifying actions.

The function of an AS process is to decide between different actions depending on the context. As a matter of fact, invertebrate neural organisms like *C. elegans* [11, 12], cnidarians [13], and fruit flies [14] do well in choosing among several

actions with only small circuits of neurons. Those include command-center neurons and central pattern generators (CPG) [15–18], which are well recognized for their intrinsic oscillation property. A deduction that could be drawn is that modeling an AS process does not necessarily require complexity as in high brain structures. Thus, the working hypothesis for the emulation of an AS process states that a simple mechanism should then be derived. In this paper, we use a basic CPG neural structure which helps in simulating an AS process containing sensory inputs, motor outputs, and decision neurons [19].

We propose to study the AS process within a SNN framework, targeting bioinspired robots controllers. Our first motivation is to combine AS and OC processes in a single neuro-robotic model. The main goal is to build a simple yet adaptive AS mechanism merged with the plasticity feature of an OC learning rule, while both occur under a dynamical scenario. A second objective is to develop a fast track method for implementing general AS processes into SNN. This research was driven by the fact that it is still a challenge to create a robot controller with the ability to learn from multiple sensory cues and actions in a SNN paradigm.

Theoretical Background. In neurosciences, the drive to accomplish a behavior emerges from a real-time dynamic of external sensory cues and internal values, where the different competitive neural signals ultimately orient the agent toward one preferred action. In a psychological view of the AS problem, serial processes occur from sensors to motors ending in a behavioral choice. According to the literature in computational cognitive science, the affordance competition hypothesis [20] argues that such a process is parallel and implies a prior specification of possible actions from ongoing sensory inputs. Specifically, when dynamical processes include several feedback loops in high neural structures and an attentional mechanism, the brain focuses on a specific winning action while continuously searching for other actions to do, depending on the context [21].

In the robotic domain, computational models of the AS process have been proposed (stochastic accumulator, linear-ballistic accumulator, and integrated accumulator models) [22, 23] as well as CPG in conjunction with SNN [24]. Since only a few studies in that field have investigated the AS process using SNN as bioinspired brain-controllers, our study takes another step in this direction. Therefore, our focus is on the close interaction between AS and the OC learning function, which we propose as a novelty in the domain.

Empirically, the angle of the AS problem was often to reach an optimal solution with a statistical approach [5, 25] or reproducing biological data. In our research, we wanted to consider the modulation factors that may influence the dynamic of an AS model from its interaction with a learning rule. In this perspective, a learning skill may improve a robot’s choice of actions to determine the future solutions. OC consists in one of these primary learning functions allowing cognitive agents to associate a feedback from their own actions. The natural OC process is well understood at the level of invertebrates [26]. Therefore, among others [27, 28], learning with OC represents one potential modifier of the AS

mechanism, perhaps allowing more flexibility from synaptic plasticity features in their adaptive behaviors. From its own past actions, which gave rewarding or punishing feedbacks, a robot may eventually pick up a different action, accelerating or decelerating the bias toward an oriented alternative.

We address these questions of the AS process combined with OC by evaluating a simple scenario in virtual and physical robots. This current work does not focus on extensive tests nor evaluates the overall computational impact of the parameters involved in the AS-OC models. It was also beyond the scope of this paper to challenge other AS approaches. Despite these limitations, we show a biologically plausible core base of these mechanisms in a neuro-robotic implementation. A benefit for robots to include the AS and OC critical processes is undeniable, since most physical robots are now able to perform a rich selection of actions that may be organized in hierarchical priorities, sequential fixed-patterns, competitive actions, and conflicting parallel behaviors.

In resume, we show an AS mechanism based on a CPG structure and few elementary neural units. This AS process was subject to modulation when merged with an OC learning rule. Together, these processes offer more flexibility to choose the best action under dynamical and variable contexts. Further demonstrations in more complex scenarios remain to be studied.

2. Methodology

2.1. The Spiking Neural Model. We propose a simple scenario to explore the AS and OC interrelation, explaining both processes in a neuro-robotic paradigm. The robot’s controller consists in artificial neural units connected by synapses. Our SNN model [29], similar to standard leaky integrate-and-fire neuron models, is based on a membrane potential variation, integrating nonlinearly, and temporally ongoing inputs through the SNN (1). In these neurons, when the membrane potential reaches a specific threshold, an all-or-none action potential is triggered. To start the CPG dynamic at the beginning of a simulation, a realistic neural property of endogenous pacemaker is implemented from adding a stronger leak (see (1) and starter neuron in the SNN). Consecutive to a spike emission, an electrical flux is sent, transformed at the synapse into a local excitatory or inhibitory synaptic postpotential current. This is then received at the targeted elements (2). The synapse is computationally modeled as a dynamical weight and is subject to be modulated from learning functions. The learning rule we used in this SNN is an adapted spike-timing dependent plasticity (STDP) [30–32]. The result of a STDP function is to increase a synaptic weight if the preneuron spikes before the postneuron unit, in a defined short time window. If the prespike arises after the postspike, then the inverse correlation leads to decreasing the synaptic weight (3).

Equation 1: Discrete-Time Neural Input Integration Function. Consider the following:

$$v_m(k) = f(v_m(k-1) + v_t(k)) + l. \quad (1)$$

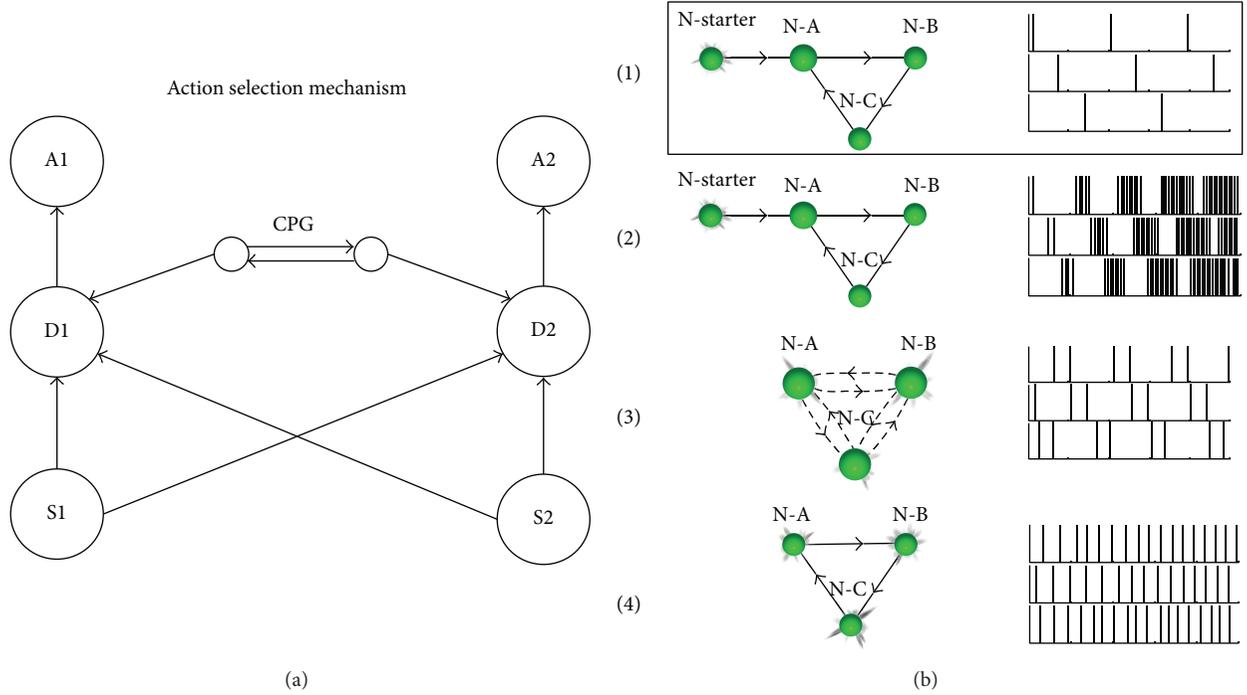


FIGURE 1: Dynamic of the AS mechanism and CPG topologies. (a) First, a stimulus is perceived from a sensory neuron. Then, the spikes emission from the sensory neuron targets all decision neurons, conserving a neutral aspect in the choice of possible actions to come. Finally, when the sensory input coincides with the CPG input at the decision neuron, the linked action will be performed (alternatively switching between actions). (b) In (1), the CPG represents the serial circular excitatory topology we used in this work. In (2), we show one option among several possibilities of tuning differently the internal neural parameters such as the postpotential state value, the refractory period, and the length of the neural curves. In (3), we show an all-to-all reciprocal inhibitory CPG circuit built with endogenous pacemaker neurons. In (4), we still keep the pacemaker neurons in the CPG but it is now linked with excitatory synapses. Overall, (b) (comparative spike bars in the same time scale on the right side) demonstrates the diversity of the output patterns reflected in different CPG, derived from small structural and functional changes.

$v_m(k)$ = membrane potential at cycle k , $v_t(k)$ = sum of the synaptic input as calculated in (2), f = ascending exponential function set between 0 and threshold (set as 65), and l = leak current for pacemaker property (set as 1).

Equation 2: General Alpha Function Representing the Postsynaptic Potential Curve. Consider

$$f(t) = ge^{-t/\pi}. \quad (2)$$

g = amplitude (set as 20), π = tau (set as 7), and t = time since spike (in cycle).

Equation 3: STDP Function Used. Consider the following:

$$\Delta w = \alpha_{t_{\text{post}} - t_{\text{pre}}} e^{(t_{\text{post}} - t_{\text{pre}})/\pi}. \quad (3)$$

Δw = synaptic weight change, $\alpha_{t_{\text{post}} - t_{\text{pre}}} = 1$ or -1 , depending on the sign of $t_{\text{post}} - t_{\text{pre}}$, and π = time constant.

STDP coefficient for Δw is as follows.

Maximum variation period = 3000 cycles.

Maximum synaptic change = 35%.

Maximum STDP time window = 25 cycles.

2.2. The AS Process. The elements in the AS mechanism consist of four basic groups of cells. The first group represents the decisions or command-neurons pointing to the action neurons which activates actuators (second group). The third group contains the sensory neurons providing contextual inputs linked to the decision neurons. Finally, the last group of cells contains the CPG neurons weakly connected to the decision neurons. The main function of the CPG is to provide a regular oscillation output pattern to bias one preferred decision neuron over the others. Notice that, a CPG neuron output is never allowed to trigger its linked decision neuron since the EPSP is too weak to reach the spike threshold. However, when pairing sensory and CPG inputs, only then can it reach its threshold and spike (see Figure 2). Therefore, the tuning of the parameters must overlap in relation of the CPG period and the sensory duration. In our experiments, a full CPG loop takes 90 cycles; hence, one CPG neuron spikes every 30 cycles. The sensory input duration last approximately 110 cycles. A second effect of the CPG is to disambiguate equal sensory inputs, a known conflicting problem difficult to resolve in the AI domain. Finally, CPG could also be understood as rhythmic internal values, feeding input in the AS process.

To graphically represent the AS process (Figure 1, left side), we show it in a complete generic scenario of two sensors

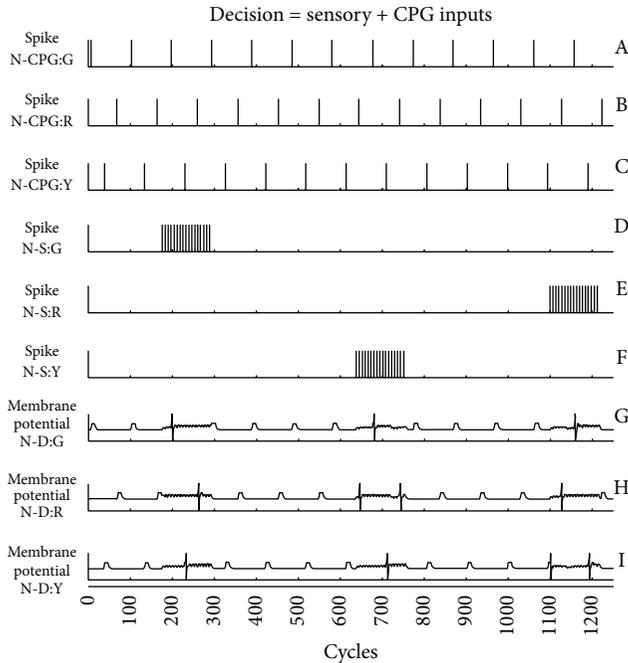


FIGURE 2: The figure shows the regular output spikes pattern from the serial circular excitatory CPG neurons (graphics A to C), which is the exact topology we used in our experiments. The mathematical neural dynamic behind these neurons is the same as the other spiking cells. When the CPG spike coincides with the sensory spikes (graphics D to F), the membrane potential of the decision neurons (graphics G to I) reaches the threshold for a spike emission (see the AS process in a complete SNN architecture 4).

and two actions. The SNN architecture is divided into three distinct layers: the sensory inputs, the internal integrative states, and the external action outputs. In Figure 3, the AS components are also clustered in a singular module within the generic but detailed SNN.

For the CPG’s kernel, we chose to embed the most regular and minimalist structure (see option 1 in the highlighted right side of Figure 1). The synaptic weights were all set to 100%, in order to have a continuous spike loop. To start the CPG, we used a biologically plausible endogenous pacemaker that shuts down just after initiating the dynamic. This starter option could be understood in terms of an internal value (i.e., low batteries, attentional process, and sensory-motor input) or could also be any other kind of triggers. As a result, the three neurons of the CPG are stimulated one after the other because of the circular serial excitatory connections 2.

2.3. The OC Learning Procedure. The cellular components included in the OC process consist in sensor neurons that provide the contextual inputs for Decision-to-Action neurons to generate the behaviors. Also, external reinforcer pointing to predictor neurons are also connected to Decision-to-Action neurons. Since sensor neurons are weakly linked to the predictor neurons but contain an STDP rule, the repetitive coincidence of the reinforcer (following the desire action) and the sensory input at the predictor neurons will increase

the synaptic weight. Therefore, sensory inputs will eventually trigger actions without any further needs of reinforcers [29].

2.4. The SNN Architecture. Specifically used for our results in a three-sensor and three-action context (Figure 4), the sensory neurons are composed of three color sensors (green, yellow, and red) in addition to one light sensor to perceive the rewarding light. The motor output neurons are represented by three LEDs (green, yellow, and red). Our AS process includes, as a modulating element, CPG placed at the intermediary neural layer. It contains three neurons paired with the same number of the possible actions. The proposed CPG kernel consists in excitatory neurons organized in a serial circular topology.

Each CPG neuron in the CPG network is connected to its own decision neuron with a small synaptic weight. One target of a decision neuron is its connected action neuron with a strong synaptic weight between these units; when a decision neuron spikes, the linked action neuron spikes as well. Each decision neuron is also weakly connected to its own predictor neuron for the learning context interrelation. In this experiment, the predictor neurons target their output to all other decision neurons with inhibitory strong synaptic links. Therefore, when the sequence sensor-action-reward is learned from a precise predictor neuron, it will shut down all other possible actions. This arbitrator mechanism could be understood as a type of neural competition. Initial synaptic values used in our SNN were manually tuned and can be retrieved in Table 1.

2.5. The Task and the Actions. In the virtual experiment (Figure 5), the SNN is implemented in a static robot. The robot’s task consists in learning to match colors between its three possible actions of LED emission (green, yellow, and red) and the color blocks perception. Our 3D simulation software environment (SIMCOG-NeuroSim, AI-Future) allows three different color blocks (green, yellow, and red) to move continuously, at a constant speed, in a clockwise circular trajectory, passing one at a time just in front of the robot. The time frame perceptive contact enables the robot to, at least, produce one different action for each block over 1000 cycles. In the first part of the experiment (0–10000 cycles), a rewarding light (not shown) triggers only when the color LED action matches the block of the same color. At cycle 10500, the robot was moved temporarily for 3000 cycles (cycle 10500–13500) to a location where no sensory input is received, allowing a forgetting factor to operate and reset the synaptic weights. Then, the robot was replaced to its initial position for another round. However, in this part, the rewarding light follows only when the LED emission is on the next color block. The purpose of this part is to show how efficient the AS and OC dynamics can modify the behavior, since these novel learning associations are achieved in a single trial.

As a proof of concept and endpoint in the robotic domain, we reproduced the virtual setup in a physical experiment (Figure 6). The SNN is totally identical and we simply transferred it into the physical robot without any further adjustments. For simplicity, we chose the EV3 Lego Mindstorm (Lego Inc.) as physical platform. The main processor is an

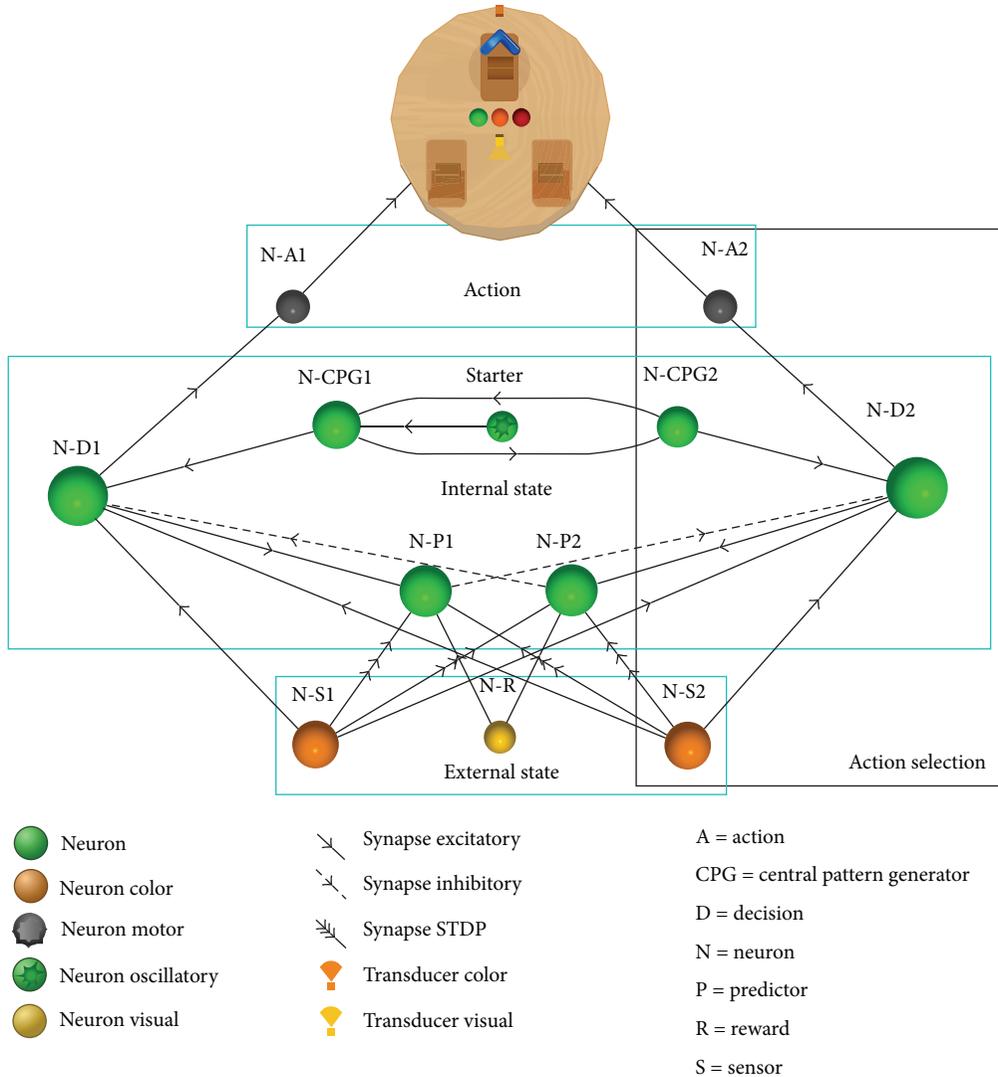


FIGURE 3: Generic SNN of AS and OC in a two-sensor and two-motor configuration.

ARM9 core clocking at 300 MHz and it contains 64 MB RAM. The LEDs are similar to the virtual scenario except that there are only two colors available, the green and the red. When the two of them are opened at the same time, the resulting color is orange, hence having our third color for the experimentation. A light sensor is also used to read the external rewarding light, which was synchronized and delivered from a Raspberry Pi board, just after a desired action is done by the robot. A NXT Lego Mindstorm controller (Lego Inc.), mounted on a shaft, controls the rotation of three color bricks (green, orange, and red) using one attached motor. A slow stepwise speed was set with no possibility of modulation from the robot. In this configuration, the bricks pass just in front of the color sensor. When a sensor catches a color block, the numerical value is converted in an artificial electrical current with an adapted scaling factor for the SNN. Only the first learning part was done for the demonstration. Supplementary material is available at <https://www.youtube.com/watch?v=8MXA4wxJSpE> and consists of a video of the experiment.

3. Results

The results from the virtual experiment were obtained in a single trial. The following graphic data will refer to Figure 4 for the SNN architecture and Table 1 for its associated synaptic weight matrix. In Figure 7, we can observe at the beginning of the simulation that when the Green Sensor neuron (N-S:G) spikes (black bars in graphic A), the robot tries alternative actions of lighting up each LED (graphic B: green, graphic G: yellow, and graphic L: red). Since there was no reward for any actions triggered prior to cycle 300, no learning from the STDP rule was observed at the synapse going from the sensor to the predictor neuron (D, I, and N). At around cycle 500, a first yellow block is perceived from the sensor yellow (F), while the CPG continuously provides alternative actions of LED emissions. Specifically, with the lighting up of the yellow LED (G), and with the following light reward (not shown), the associated predictor neuron spikes (H). Consequently, a positive association between the Yellow

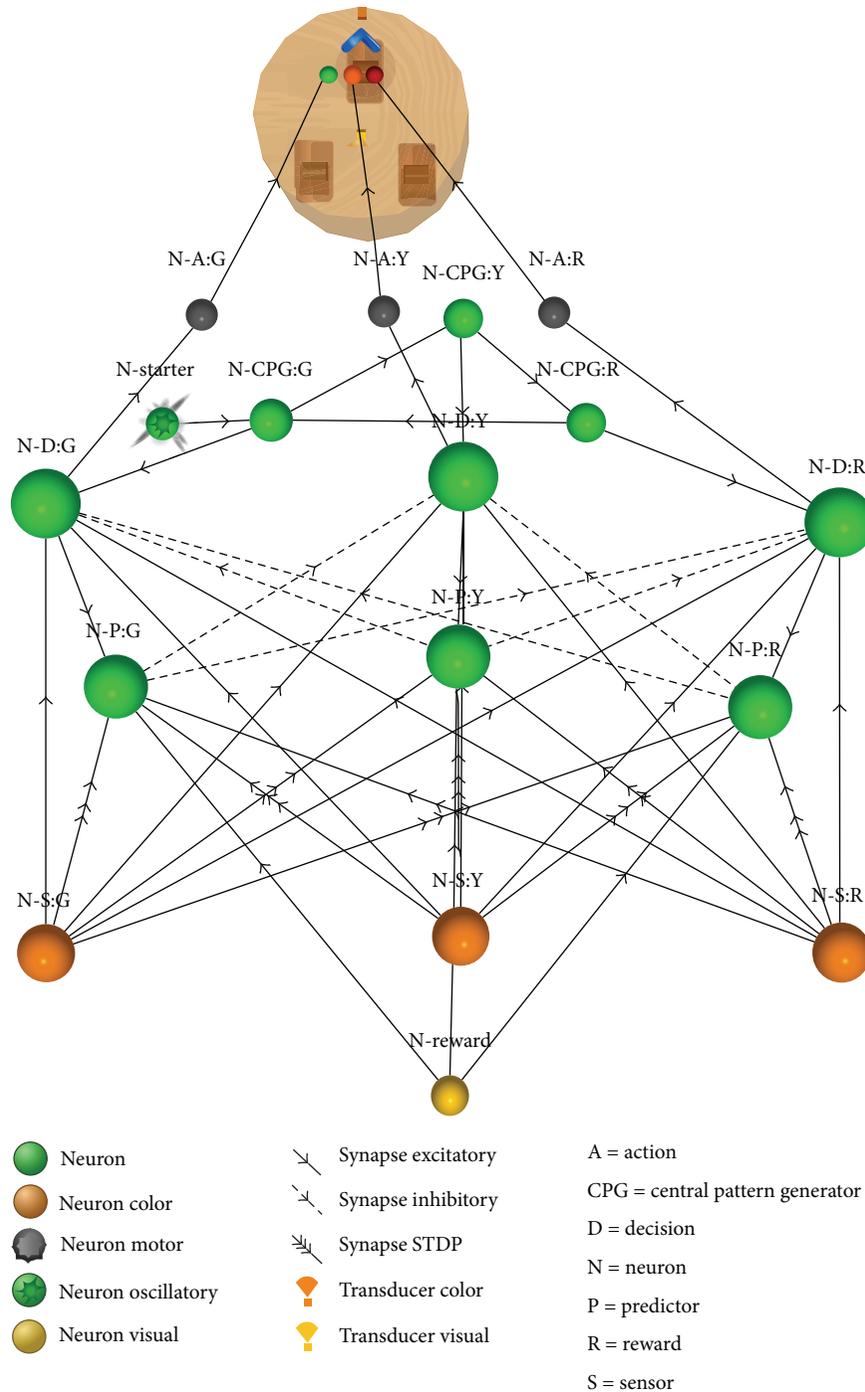


FIGURE 4: SNN architecture used for our experiments, demonstrating the interrelation of the AS process and the OC learning rule in a three-sensor and three-motor configuration. See also Figure 3: A = action, CPG = central pattern generator, D = decision, G = green, N = neuron, P = predictor, S = synapse, R = red, Y = yellow, and STDP = three consecutive arrows on a synapse.

Sensor neuron and this predictor neuron starts to increase the STDP coefficient (I). This affects the synaptic weight to a bound limit when several associations occurred, stabilizing at around cycle 4000. The role of the predictor neuron (H) in this SNN is to inhibit the other decision neurons and their connected action neurons (B, L). At around cycle 8500, one

can see that the robot has fully learned the three sensory-motor contexts by pairing the good LED action with the good perceived color block. Since the period of the CPG neurons and the rotation of the color blocks did not fit perfectly, the learning time frame for each sensory-motor pair is not identical.

TABLE 1: Synaptic weights associated with the SNN of Figure 4. These are represented by numerical percentage values. Delays in cycle as well as the type of the synapse are also mentioned.

Synaptic weight table			
Name	Strength	Delay	Type
S-A:G/E:G	5	0	Excitatory
S-A:R/E:R	5	0	Excitatory
S-A:Y/E:Y	5	0	Excitatory
S-CPG:G/CPG:Y	100	30	Excitatory
S-CPG:G/D:G	30	0	Excitatory
S-CPG:R/CPG:G	100	30	Excitatory
S-CPG:R/D:R	30	0	Excitatory
S-CPG:Y/CPG:R	100	30	Excitatory
S-CPG:Y/D:Y	30	0	Excitatory
S-D:G/A:G	100	0	Excitatory
S-D:G/P:G	15	0	Excitatory
S-D:R/A:R	100	0	Excitatory
S-D:R/P:R	15	0	Excitatory
S-D:Y/A:Y	100	0	Excitatory
S-D:Y/P:Y	15	0	Excitatory
S-Reward/P:G	8	0	Excitatory
S-Reward/P:R	8	0	Excitatory
S-Reward/P:Y	8	0	Excitatory
S-S:G/D:G	15	0	Excitatory
S-S:G/D:R	15	0	Excitatory
S-S:G/D:Y	15	0	Excitatory
S-S:G/P:G	15	0	Excitatory
S-S:G/P:R	15	0	Excitatory
S-S:G/P:Y	15	0	Excitatory
S-S:R/D:G	15	0	Excitatory
S-S:R/D:R	15	0	Excitatory
S-S:R/D:Y	15	0	Excitatory
S-S:R/P:G	15	0	Excitatory
S-S:R/P:R	15	0	Excitatory
S-S:R/P:Y	15	0	Excitatory
S-S:Y/D:G	15	0	Excitatory
S-S:Y/D:R	15	0	Excitatory
S-S:Y/D:Y	15	0	Excitatory
S-S:Y/P:G	15	0	Excitatory
S-S:Y/P:R	15	0	Excitatory
S-S:Y/P:Y	15	0	Excitatory
S-Starter/CPG:G	50	0	Excitatory
S-P:G/D:R	10	0	Inhibitory
S-P:G/D:Y	10	0	Inhibitory
S-P:R/D:G	10	0	Inhibitory
S-P:R/D:Y	10	0	Inhibitory
S-P:Y/D:G	10	0	Inhibitory
S-P:Y/D:R	10	0	Inhibitory

Action selection and operant conditioning

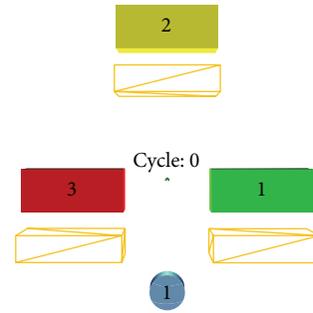


FIGURE 5: Snapshot of the 3D environment including the robot and three color blocks rotating constantly clockwise and passing just in front of the color sensor of the robot.

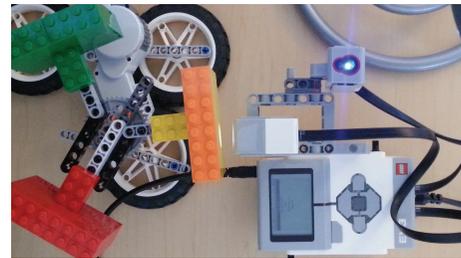


FIGURE 6: Picture of the EV3 brick Lego Mindstorm and the rotative color blocks setup. The color sensor points to the blocks and the light sensor points up.

Between cycle 10500 and 13500, we changed the robot's location to avoid perception of the color blocks. This was done in order to allow the SNN to reset the synaptic weights to their initial values (using a forgetting parameter present in the STDP rule). This was optional, and learning forever would be the result scenario if the feature was not active. Unlearning could also be obtained from inverting the temporal sequence of the sensor, action, and reward. If there is no correlation anymore, the STDP rule will progressively decrease the synaptic weight. In another simulation setup, a punishment (inhibition) could also serve as a fast negative modulation factor of the synaptic weights.

The last part (>13500 cycle) of Figure 7 demonstrates the online adaptive behavior aspect of the SNN embedding OC and AS. One can observe that the robot must choose a different action in order to receive the reward. In this case, lighting up a green LED on a yellow block, a yellow LED on a red block, and a red LED on a green block triggers the reward. The corresponding STDP factors (E, J, and N) match these three learning sets.

As for our physical simulation, the architecture was not modified in any way, except for the binding of logical sensors and motors to the robot. Figure 8 shows the results of the simulation, which were obtained in a single trial. They show approximately the same data, with more or less precision and small artifacts. This is due to the fact that it is indeed much easier to configure variables and the context of virtual environments than it is in the real world.

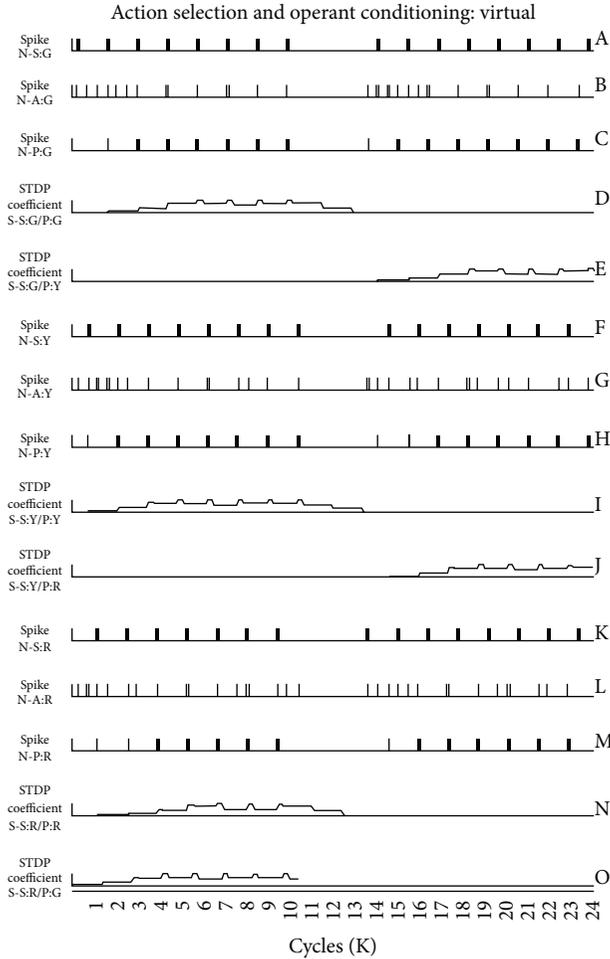


FIGURE 7: Graphics of the neural spikes and the STDP rule factors occurring over 24000 cycles. The first five (graphics A to E) concern the green block and green LED. The middle five capture the logic of the yellow block and the yellow LED. The last group of five is associated with the red color. Around cycle 8500, the robot has learned to perform the right action with the appropriate cue. This means that, upon triggering an action, if the predictor spikes from a following reward, the STDP coefficient will increase, boosting the synaptic weight between the sensor and the predictor. Once this synaptic weight reaches a critical threshold, the sensor input will trigger a spike to the reinforced predictor neuron, without the need of the reward.

4. Discussion

In this paper, we explored the AS process through a neuro-robotic perspective. Since this general mechanism directly involves actions, we demonstrated the phenomenon in the context of OC procedures which also imply a selection of actions from reinforcer. Our main objective was to study the benefit effects of merging this learning rule with an AS process. A second concern was to provide a fast track solution to efficiently design more complex SNN used as brain's controller for virtual and physical robots that include several motor outputs. We propose a basic CPG motif as one

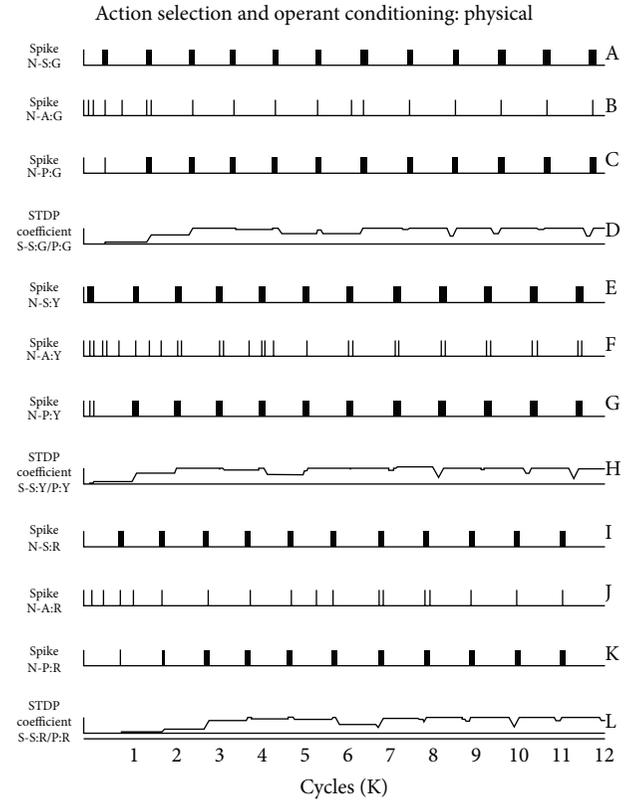


FIGURE 8: Graphics of the neural spikes and the STDP rule factors occurring over 12000 cycles.

key component of an AS process, in order to neutrally switch between its available actions. With the CPG structure used in relation with a sensory input context, a decision neuron gets all the information needed to bias toward one preferred action. We also showed that the OC learning function influences the AS process, conferring supplementary adaptive behaviors from synaptic plasticity.

We chose a simple CPG topology as one component of the AS mechanism. Other CPG configurations are possible [33], including those built with reciprocal inhibitory synaptic links and endogenous pacemaker neurons, though the analytical issues are more complex to track and predict. Tuning the parameters (i.e., postpotential spike value, threshold) of individual neuron differently could also influence the rhythm, affecting the CPG network by increasing or decreasing their output periods. After several options, we found that a serial excitatory circular CPG motif is a good trade-off between simplicity and benefits. In our AS model, without any other synaptic feed, this CPG configuration will spike one unit after the other, indefinitely and at a constant rate. We showed the AS model in a generic two-two example and in a specific three-three sensory input and motor output configurations. Adding more sensors and actions will necessary require other neurons in the CPG network, though their numbers are linearly related to their attached decision and action neurons, acting as a premotor structures. In this case, hierarchical

groups of CPG/actions could also replace the serial circular topology, possibly avoiding useless spikes or triggering other networks. Allowing different combinations and compositions of CPG units also dramatically increase the behavior possibilities, without considering a one-to-one CPG-action, though it was not explored in this paper.

Adding the decision neurons (equivalent to command-neuron in invertebrates) into the AS model allowed flexibility, regarding several contextual input sources. The CPG units bring the decision neurons membrane potential to a subthreshold firing level. Since the CPG period is fixed, the speed-accuracy trade-off (SAT) of the decision-making [34] result is fast and accurate. Unfortunately, but no adjustment is possible, a major point to consider when modeling an AS process. Sometimes, cognitive agents must take decisions quickly while in other conditions, it is necessary to take the time to compute the best decision. According to a recent hypothesis [35], flexibility of the SAT's response variables depends on adjusting the baseline firing rate, the sensory gain and noise inputs, the firing threshold, and related bound parameters of the receiving neurons [25, 36].

Having those various AS modulator factors in mind, designing complex SNN with several populations of neurons including heterogeneous individual neural parameters values is possible, but highly complex to tune properly. In this perspective, progressively integrating stronger/lower and faster/slower CPG inputs could add discriminative and flexible response advantages as well as offering more realistic behavioral features of the AS mechanism. A computational challenge in an AS model within the neurobotic field is to allow the SNN to dynamically change all these initial fixed parameters values, conferring considerable adaptive properties at the level of the cognitive agent. In this vein, perhaps a question remains about the SAT: what are the variables biases in the AS process when there is no emergency to choose one action?

In our experiments, reversing the rotation order for the block to counter clockwise, accelerating or decelerating the speed of rotation as well as mixing the color order do not change the qualitative aspect of the learning curve. However, the temporal relation between the perception time contact of the sensor inputs and the timing of the CPG influences the number of occurrences of these associations and, thus, the length of time needed to learn. In any case, the EPSP timing between the CPG and the sensory neurons is of major importance and needs a full coherence into the whole dynamic system. In this perspective, the physical experiment shed some light on the temporal robustness of our AS and OC models, justifying its inclusion in the study. Without changing any parameters in the SNN, the EV3 robot was able to learn very well how to receive the reward when achieving the good action, even if the rotation of the blocks was irregular due to the imprecision of the material.

We explained how the OC learning rule modulates the AS process in a SNN paradigm. The sensory-motor context does influence the decision to do one action over others. These decisions were not just built-in reflexes. Moreover, the behavioral plasticity was observed even if the CPG dynamic was fixed. At this point, some interesting variances could be to add

other learning rules that are not limited by an OC procedure. As such, integrating nonassociative (habituation) and other associative learning (classical conditioning) functions could complete the design of an AS model, but it was beyond the scope of this paper.

One concern we avoid in this paper is the attentional problem. We chose to ignore this major cognitive component, mostly because of the current lack of neural mechanism and theory when applied in lower neural system organisms. We understand that basal ganglion or any subcortical or cortical implications in the AS process seem to be relevant structures in higher biological neural systems such as in humans or primates. However, our present perspective on the AS problem lies in the AI neurobotic domain, which is still beyond the reach of lower cognitive natural species. Our strategy aims to engineer a bioinspired minimalist solution from emulating simple neural organisms such as in *C. elegans*, which selects its actions without involving huge structures. Instead, command-neurons and CPG neurons are common cellular elements found in the primitive invertebrate neural circuits. No doubt, complex neural layers may extend and add profitable values in the simulation of an AS process but should not be a necessary requirement to achieve a basic one. These evolutionary concerns may eventually find an echo in a multiplicity and hierarchical AS mechanisms.

The AS scope obtained from these results is theoretically not limited to only a few simple actions or unimodal single sensory stimuli. The generic aspect of the AS process comes from the parsimonious components and parameters inside the kernel. The simplicity of this AS module already allows to be adaptive from a 2×2 to a 3×3 sensors-actions scenario without much changes in the SNN architecture. In these two scenarios, as long as there are the same numbers of possible ending actions and sensors as input, the AS core process will operate and tune the same. Therefore, building more complex SNN including several actions should be anticipated as faster and easier, though it remains to be proven in other situations. In that sense, we currently work on a shaping behavior learning technique based on the AS process, while simulating an indoor dynamical navigation task with several possible behaviors. This is an example that can demonstrate how this bioinspired AS process could help in concrete application in the robotic field.

5. Conclusion

This paper showed an AS process made from simple cellular elements. It is based on CPG and sensory neurons which influence decision neurons in their choice to generate a behavior from the action neurons. We demonstrated this basic AS mechanism in an OC learning context that allows behavioral flexibility from their mutual influences. The experiments were conducted under a biologically inspired paradigm, specifically with a SNN acting as brain-controller for virtual and physical robots. In addition, the simplicity and the generic aspect of our AS model may provide a fast track solution to build more complex SNN, including multiple actions in different dynamic scenarios.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] A. K. Seth, T. J. Prescott, and J. J. Bryson, *Modelling Natural Action Selection*, Cambridge University Press, Cambridge, UK, 2011.
- [2] M. D. Humphries, K. Gurney, and T. J. Prescott, "Is there a brainstem substrate for action selection?" *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 362, no. 1485, pp. 1627–1639, 2007.
- [3] J. I. Gold and M. N. Shadlen, "The neural basis of decision making," *Annual Review of Neuroscience*, vol. 30, pp. 535–574, 2007.
- [4] M. Khamassi, L. Lachèze, B. Girard, A. Berthoz, and A. Guillot, "Actor-critic models of reinforcement learning in the basal ganglia: from natural to artificial rats," *Adaptive Behavior*, vol. 13, no. 2, pp. 131–148, 2005.
- [5] N. F. Lepora, C. W. Fox, M. H. Evans, M. E. Diamond, K. Gurney, and T. J. Prescott, "Optimal decision-making in mammals: insights from a robot study of rodent texture discrimination," *Journal of the Royal Society Interface*, vol. 9, no. 72, pp. 1517–1528, 2012.
- [6] T. J. Prescott, F. M. M. González, K. Gurney, M. D. Humphries, and P. Redgrave, "A robot model of the basal ganglia: behavior and intrinsic processing," *Neural Networks*, vol. 19, no. 1, pp. 31–61, 2006.
- [7] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
- [8] T. C. Stewart, T. Bekolay, and C. Eliasmith, "Learning to select actions with spiking neurons in the basal ganglia," *Frontiers in Neuroscience*, vol. 6, article 2, 2012.
- [9] S. Skorheim, P. Lonjers, and M. Bazhenov, "A spiking network model of decision making employing rewarded STDP," *PLoS ONE*, vol. 9, no. 3, Article ID e90821, 2014.
- [10] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 2003.
- [11] S. Faumont, T. H. Lindsay, and S. R. Lockery, "Neuronal microcircuits for decision making in *C. elegans*," *Current Opinion in Neurobiology*, vol. 22, no. 4, pp. 580–591, 2012.
- [12] M. Zhang, W. R. Schafer, and R. Breitling, "A circuit model of the temporal pattern generator of *Caenorhabditis* egg-laying behavior," *BMC Systems Biology*, vol. 4, no. 1, article 81, 2010.
- [13] A. L. Stöckl, R. Petie, and D.-E. Nilsson, "Setting the pace: new insights into central pattern generator interactions in box jellyfish swimming," *PLoS ONE*, vol. 6, no. 11, Article ID e27201, 2011.
- [14] C. R. von Reyn, P. Breads, M. Y. Peek et al., "A spike-timing mechanism for action selection," *Nature Neuroscience*, 2014.
- [15] S. Grillner, "Neurobiological bases of rhythmic motor acts in vertebrates," *Science*, vol. 228, no. 4696, pp. 143–149, 1985.
- [16] K. Matsuoka, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biological Cybernetics*, vol. 56, no. 5-6, pp. 345–353, 1987.
- [17] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.
- [18] M. Heisenberg, "Action selection," in *Invertebrate Learning and Memory*, R. Menzel and P. Benjamin, Eds., vol. 22, Academic Press, 2013.
- [19] P. Redgrave, T. J. Prescott, and K. Gurney, "The basal ganglia: a vertebrate solution to the selection problem?" *Neuroscience*, vol. 89, no. 4, pp. 1009–1023, 1999.
- [20] P. Cisek, "Making decisions through a distributed consensus," *Current opinion in neurobiology*, vol. 22, no. 6, pp. 927–936, 2012.
- [21] P. Cisek, G. A. Puskas, and S. El-Murr, "Decisions in changing conditions: the urgency-gating model," *The Journal of Neuroscience*, vol. 29, no. 37, pp. 11560–11571, 2009.
- [22] S. D. Brown and A. Heathcote, "The simplest complete model of choice response time: linear ballistic accumulation," *Cognitive Psychology*, vol. 57, no. 3, pp. 153–178, 2008.
- [23] M. Usher and J. L. McClelland, "The time course of perceptual choice: the leaky, competing accumulator model," *Psychological Review*, vol. 108, no. 3, pp. 550–592, 2001.
- [24] M. Kassim, N. Zainal, and M. Arshad, "Central pattern generator in bio-inspired simulation using matlab," in *Proceedings of the MEDINFO*, vol. 98, 1998.
- [25] D. Standage, D. Wang, and G. Blohm, "Neural dynamics implement a flexible decision bound with a fixed firing rate for choice: a model-based hypothesis," *Frontiers in Neuroscience*, vol. 8, article 318, 2014.
- [26] B. Brembs, F. D. Lorenzetti, F. D. Reyes, D. A. Baxter, and J. H. Byrne, "Operant reward learning in *Aplysia*: neuronal correlates and mechanisms," *Science*, vol. 296, no. 5573, pp. 1706–1709, 2002.
- [27] K. M. Crisp and K. A. Mesce, "Beyond the central pattern generator: amine modulation of decision-making neural pathways descending from the brain of the medicinal leech," *Journal of Experimental Biology*, vol. 209, no. 9, pp. 1746–1756, 2006.
- [28] T. M. Wright Jr. and R. L. Calabrese, "Patterns of presynaptic activity and synaptic strength interact to produce motor output," *The Journal of Neuroscience*, vol. 31, no. 48, pp. 17555–17571, 2011.
- [29] A. Cyr, M. Boukadoum, and F. Thériault, "Operant conditioning: a minimal components requirement in artificial spiking neurons designed for bio-inspired robot's controller," *Frontiers in Neurobotics*, vol. 8, article 21, 2014.
- [30] N. Caporale and Y. Dan, "Spike timing-dependent plasticity: a Hebbian learning rule," *Annual Review of Neuroscience*, vol. 31, pp. 25–46, 2008.
- [31] A. Cyr and M. Boukadoum, "Classical conditioning in different temporal constraints: an STDP learning rule for robots controlled by spiking neural networks," *Adaptive Behavior*, vol. 20, no. 4, pp. 257–272, 2012.
- [32] H. Markram, W. Gerstner, and P. J. Sjöström, "A history of spike-timing-dependent plasticity," *Frontiers in Synaptic Neuroscience*, vol. 3, article 4, 2011.
- [33] A. I. Selverston, "Invertebrate central pattern generator circuits," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 365, no. 1551, pp. 2329–2345, 2010.
- [34] P. Cassey, A. Heathcote, S. D. Brown, and O. Sporns, "Brain and behavior in decision-making," *PLoS Computational Biology*, vol. 10, no. 7, Article ID e1003700, 2014.
- [35] R. Bogacz, E.-J. Wagenmakers, B. U. Forstmann, and S. Nieuwenhuis, "The neural basis of the speed-accuracy tradeoff," *Trends in Neurosciences*, vol. 33, no. 1, pp. 10–16, 2010.
- [36] R. P. Heitz and J. D. Schall, "Neural mechanisms of speed-accuracy tradeoff," *Neuron*, vol. 76, no. 3, pp. 616–628, 2012.

Research Article

Bioinspired Tracking Control of High Speed Nonholonomic Ground Vehicles

Adam Shoemaker and Alexander Leonessa

Center for Dynamic Systems Modeling and Control, Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24061, USA

Correspondence should be addressed to Adam Shoemaker; shoe@vt.edu

Received 5 January 2015; Revised 19 May 2015; Accepted 24 June 2015

Academic Editor: Maki K. Habib

Copyright © 2015 A. Shoemaker and A. Leonessa. 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 behavior of nature's predators is considered for designing a high speed tracking controller for nonholonomic vehicles, whose dynamics are represented using a unicycle model. To ensure that the vehicle behaves intuitively and mimics the biologically inspired predator-prey interaction, saturation constraints based on Ackermann steering kinematics are added. A new strategy for mapping commands back into a viable envelope is introduced, and the restrictions are accounted for using Lyapunov stability criteria. Following verification of the saturation constraints, the proposed algorithm was implemented on a testing platform. Stable trajectories of up to 9 m/s were achieved. The results presented show that the algorithm demonstrates significant promise in high speed trajectory tracking with obstacle avoidance.

1. Introduction

With the development of driverless technologies, there is an increasing demand for high speed systems capable of dealing with unstructured environments. Many commonly used obstacle avoidance methods focus on generating a path and forcing the vehicle directly to the desired trajectory [1–7]. Unfortunately, in many cases, such as potential field methods, the desired route is rarely smooth and does not sufficiently account for the desired velocity or vehicle capability [3, 8–13].

To find a more intuitive behavior, we look to biology for inspiration, specifically a generic predator-prey interaction. A cheetah chasing its prey, for example, does not directly mimic the path of its target. Instead, it creates a relatively smooth path, although its target may be moving somewhat chaotically [14]. The smooth path, in turn, allows the cheetah to maintain much of its high speed and still meet the desired goal [14]. As will be shown, the proposed algorithm mimics this behavior by introducing a reference system for the vehicle to follow, analogous to how a predator chases its prey.

Due to the high speed nature of our goal, this study focuses on Ackermann steering based platforms. Because of its simplicity and ability to capture nonholonomic constraints, we begin by considering a unicycle model, which

will be shown to fit within the confines of an Ackermann platform by adding saturation constraints. While there exists extensive research in the control of unicycle type robotic systems [15–21], the study begins with the framework laid out in [21, 22] as it introduces a relationship similar to the predator-prey interaction. In particular, when considering the path of the cheetah, its trajectory can be thought of as a filter to the prey's path; this is possible due to the cheetah maintaining a certain following distance from its prey. The larger the distance is, the more filtered the path becomes. In [22], the authors provide a control algorithm that mimics this behavior. By controlling velocity and angular rate, an algorithm is designed in [22] which theoretically tracks a virtual target at a time-varying distance. However, while capturing certain constraints, the unicycle model, and by consequence the work of [22], are not comprehensive. Much as the cheetah is unable to immediately turn independently of speed, neither are many ground vehicles. This paper focuses on extending preceding work to better satisfy the desired bioinspired model.

This work is organized as follows. In Sections 2 and 3 we introduce the algorithm laid out in [22] along with additional restrictions placed on the system structure and following distance parameters, which are aimed at bolstering controller

reliability. Moreover, we evaluate several limitations of the base algorithm in the context of the biologically inspired predator-prey relationship and high speed ground vehicles. In Section 4, we introduce constraints on angular rate and velocity commands, which are based on Ackermann steering platform kinematics. In this section, we develop a new strategy for mapping commands into a viable control space that reflects these kinematic restraints. We additionally provide constraints on the following distance parameters to ensure stability. The algorithm is shown to satisfy the Lyapunov stability criteria, while still meeting all of these requirements.

The proposed algorithm was ultimately implemented on a testing platform, the results of which are presented in Section 5. To satisfy high-level commands, low-level PI controllers were used to control commanded velocity and angular rate. The reference system was manipulated using a basic potential field methodology of strictly attractors and repulsors similar to that presented in [23–25]. Using an Extended Kalman Filter to provide reliable state information, the full algorithm was used to achieve stable trajectories up to 9 m/s, the top speed of the platform. Further work will focus on accounting for actuator dynamics and extending the method to multiple platforms.

2. System Definition

We begin with a unicycle model as it adequately captures the nonholonomic constraints of our Ackermann steering platform. Moreover, the simplicity of the model directly lends itself to controller design. While typically used for differential drive vehicles, we note that this relationship is not exclusive, and by developing saturation constraints, we can force the unicycle model to operate in the confines of an Ackermann steering platform. The unicycle model used is as follows,

$$\dot{p}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} v(t) \cos(\theta(t)) \\ v(t) \sin(\theta(t)) \end{bmatrix}, \quad (1)$$

$$\dot{\theta}(t) = \omega(t), \quad (2)$$

where $p(t) \triangleq [x(t) \ y(t)]^T \in \mathbb{R}^2$ is the vehicle's position, $v(t) \in \mathbb{R}$ is the longitudinal velocity, $\theta(t) \in \mathbb{R}$ is the heading, and $\omega(t) \in \mathbb{R}$ is the vehicle's angular velocity. A reference system is then introduced to represent a virtual target,

$$p_r(t) \triangleq \begin{bmatrix} x_r(t) \\ y_r(t) \end{bmatrix}, \quad (3)$$

where $p_r(t) \in \mathbb{R}^2$ is the reference system's position. In the interest of continuous controller commands, we must guarantee that the reference system is composed of class C^1 functions in time. For convenience, the linear velocity of the reference system is defined as

$$v_r(t) \triangleq \sqrt{\dot{x}_r(t)^2 + \dot{y}_r(t)^2}, \quad (4)$$

where $v_r(t) \in \mathbb{R}$.

In the following section, we will see that the controller design will guarantee the system described by (1) and (2)

will converge to a sufficiently small neighborhood around the reference system, described by (3), such that $\|p(t) - p_r(t)\| \rightarrow d^*(t)$, as $t \rightarrow \infty$, where $d^*(t) > 0$ is a user defined nominal following distance. By guaranteeing that the vehicle follows the reference system in this manner, we enforce our bio-inspired predator-prey interaction mentioned previously.

3. Control Design

The overall controller is presented in two distinct steps. The first step involves the basic design, while the second step includes saturation algorithms based on Ackermann steering kinematics. In this first step, we assume that the reference system can be controlled to follow a desired trajectory. By ensuring that the actual system converges to a small neighborhood of the reference system, we ensure that the system follows the desired trajectory as well. At the same time, a sufficiently large separation between reference system and vehicle allows a buffer for the vehicle to track the reference, without violating its nonholonomic constraints.

In order to properly design the following distance, it is necessary to define the error between the vehicle and reference system. We start by assuming that the origins of the body fixed coordinate frame, \mathcal{B} , and of the global inertial coordinate frame, \mathcal{U} , coincide with the center of mass of the vehicle in the horizontal plane. An orthonormal transformation from \mathcal{B} to \mathcal{U} is then defined,

$$R(\theta(t)) \triangleq \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) \\ \sin(\theta(t)) & \cos(\theta(t)) \end{bmatrix}. \quad (5)$$

Using this transformation along with (1) and (3), the position error can be expressed in the body coordinate frame as

$$e(t) \triangleq R^T(\theta(t))(p_r(t) - p(t)), \quad (6)$$

where $e(t) \in \mathbb{R}^2$. This position error can be thought of as a vector expression of longitudinal and lateral error in the body frame. Next we introduce the commanded distance vector,

$$\delta(t) \triangleq \begin{bmatrix} d(t) \\ 0 \end{bmatrix}, \quad (7)$$

where $d(t) > 0$ is the commanded following distance. By guaranteeing that $e(t) \rightarrow \delta$ as $t \rightarrow \infty$, the longitudinal error will converge to $d(t)$ while the lateral error goes to zero.

To provide this behavior, we introduce the following control law ([22]),

$$\begin{aligned} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} &= \Delta^{-1}(t) \\ &\cdot (K \tanh(e(t) - \delta(t)) + R^T(\theta(t)) \dot{p}_r(t) - \dot{\delta}(t)), \end{aligned} \quad (8)$$

where

$$\Delta(t) \triangleq \begin{bmatrix} 1 & 0 \\ 0 & d(t) \end{bmatrix}, \quad K \triangleq \begin{bmatrix} k_v & 0 \\ 0 & k_\omega \end{bmatrix}, \quad (9)$$

and $k_v > 0$ and $k_\omega > 0$ are scalar tuning constants used to provide bounded signals to the commanded longitudinal and angular velocities, respectively. This control law, which is more thoroughly derived in [22], is based on Lyapunov stability theory. In [22], the authors show it to be the natural result that guarantees a negative definite Lyapunov time derivative, given the Lyapunov function (12) introduced in the following theorem.

In order to ensure that $\Delta^{-1}(t)$ is defined for all $t \geq 0$, we must guarantee that $d(t) \neq 0$ for all $t \geq 0$. In practice, if $d(t) < 0$, the vehicle will follow the reference system but in reverse, such that $v(t) < 0$. As such, we restrict the commanded following distance to $d(t) > 0$. For design purposes, a more rigorous constraint of $d(t) > (\beta - \varepsilon) > 0$, for all $t \geq 0$, is enforced, where $\beta > \varepsilon > 0$ are tuning parameters used to place a minimum bound on $d(t)$. The commanded following distance, $d(t)$, is designed such that $d(t) \rightarrow d^*(t)$ as $t \rightarrow \infty$, provided that $d^*(t) \geq \beta$ as well.

At this point, we note the intuitive form of (8). For instance, from the $R^T(\theta(t))\dot{p}_r(t)$ term, we see that the reference velocity, projected along the vehicle's longitudinal axis, positively contributes to the commanded velocity, $v(t)$. Likewise the reference velocity, projected along the lateral axis, contributes to the commanded angular rate, $\omega(t)$. In examining the $\delta\dot{d}(t)$ term, we see that an increasing distance negatively impacts the commanded velocity, as would be expected. Lastly, with regard to the $K \tanh(e(t) - \delta(t))$ term, we see that the difference between longitudinal error and the commanded distance, $d(t)$, contributes positively to the commanded velocity. Along the same lines, lateral error contributes positively to the angular rate command.

With regard to the $\tanh(\cdot)$ term of (8), which is a componentwise operation, we note that while it is unnecessary for theoretical stability, it adds a layer of tuning to the controller. This layer proves particularly useful in experimentation. The function is chosen for its natural saturation capability. Based on the constants k_v and k_ω , the function directly contributes bounded signals to commanded longitudinal and angular velocities, which are based on the difference between the tracking error $e(t)$ and distance vector $\delta(t)$.

Theorem 1. Consider the system described by (1) and (2), the reference system described by (3), and the feedback controller described by (8). If the nominal distance is restricted such that $d^*(t) \geq \beta$, for all $t \geq 0$, and the distance $d(t)$ is updated according to

$$\dot{d}(t) = \begin{cases} \Gamma(t), & d(t) \geq \beta \\ \Gamma(t) + \frac{\beta - d(t)}{d(t) - (\beta - \varepsilon)}, & d(t) < \beta, \end{cases} \quad (10)$$

$$d(0) \geq \beta,$$

where

$$\Gamma(t) \triangleq \dot{d}^*(t) - \lambda(d(t) - d^*(t)), \quad (11)$$

with tuning constant $\lambda > 0$, then the distance $d(t)$, between the vehicle and reference system, converges to the desired distance,

$d^*(t)$, while guaranteeing that $d(t) > (\beta - \varepsilon) > 0$. Meanwhile, the tracking error $e(t)$ converges to the distance vector $\delta(t)$.

Proof. Consider the Lyapunov function candidate

$$V(t) = \frac{1}{2} e_1^T(t) e_1(t) + \frac{1}{2} (d(t) - d^*(t))^2, \quad (12)$$

where

$$e_1(t) \triangleq e(t) - \delta(t). \quad (13)$$

In order to examine the derivative of the Lyapunov function along the system trajectories, the time derivative of the tracking error, $e(t)$, given by (6), is computed as follows,

$$\begin{aligned} \dot{e}(t) &= \dot{R}^T(\theta(t), \omega(t)) (p_r(t) - p(t)) \\ &\quad + R^T(\theta(t)) (\dot{p}_r(t) - \dot{p}(t)). \end{aligned} \quad (14)$$

The time derivative of the orthonormal transformation, $R(\theta(t))$, is given by

$$\begin{aligned} \dot{R}(\theta(t), \omega(t)) &= \begin{bmatrix} -\omega(t) \sin(\theta(t)) & -\omega(t) \cos(\theta(t)) \\ \omega(t) \cos(\theta(t)) & -\omega(t) \sin(\theta(t)) \end{bmatrix} \\ &= R(\theta(t)) S(\omega(t)), \end{aligned} \quad (15)$$

where

$$S(\omega(t)) \triangleq \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix}. \quad (16)$$

By substituting (1) and (15) into (14), the error dynamics can be simplified to

$$\begin{aligned} \dot{e}(t) &= S^T(\omega(t)) R^T(\theta(t)) (p_r(t) - p(t)) \\ &\quad + R^T(\theta(t)) \dot{p}_r(t) \\ &\quad - R^T(\theta(t)) \begin{bmatrix} v(t) \cos(\theta(t)) \\ v(t) \sin(\theta(t)) \end{bmatrix} \\ &= -S(\omega(t)) e(t) + R^T(\theta(t)) \dot{p}_r(t) - \begin{bmatrix} v(t) \\ 0 \end{bmatrix} \\ &= -S(\omega(t)) e(t) + R^T(\theta(t)) \dot{p}_r(t) \\ &\quad - \begin{bmatrix} v(t) \\ d(t) \omega(t) \end{bmatrix} + \begin{bmatrix} 0 \\ d(t) \omega(t) \end{bmatrix} \\ &= -S(\omega(t)) e_1(t) - \Delta(t) \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \\ &\quad + R^T(\theta(t)) \dot{p}_r(t). \end{aligned} \quad (17)$$

By substituting the control law given in (8), the error dynamics are then further simplified to

$$\dot{e}(t) = -S(\omega(t)) e_1(t) - K \tanh(e_1(t)) + \dot{\delta}(t). \quad (18)$$

The error dynamics given by (18) are then substituted into the time derivative of the Lyapunov function,

$$\dot{V}(t) = e_1^T(t) \dot{e}_1(t) + (d(t) - d^*(t)) (\dot{d}(t) - \dot{d}^*(t)). \quad (19)$$

If we consider the case in which $d(t) \geq \beta$, we then substitute (10), (11), and (18) into (19) to obtain

$$\begin{aligned} \dot{V}(t) &= e_1^T(t) (-S(\omega(t)) e_1(t) - K \tanh(e_1(t))) \\ &\quad - \lambda (d(t) - d^*(t))^2 \\ &= -e_1^T(t) K \tanh(e_1(t)) - \lambda (d(t) - d^*(t))^2. \end{aligned} \quad (20)$$

From (20), it is trivial to show that again $\dot{V}(t) \leq 0$ for all $t \geq 0$, $d(t) \geq \beta$.

The Lyapunov time derivative is again examined when $d(t) < \beta$. Again, we substitute (10), (11), and (18) into (19) to obtain

$$\begin{aligned} \dot{V}(t) &= -e_1^T(t) K \tanh(e_1(t)) + (d(t) - d^*(t)) \\ &\quad \cdot \left(-\lambda (d(t) - d^*(t)) + \frac{\beta - d(t)}{d(t) - (\beta - \varepsilon)} \right) \\ &= -e_1^T(t) K \tanh(e_1(t)) - \lambda (d(t) - d^*(t))^2 \\ &\quad + (d(t) - d^*(t)) \left(\frac{\beta - d(t)}{d(t) - (\beta - \varepsilon)} \right). \end{aligned} \quad (21)$$

The commanded following distance, $d(t)$, is restricted such that $d(t) > (\beta - \varepsilon)$ by design. This behavior is the result of $(\beta - d(t))/(d(t) - (\beta - \varepsilon)) \rightarrow \infty$ as $d(t) \rightarrow (\beta - \varepsilon)$. As $d(t)$ decreases below β but still remains larger than $(\beta - \varepsilon)$, the additional corrective term pushes $d(t)$ back toward β , eventually guaranteeing $\dot{d}(t) > 0$ as $d(t) \rightarrow (\beta - \varepsilon)$. As such, the corrective term $(\beta - d(t))/(d(t) - (\beta - \varepsilon)) > 0$ for $d(t) < \beta$. Additionally, since $d^*(t) \geq \beta > d(t)$, we find that $(d(t) - d^*(t))((\beta - d(t))/(d(t) - (\beta - \varepsilon))) < 0$. Since the remainder of the Lyapunov time derivative terms are the same as given in (20), we can conclude that $\dot{V} \leq 0$ for all $t \geq 0$. \square

3.1. Simulation Results. The behavior of the control algorithm is examined prior to the introduction of saturation constraints on commanded longitudinal and angular velocities. Before simulation, it is necessary to choose a nominal following distance. By choosing

$$d^*(t) = \alpha v_r(t) + \beta, \quad (22)$$

for $\alpha > 0$ and $\beta > 0$, we enforce a behavior that tends toward the bioinspired predator-prey model discussed earlier. For example, with higher reference velocities, we intuitively allow a greater distance to respond to instantaneous changes in reference direction. Likewise, (22) draws the vehicle closer to tightly follow slow trajectories. Analogously, large separation allows for a cheetah to easily maneuver while maintaining high speed, whereas a small separation lends itself to tighter tracking.

TABLE 1: Parameters for simulation without saturation.

Desired Reference Trajectory	Tuning Parameters	Initial Conditions
$r_x(t) = 0.5t$	$k_v = 1$	$x_r(0) = 0$
$r_y(t) = 10 \sin(0.5t)$	$k_\omega = 1$	$y_r(0) = 0$
	$\lambda = 1$	$x(0) = -0.1$
	$\alpha = 0.5$	$y(0) = 0$
	$\beta = 0.1$	$\theta(0) = 0$
		$d(0) = 0.1$

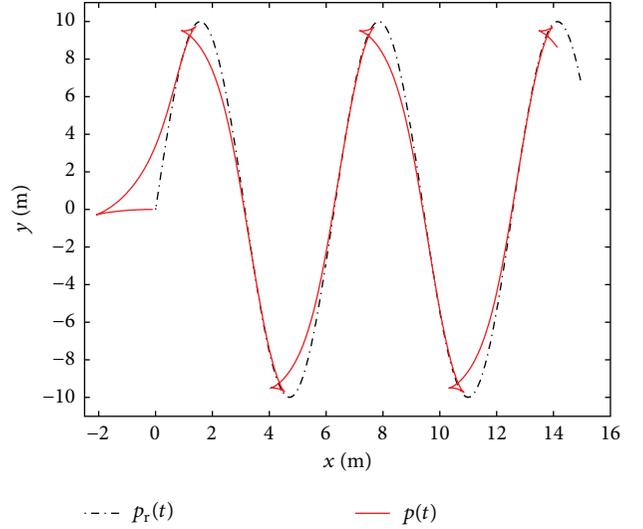


FIGURE 1: Sinusoidal trajectory without saturation.

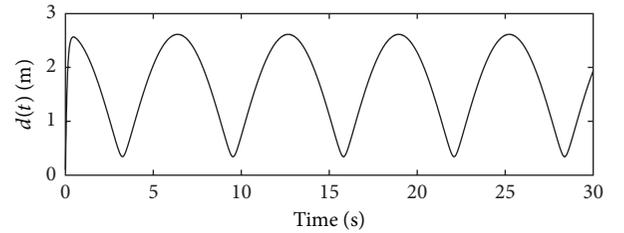


FIGURE 2: Following distance without saturation.

To evaluate the performance of the controller, we consider the situation in which the reference system is defined as two decoupled first-order systems,

$$\dot{p}_r(t) = \begin{bmatrix} -10x_r(t) + 10r_x(t) \\ -10y_r(t) + 10r_y(t) \end{bmatrix}, \quad (23)$$

where $r_x(t)$ and $r_y(t)$ are the desired reference trajectories in x and y , respectively. Figures 1–3 show the results of simulation using the parameters given in Table 1.

Figure 1 shows the theoretical performance of the system. As intended, the vehicle converges tightly to the reference system's path, while maintaining a varying following distance, shown in Figure 2. Further inspection shows that the minimal distance coincides with the sinusoidal peaks in the trajectory.

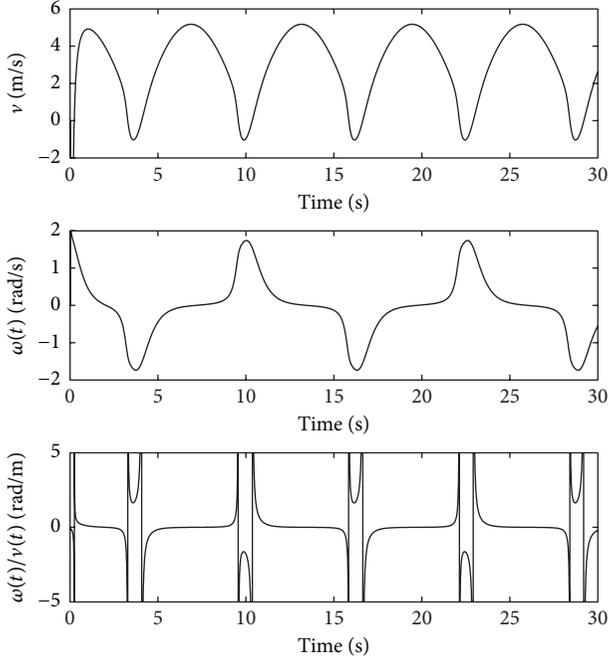


FIGURE 3: Controller commands for sinusoidal trajectory without saturation.

This relationship is a result of lower reference velocity in the y direction, which leads to close tracking at low speeds around the peaks of the sine wave.

With regard to the controller commands, seen in Figure 3, the trends are as expected. Since the controller is based on a unicycle model, there is no obvious correlation between velocity and angular rate commands. In fact, many of the largest angular rate commands, with regard to magnitude, correspond to the lowest velocity commands. This relationship is exemplified by the third graph which examines the ratio of angular rate to longitudinal velocity. At multiple instances throughout the simulation, this ratio experiences singularities.

4. Saturation Constraints

Since the control algorithm is geared toward high speed control and obstacle avoidance, we consider an Ackermann steering platform for testing and implementation. However, due to the behavior discussed in the previous section, where the ratio of angular to longitudinal velocity becomes singular, the control algorithm is not suitable for an Ackermann steering platform without modification.

Additionally, we observe in Figure 3 that the controller naturally commands negative velocities. Specifically, at the start of simulation, the vehicle immediately backs up to satisfy the error and distance requirements, which for this case are largely dictated by $\hat{\delta}(t)$. Further instances of negative velocity occur throughout the trajectory while undergoing tight cornering. In practice, however, we would like the behavior to better follow that of a predator chasing its prey. In the case of the cheetah, it is unlikely that the creature will stop,

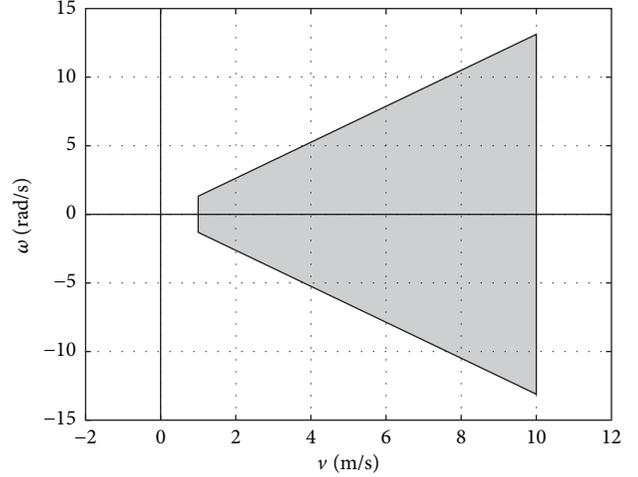


FIGURE 4: Graphical representation of viable control envelope (shaded).

back up while turning, and then continue the chase. Instead, it is more intuitive to continue along its current path, preserving speed while correcting heading to the best of its ability.

In order to elicit the same behavior, the following saturation constraints are introduced based on an Ackermann steering platform,

$$\left| \frac{\omega(t)}{v(t)} \right| \leq \frac{\tan(\phi_{\max})}{L}, \quad (24)$$

$$v_{\min} \leq v(t) \leq v_{\max}, \quad (25)$$

where $L > 0$ is the wheelbase length from rear to front axle, $\phi_{\max} > 0$ is the maximum steering angle deflection referenced from the longitudinal body axis, $v_{\min} > 0$ is a lower bound on the longitudinal velocity, and $v_{\max} > 0$ is an upper bound on longitudinal velocity. In this case it is assumed that the maximum steering angle is the same in both directions, allowing ϕ_{\max} to be used for both minimum and maximum saturation of the angular rate.

The envelope is graphically demonstrated via the shaded region seen in Figure 4. In Figure 4, we note the minimum bound on velocity, which excludes the origin. As a result, the algorithm is unable to achieve a stationary command. This choice is a product of our bioinspired cheetah model, which we want to continue on a forward trajectory at a minimum velocity bound, rather than stopping or reversing direction. Additionally, if commands were mapped to the origin during saturation, the vehicle's position would remain unchanged, potentially giving it no way to exit the saturation condition. To prevent this stagnation, we again enforce a minimum velocity.

4.1. First Stage Saturation. During operation, the control law given in (8) can viably produce a command anywhere in the v - ω plane; however, it is desired to map the commands to somewhere within the given envelope while still preserving directional intent. Moreover, any mapping of commands

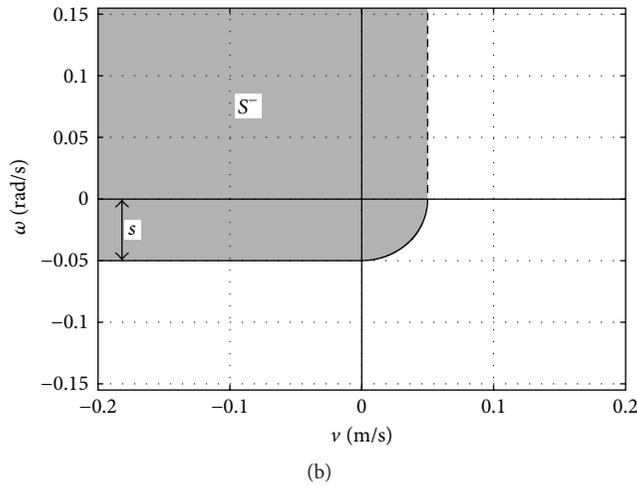
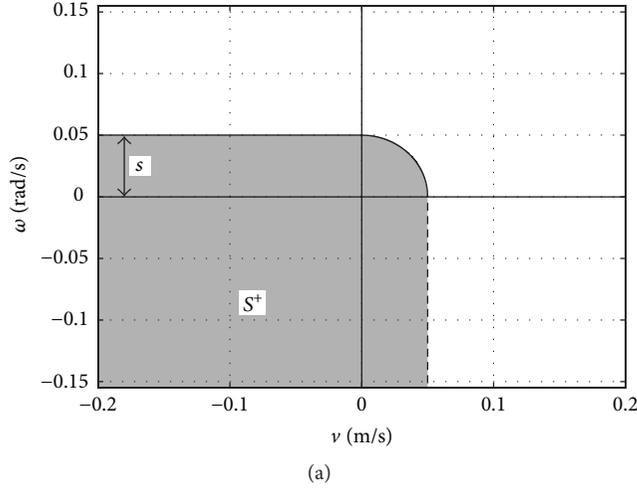


FIGURE 5: Graphical representation of S^+ (a) and S^- (b).

should be continuous, not allowing for instantaneous transitions from one region of the envelope to another. In order to address these constraints, we present a dual stage saturation algorithm.

Based on our choice of regions in the second stage of saturation, which will be discussed in the next section, it is possible for commands to instantaneously change locations on the envelope. Specifically, if the controller commands cross the negative v axis or pass through the origin, an instantaneous change will occur in the second stage mapping, which is undesirable. In order to prevent these potential problems and preserve directional intent, the first stage is used to prevent commands from entering into a small boundary around the negative v axis and the origin. It should be noted that this saturation stage does not result in commands mapped back to the envelope, which is instead achieved by the second stage.

The first stage ensures that controller commands cannot traverse the region shown in Figures 5 and 6, which is defined as

$$S \triangleq S^+ \cap S^-, \quad (26)$$

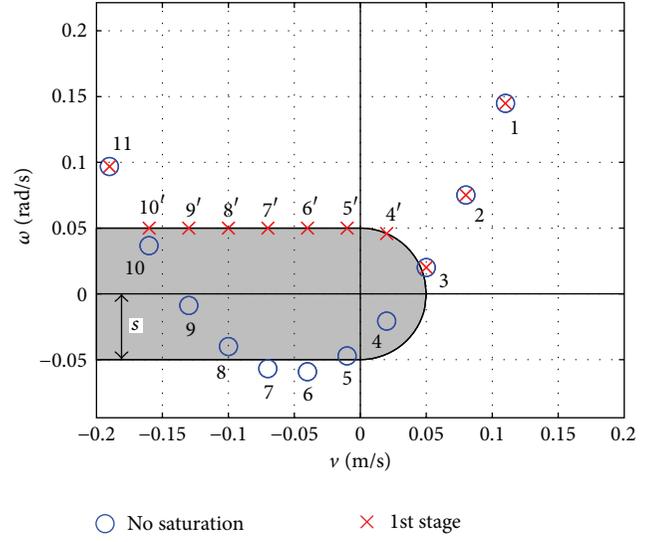


FIGURE 6: First stage representation of S with saturation example.

where

$$S^+ \triangleq \left\{ (v, \omega) \in \mathbb{R}^2 : 0 \leq \omega \leq s, v \leq \sqrt{s^2 - \omega^2} \right\} \cup \left\{ (v, \omega) \in \mathbb{R}^2 : v < s, \omega < 0 \right\}, \quad (27)$$

$$S^- \triangleq \left\{ (v, \omega) \in \mathbb{R}^2 : -s \leq \omega \leq 0, v \leq \sqrt{s^2 - \omega^2} \right\} \cup \left\{ (v, \omega) \in \mathbb{R}^2 : v < s, \omega > 0 \right\}, \quad (28)$$

where $s \in (0, v_{\min})$ defines the size of this small region around the negative v axis. In practice, s can be arbitrarily small as its only purpose is to provide a boundary around the negative v axis. In general, small values, such that $s \ll v_{\min}$, are preferred to provide the least interference with controller intent. Figure 6 gives an example of the saturation implementation, which is detailed in the following discussion.

During implementation, if at time $t_e > 0$ the controller issues a command such that $(v(t_e), \omega(t_e)) \in \partial S$, then $\omega_e \triangleq \omega(t_e)$ is recorded. If $\omega_e \geq 0$, then for all subsequent controller commands such that $(v(t), \omega(t)) \in S^+$ the algorithm maps the commands to $(v(t), \omega_s(t)) \in \partial S^+ \cap S^+$. Likewise in the case of $\omega_e < 0$, for all subsequent commands such that $(v(t), \omega(t)) \in S^-$ the algorithm maps the command to $(v(t), \omega_s(t)) \in \partial S^- \cap S^-$. We note that $\omega_s(t)$, the resultant angular rate command, is defined by the point on the boundary that corresponds to the original velocity command.

Figure 6 demonstrates a parabolic command evolution before and after the first saturation stage. In this case, we consider the commands evolving from right to left in the order they are numbered. The circles represent the original commands from the controller, and the crosses demonstrate the resulting command after the first saturation stage. Examining the figure, we see that commands 1–3 are left unaltered, as they are outside of the region S . At some point between 3 and 4, the command crosses the boundary of S , at which point ω_e is recorded to be of some positive value. Since $\omega_e \geq 0$, all subsequent commands such that $(v(t), \omega(t)) \in S^+$ are mapped

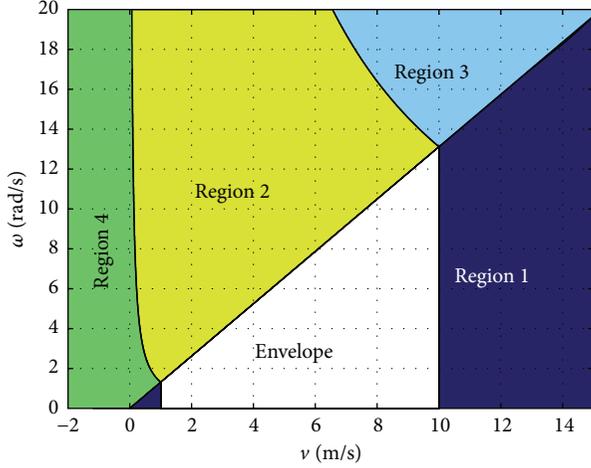


FIGURE 7: Region classification for second stage saturation.

to $\partial S^+ \cap S^+$, while preserving the velocity command, $v(t)$. This trend is the case for commands 4–10 which are mapped to 4'–10', respectively. Of particular interest are commands 6 and 7. Although these commands exit the region S , they are still contained in S^+ . As such, the saturation algorithm continues to map these commands to ∂S^+ using the same methodology. Lastly, we look at command 11, which occurs at some point after the commands have exited S^+ . In this case, since $(v(t), \omega(t)) \notin S^+$ the saturation algorithm is no longer active.

4.2. Second Stage Saturation. Once the commands are passed through the first stage of saturation, they are categorized to a specific region based on their location in the v - ω plane. Figure 7 shows four separate regions which use various methodologies to map the commands into the viable envelope. In Region 1, we focus on preserving the radius of curvature of the original commands as this directly relates to the intended path of the controller. Due to the limitations of the Ackermann platform, however, not all curvatures are achievable. As such, in Region 2, the second stage saturation algorithm maps commands to the nearest achievable radius of curvature while preserving lateral acceleration. We choose to preserve lateral acceleration so that the acceleration perceived by the vehicle and its sensors will remain unchanged. In the case of Regions 3 and 4, commands originate outside of any achievable radius of curvature or lateral acceleration. As a result, these commands are mapped to the closest achievable points that preserve intent. As expected, all commands originating within the envelope are left unaltered. For the remainder of this discussion, we only consider cases of $\omega(t) \geq 0$, as the other case is simply a reflection about the v axis.

4.2.1. Region 1 Methodology. Ideally, during saturation it is desirable to reproduce the same trajectories that are commanded by the controller. In order to elicit this behavior, we examine the radius of curvature given by

$$r(t) = \frac{v(t)}{\omega(t)}. \quad (29)$$

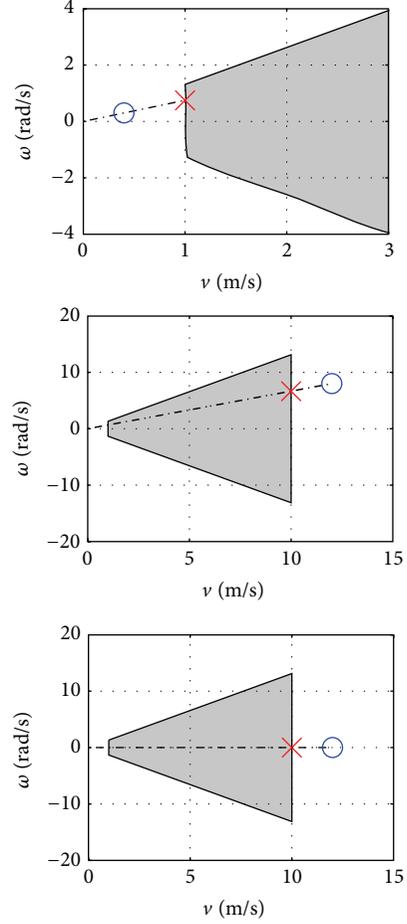


FIGURE 8: Region 1 examples for second stage saturation.

When the requested command lies within Region 1, which is given by the dark blue regions on the left and right sides of the envelope in Figure 7, the second stage saturation algorithm maps the new command to the nearest point on the envelope that provides the same radius of curvature. By maintaining the same radius of curvature, we ensure that the intended path is followed to the best of the vehicles capability. We note that the origin is included as part of Region 1, however, this presents a potential for discontinuous mapping depending on command evolution. For this reason, the first stage saturation method is used to prevent transitions through the origin.

Figure 8 provides several examples of Region 1 saturation. In Figure 8, the circles represent the command received from the first stage of saturation, and the crosses represent the command following the second saturation stage. In the first example, the command is mapped to the minimum velocity while still preserving the $\omega(t)/v(t)$ ratio. The second and third examples exhibit the same behavior, mapping to the maximum velocity instead. Of particular interest is the third example, which maps along the v axis. In this case, the radius of curvature is infinite, but by using the inverse of curvature, the methodology is preserved. For this reason, the reciprocal of the radius is typically used to prevent singularities in computation.

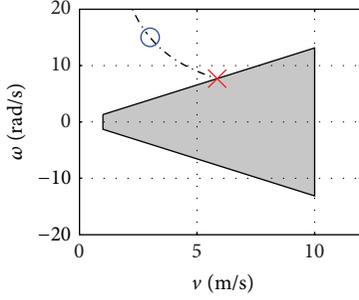


FIGURE 9: Region 2 example for second stage saturation.

4.2.2. Region 2 Methodology. Due to the constraints of an Ackermann platform, there are a variety of conditions in which the desired radius of curvature is not achievable, as is the case of Region 2. To find a reasonable alternative, we start by considering the lateral acceleration of the vehicle given by

$$a_L(t) = v(t) \omega(t). \quad (30)$$

Commands originating in Region 2 are mapped to the point on the envelope that is characterized by the same lateral acceleration. This point also corresponds to the nearest achievable radius of curvature. Figure 9 demonstrates saturation behavior for Region 2. The circle represents the command received from the first stage of saturation, and the cross represents the command following the second saturation stage. In this figure, the black dashed line represents the laws of constant lateral acceleration. As explained, both commands have the same lateral acceleration, with one being in the achievable command space.

4.2.3. Region 3 Methodology. While lateral acceleration offers an intuitive choice for mapping commands to an obtainable radius of curvature, Ackermann platforms inherently have a maximum lateral acceleration, given by

$$a_{L_{\max}} \triangleq v_{\max}^2 \frac{\tan(\phi_{\max})}{L}. \quad (31)$$

When the requested command exceeds the minimum radius of curvature and the maximum lateral acceleration, the command is mapped to the maximum velocity and maximum turning angle, represented by the corner of the envelope. Region 3 of Figure 7 depicts the area in question. By holding the command at the corner of the envelope in Region 3, we ensure a continuous mapping from Region 2 to Region 3 and from Region 3 to Region 1.

4.2.4. Region 4 Methodology. Although lateral acceleration offers a good methodology for mapping commands, it does not always elicit optimal behavior. Recalling from Section 3.1, the controller can command high angular rates while also commanding little to no longitudinal velocity. In these cases, the lateral acceleration is small. If we were to map to the envelope maintaining this lateral acceleration, the result would yield a forward velocity, with a nearly zero angular rate, which

would be opposite to desired. Instead, the vehicle should ideally turn as sharp as possible.

In Region 4, we examine the situation in which the radius of curvature is not obtainable and the command lies below the minimum lateral acceleration threshold given by

$$a_{L_{\min}} \triangleq v_{\min}^2 \frac{\tan(\phi_{\max})}{L}. \quad (32)$$

In this case, the command is mapped to the minimal velocity and maximum angular rate, characterized by the left corner of the envelope given in Figure 7. For the situations in which the vehicle has a large requested angular rate with small velocity, the algorithm ensures that the vehicle turns as sharp as possible at a minimal velocity until the controller requests more achievable behavior.

We extend Region 4 to encompass all negative velocity commands as well since the goal is to turn as much as possible until the controller requests positive velocities. We note that while Region 4 includes portions of the first stage of saturation, these portions will not be reached due to the effects of the first stage of saturation. If the controller commands were to cross the v axis while the requested velocity is negative, indicated by Figures 5 and 6, the second stage saturation algorithm would fully reverse the steering command. However, the first level of saturation restricts this potential chattering behavior by keeping the command in the same quadrant until the controller requests positive velocities.

4.3. Maintaining Lyapunov Criteria. For the above cases, the controller commands are altered without regard to how it affects the tracking performance of the overall system. In order to ensure that the asymptotic stability of the error dynamics is still achieved, the control law is backsolved from (8) as ([22])

$$\begin{aligned} \dot{p}_r^{\text{sat}}(t) = R(\theta(t)) \left(\Delta(t) \begin{bmatrix} v^{\text{sat}}(t) \\ \omega^{\text{sat}}(t) \end{bmatrix} \right. \\ \left. - K \tanh(e(t) - \delta(t)) + \dot{\delta}(t) \right), \end{aligned} \quad (33)$$

where $\dot{p}_r^{\text{sat}}(t) \in \mathbb{R}^2$ is the resulting reference time derivative, and, in this case, $v^{\text{sat}}(t)$ and $\omega^{\text{sat}}(t)$ are determined from the saturation algorithm. It is important to realize that $d^*(t)$ must be at least a class C^1 function in time to guarantee both that $\dot{d}(t)$ exists and also that the control law is continuous. To elicit this behavior, we modify the nominal following distance to be

$$\ddot{d}^*(t) + 2\zeta_d \omega_d \dot{d}^*(t) + \omega_d^2 d^*(t) = \omega_d^2 d_{\text{ref}}(t), \quad (34)$$

where $\zeta_d > 0$, $\omega_d > 0$ are tuning constants and $d_{\text{ref}}(t)$ is the user desired reference distance.

4.4. Simulation Results with Saturation. Prior to saturation, we chose $d^*(t)$ to follow the relationship given by (22). In order to obtain similar behavior during saturation and adhere

TABLE 2: Parameters for simulation with saturation.

Desired Reference Trajectory	Tuning Parameters	Initial Conditions
$r_x(t) = 0.5t$	$k_v = 1$	$x_r(0) = 0$
$r_y(t) = 10 \sin(0.5t)$	$k_\omega = 1$	$y_r(0) = 0$
	$\lambda = 1$	$x(0) = -0.1$
	$\alpha = 0.5$	$y(0) = 0$
	$\beta = 0.1$	$\theta(0) = 0$
	$L = 0.3556$	$d(0) = 0.1$
	$\gamma_{\max} = 25^\circ$	$d^*(0) = 0.1$
	$v_{\min} = 1$	$\dot{d}^*(0) = 0$
	$v_{\max} = 10$	
	$s = 0.01$	
	$\omega_d = 2.5$	
	$\zeta_d = 0.85$	

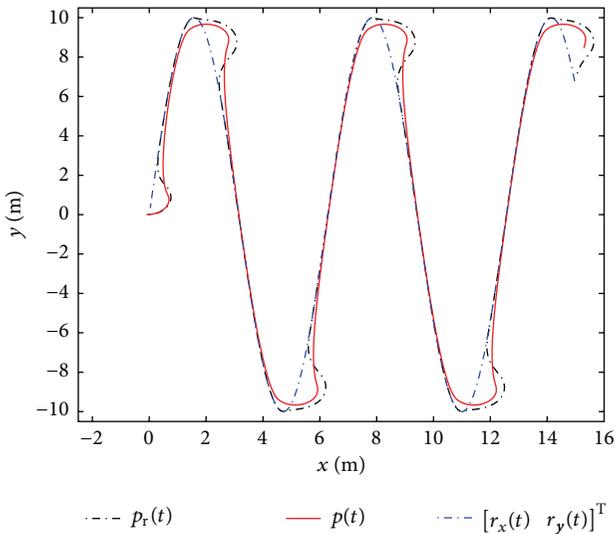


FIGURE 10: Sinusoidal trajectory with saturation algorithm.

to the criteria of the previous section, we choose a desired reference distance of

$$d_{\text{ref}}(t) = \alpha v_r(t) + \beta. \quad (35)$$

For comparison purposes, we introduce the same reference trajectory of (23). All simulation parameters are listed in Table 2.

Figure 10 demonstrates the resulting trajectory information with the addition of saturation constraints. Overall, the vehicle still follows the trajectory given by $[r_x(t) \ r_y(t)]^T$ of Table 2, which is desired. However, in the case of sharp turns, the vehicle continues forward, locking itself into a circular trajectory until it is able to reobtain the desired path. To facilitate this behavior, the reference system is pushed away from the trajectory detailed by $[r_x(t) \ r_y(t)]^T$, overriding the dynamics described by (23). In addition to following the path, the saturation algorithm also eliminated the prior initial behavior, characterized by an immediate reversal of trajectory

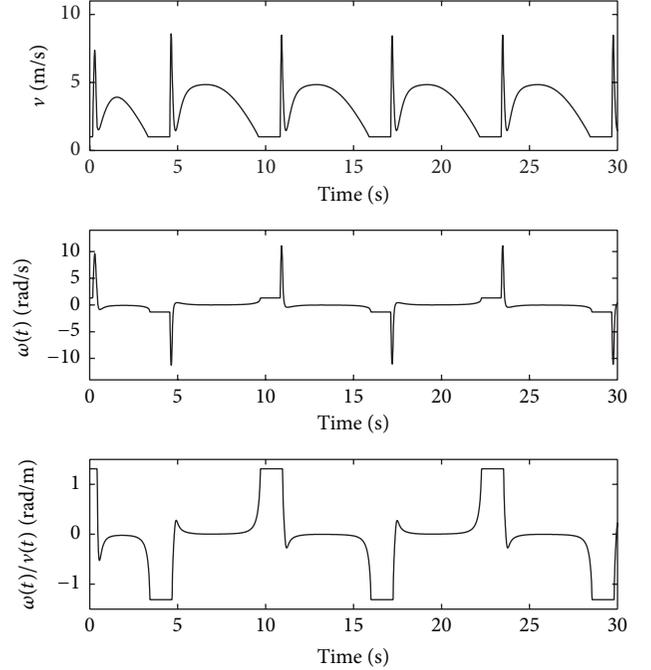


FIGURE 11: Controller commands for sinusoidal trajectory with saturation.

to satisfy distance requirements. Instead, we observe a situation in which the vehicle continues smoothly in a forward direction while steering back into the path.

From Figure 10 we can see the vehicle better adheres to our bioinspired model of the predator-prey interaction. Instead of reversing direction or undergoing unreasonably tight turns, the vehicle preserves speed and corrects heading to the best of its ability. In the same way, this behavior is largely what we would expect from a cheetah chasing its prey.

Further examination of the behavior is given in Figure 11. Clear limitations in the velocity are observed between 1 m/s and 10 m/s. Additionally, the implemented algorithm provides a smooth velocity command, as expected. This behavior is a result of a continuous mapping of saturation commands back to a region on or within the envelope.

Looking further at Figure 11, we see that the angular rate commands are significantly different from those obtained without saturation constraints. In general, the commanded angular rates achieve higher magnitudes than previously observed. From the third graph, however, it is obvious that there are clear bounds on the ratio of $\omega(t)/v(t)$. As such, the vehicle achieves higher angular rates, while still following the criteria of an Ackermann based platform.

While the saturation constraints perform as desired, it is important to ensure that the overall distance following behavior is not sacrificed as this is a significant trait of our bioinspired desired behavior. Figure 12 demonstrates the comparison of the new following distance to the original. As expected, the added filter (34) between $d^*(t)$ and $d_{\text{ref}}(t)$ introduces a phase shift and attenuation of $d(t)$ when saturation constraints are introduced. Of some interest is the fact that the distance appears more irregular, as opposed to

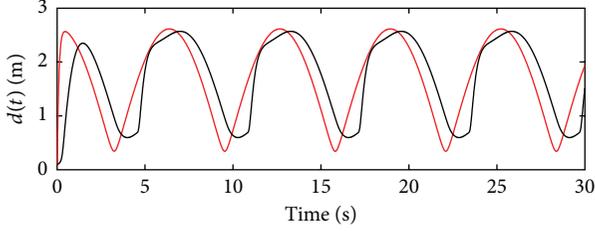


FIGURE 12: Comparison of following distance with (black) and without (red) saturation.



FIGURE 13: Experimental testing platform.

the smooth original distance. This trend is a result of the fluctuating reference velocity during periods of saturation. Despite these differences, the desired trend is still present in regard to reference system behavior. Consequently, the saturation constraints are held as a valid addition to the overall control algorithm.

5. Experimental Testing and Results

To investigate the practicality of the controller, the algorithm was implemented on an experimental platform. This section discusses the experimental setup, the reference system manipulation, and the results.

5.1. Experimental Setup. For the purpose of testing, the robotic platform is shown in Figure 13. This vehicle incorporates an Inertial Measurement Unit (IMU) that provides Euler angles at $\pm 2^\circ$, acceleration on 3 axes at ± 0.01 g, and angular rates at $\pm 0.2^\circ/\text{sec}$. Additionally, the vehicle hosts an onboard GPS which provides position feedback information within ± 1.5 m as well as velocity information accurate to within ± 0.5 m/s. The sensor information was incorporated into an Extended Kalman Filter, designed specifically for this project, which provided feedback information on position, heading, and velocity, as well as the biases for heading and acceleration measurements.

The onboard actuators consist of a brushless motor for speed and standard hobby servo for steering, both of which were controlled using a generic servo controller. In order to implement the controller in the form presented by this paper, low-level PI controllers were experimentally designed and implemented to directly regulate speed and angular velocity.

These controllers, the Kalman Filter, and the overall control algorithm were implemented through an onboard computer at an update rate of 40 Hz.

5.2. Potential Field Reference System Design. For the purpose of experimental testing, the reference system was manipulated according to a simple potential field algorithm similar to that presented in [23–25]. This method was chosen primarily because it offers a computationally simple manner to control the reference system. It also provides an intuitive velocity behavior, which is characterized by reducing velocity when obstacles are encountered.

To define the potential field algorithm, we begin by introducing distance between the reference system and the destination,

$$d_{\text{des}}(t) \triangleq \sqrt{(x_{\text{des},i} - x_r(t))^2 + (y_{\text{des},i} - y_r(t))^2}, \quad (36)$$

where $d_{\text{des}}(t) \in \mathbb{R}$, $x_{\text{des},i} \in \mathbb{R}$ is the x coordinate of the i th destination waypoint, and $y_{\text{des},i} \in \mathbb{R}$ is the y coordinate of the i th destination waypoint. Furthermore, the distance between the reference system and surrounding obstacles is defined as

$$d_{\text{obs},j}(t) \triangleq \sqrt{(x_{\text{obs},j} - x_r(t))^2 + (y_{\text{obs},j} - y_r(t))^2}, \quad (37)$$

where $d_{\text{obs},j}(t) \in \mathbb{R}$, $x_{\text{obs},j} \in \mathbb{R}$ is the x coordinate of the j th obstacle waypoint, and $y_{\text{obs},j} \in \mathbb{R}$ is the y coordinate of the j th obstacle waypoint. Based on these distance parameters, virtual force vectors are given such that

$$F_a(t) \triangleq \frac{F_{\text{ac}}}{d_{\text{des}}(t)} \begin{bmatrix} x_{\text{des},i} - x_r(t) \\ y_{\text{des},i} - y_r(t) \end{bmatrix}, \quad (38)$$

$$F_r(t) \triangleq \sum_{j=1}^{N_{\text{obs}}} -F_{\text{rc}} \left(\frac{W}{d_{\text{obs},j}(t)} \right)^n \begin{bmatrix} x_{\text{obs},j} - x_r(t) \\ y_{\text{obs},j} - y_r(t) \end{bmatrix}, \quad (39)$$

where $F_{\text{ac}} > 0$, $F_{\text{rc}} > 0$, $W > 0$, and $n > 1$ are tuning constants, $F_a(t) \in \mathbb{R}^2$ is the attractive force vector, $F_r(t) \in \mathbb{R}^2$ is the repulsive force vector, and N_{obs} is the number of obstacles within a given radius of the reference system.

From (38), we can see that the attractive force, $F_a(t)$, is essentially a unit vector multiplied by the tuning constant F_{ac} . This vector points from the reference system toward the current destination at all times. Because of this relationship, the reference system will be pulled toward the goal with some constant force dictated by F_{ac} .

Likewise, we note that (39) has a similar form to (38). This results in a force vector, $F_r(t)$, that points from a particular obstacle toward the reference system, repelling it. Because of the exponential relationship caused by n , the repulsive force increases as the reference system nears the obstacle. Consequently, the reference system is largely uninfluenced until it is within the vicinity of an obstacle, at which point it is pushed away. Ultimately, the combination of the attractive and repulsive forces, $F_a(t)$ and $F_r(t)$, respectively, causes the reference system to avoid any obstacles while moving toward the goal.

TABLE 3: Potential field tuning constants for experiments.

Potential Field Parameters
$F_{ac} = 10$
$F_{rc} = 8$
$W = 2$
$n = 1.5$
$E = 14$

In order to utilize the force vectors given in (38) and (39), the reference system is given the following dynamics,

$$m\ddot{p}_r(t) + c\dot{p}_r(t) = F_a(t) + F_r(t), \quad (40)$$

where $m > 0$ and $c > 0$ are tuning constants for inertia and damping of the virtual system, respectively. To define these constants, we start by considering the maximum desired velocity in the presence of only attractive forces. When maximum velocity of the reference system is achieved, the reference acceleration reduces to $\ddot{p}_r(t) = 0$, which yields

$$c\dot{p}_r(t) = F_a(t). \quad (41)$$

By taking the norm of both sides and considering the maximum reference velocity case, we can determine the damping as

$$c = \frac{F_{ac}}{v_{r,max}}. \quad (42)$$

In regard to determining the inertia, we assume that the reference system persists at maximum velocity throughout the trajectory. If the kinetic energy is maintained along the path, then the path maintains the same characteristic shape, regardless of velocity. As such, the inertial term is defined as

$$m = \frac{2E}{v_{r,max}^2}, \quad (43)$$

where $E > 0$ is the user defined kinetic energy term of the reference system. For all experimental tests, the potential field tuning constants used are listed in Table 3. We consider only cases with one or no obstacle presented. Furthermore, only obstacles within a 10 m radius are considered during force computation. For the testing results presented in this paper, $v_{r,max}$ is varied accordingly to achieve the desired overall speeds. We do not consider the effect of the potential fields beyond the reference system as the controller is designed to drive the vehicle to the reference system, given that the reference system is composed of class C^1 functions. To better understand the effects of potential fields, we refer the readers to [23–25], which better explore their behavior and usage.

5.3. Results. Each experiment uses the tuning constants laid out in Table 4. In addition, the parameter k_ω is varied to account for actuator dynamics in the steering. Moreover, $v_{r,max}$ is altered to achieve varying speeds for each experiment.

We begin by examining the basic tracking performance with no obstacles, presented below in Section 5.3.1. In the

TABLE 4: Common controller tuning parameters for experiments.

Tuning Parameters
$k_v = 1$
$\alpha = 1.3$
$\beta = 1$
$\lambda = 1$
$s = 0.01$
$\omega_d = 2.5$
$\zeta_d = 0.85$
$L = 0.3556$
$\phi_{max} = 25^\circ$
$v_{max} = 10$
$v_{min} = 1$

three scenarios that follow, the reference velocity is set to constant values of 2 m/s, 5 m/s, and 9 m/s. Each scenario uses the same set of waypoints, which the reference system follows based on the potential field setup presented earlier. Once the reference system is within a meter of the target waypoint, it switches its desired position to the subsequent point. In each of the three cases, the vehicle travels in a clockwise motion. The vehicle starts from rest in each case. We do not consider the exact coordinates of the initial position of the vehicle and reference system. Due to the nature of the experiment, this position is not repeatable and the effects of the initial conditions in these cases are inconsequential relative to the entirety of the path.

Following the tracking performance, we consider applications to obstacle avoidance in Section 5.3.2. Again, we consider three scenarios, a low, a medium, and a high speed case. For these tests, a specific set of waypoints is chosen as well as a single, constant obstacle. The target reference velocity is then varied to examine the behavior of the system. In these tests, data collection starts after the vehicle has reached a velocity relatively close to the intended reference velocity. This decision is to better focus on obstacle avoidance rather than the transient effects of accelerating to the target speed.

5.3.1. Experimental Tracking Performance. For the low speed tracking test, the additional tuning parameters not given in Table 4 are set to $k_\omega = 1$ and $v_{r,max} = 2$ m/s. The low speed test results are given in Figure 14. As expected, the vehicle (solid-red) tracks the reference trajectory (dashed-black) tightly. This is the typical behavior at low speeds, as the vehicle lags the reference system by a relatively small amount. While there are some slight deviations at the bottom and top of the graph, the overall performance is favorable in light of experimental results.

In Figure 15, we examine the response of the vehicle to a reference system that is moving at 5 m/s. All parameters are consistent with those in the previous example, with the exception of $k_\omega = 2.5$ and $v_{r,max} = 5$ m/s. At first glance, it is clear that the vehicle is cutting to the inside of the reference system's path, as seen in the bottom of the graph and in the top. This performance is ideal based on the derivation of the

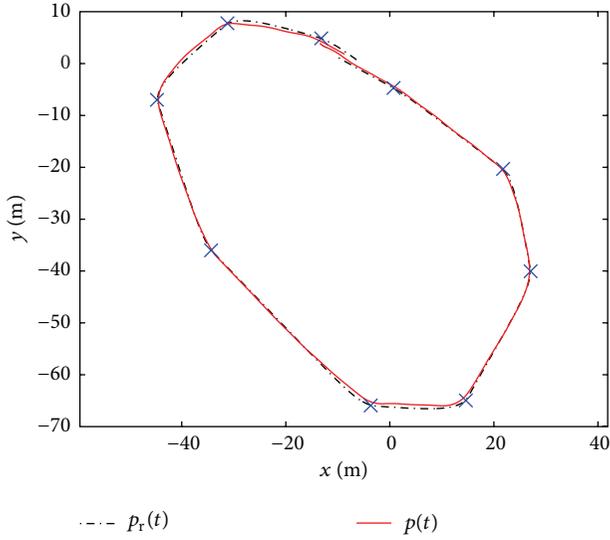


FIGURE 14: Tracking performance at 2 m/s. The crosses are desired waypoints.

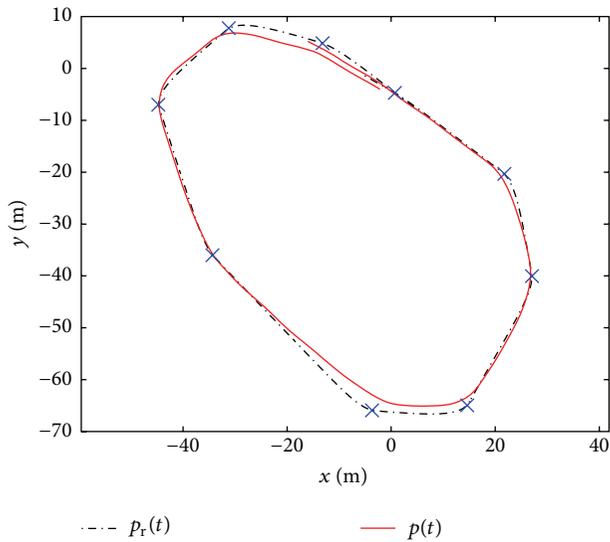


FIGURE 15: Tracking performance at 5 m/s. The crosses are desired waypoints.

controller. As the vehicle moves faster, the following distance will increase. As a result, the vehicle begins correcting its trajectory earlier. Consequently, the vehicle moves to the inside of the reference system's path, which is corroborated in experimentation.

While the vehicle stays largely within the reference system's path, in the upper left corner of both Figures 14 and 15 the vehicle travels outside the bound defined by the reference trajectory. These mild deviations are attributed to actuation dynamics. In derivation and simulation, we assume that the actuation of longitudinal and angular rates is instantaneous. In reality, there are dynamics that are crucial to the behavior of the system. When the controller commands a tight turn that is within the realm of reachable commands, the

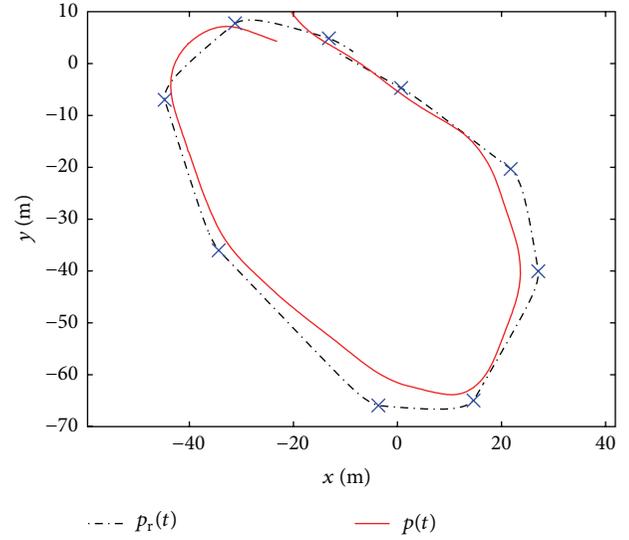


FIGURE 16: Tracking performance at 9 m/s. The crosses are desired waypoints.

dynamics may still keep the vehicle from responding within the necessary time frame.

In practice, these dynamic effects can be mitigated by adjusting the tuning parameters k_v and k_ω . By increasing k_ω for the medium speed case, the deviation is substantially less than if the gains would be kept constant from the low speed case.

We refrain from holding k_ω at high values for all tests because the dynamics are also influential at low speeds. With high values of k_ω at low speeds, significant oscillation is seen in the commands, although tracking is largely unchanged.

Next, we examine the tracking performance at high speeds, the results of which are given in Figure 16. Again, all parameters are held constant according to Table 4, with the exception of $k_\omega = 3.5$ and $v_{r_{\max}} = 9$ m/s. This test appears as a more extreme case of the medium speed result. In fact, the vehicle's path remains well within the bound laid out by the reference system aside from the initial straightaway and the last turn. As with the medium speed case, the deviation on the last turn is attributed to actuator dynamics, which are not fully compensated for but mitigated by higher values of k_ω .

5.3.2. Application to Obstacle Avoidance. The control algorithm presented can be conceptualized as a position filter, which smoothly follows a reference system despite sudden changes in direction. This behavior elicits the predator-prey interaction sought after. Potential fields, on the other hand, have natural obstacle avoidance properties, but they lack smoothness, which is particularly necessary for high speed situations. Using a combination of the two, we examine the performance of the given approach in the hope of finding more intuitive behavior.

Figures 17 and 18 examine the case of obstacle avoidance at low speed using the same parameters laid out in Table 4, with $k_\omega = 1$ and $v_{r_{\max}} = 2$ m/s. In Figures 17, 19, and 21, the triangles and circles denote the location of the reference

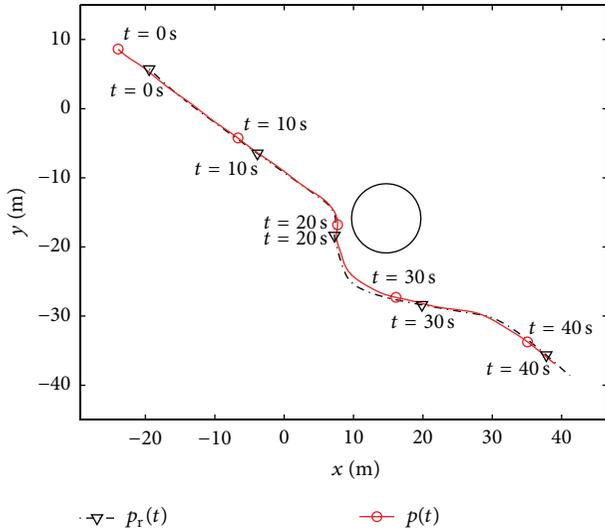


FIGURE 17: Obstacle avoidance at low speeds. Triangles and circles denote the location of reference system and vehicle, respectively, at certain instances in time.

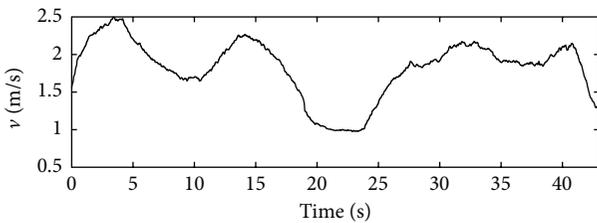


FIGURE 18: Velocity of obstacle avoidance at low speeds.

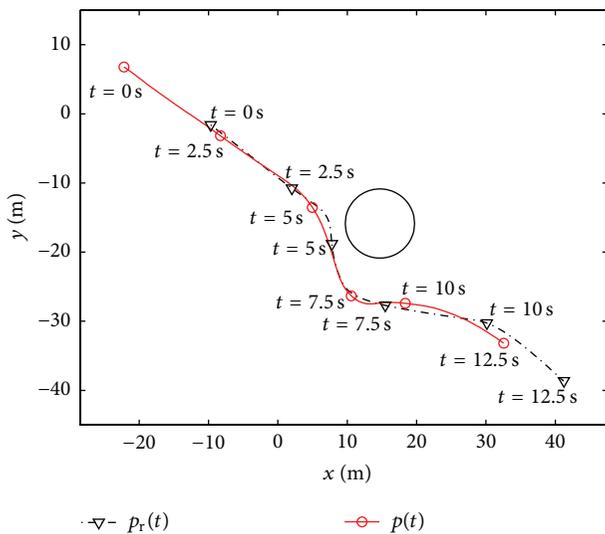


FIGURE 19: Obstacle avoidance at medium speeds. Triangles and circles denote the location of reference system and vehicle, respectively, at certain instances in time.

system and vehicle, respectively, at certain instances in time. Looking at Figure 17, the vehicle tightly tracks the reference

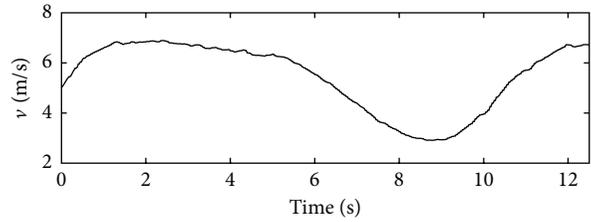


FIGURE 20: Velocity of obstacle avoidance at medium speeds.

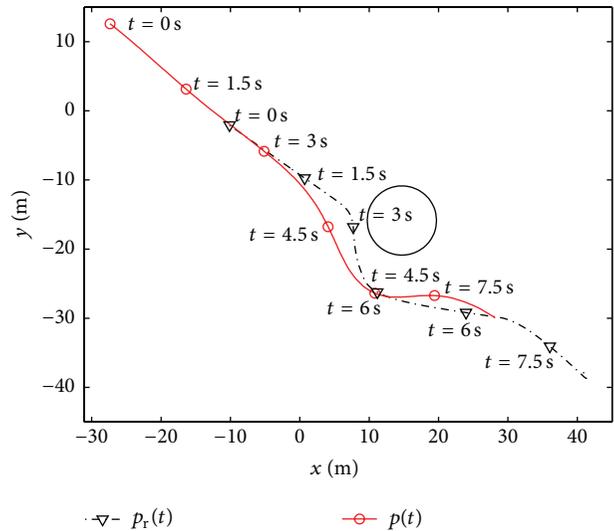


FIGURE 21: Obstacle avoidance at high speeds. Triangles and circles denote the location of reference system and vehicle, respectively, at certain instances in time.

system despite a sharp, awkward corner around the obstacle. Since the speed is small, seen in Figure 18, this behavior is acceptable. Of particular interest is the fact that the velocity drops to only 1 m/s as the system avoids the obstacle. This is a natural result of the potential field system, which we utilize to obtain an intuitive response from the platform. Additionally, we note that this lower velocity is also bounded by the saturation algorithm which maintains at least a 1 m/s speed. Consequently, by examining the time stamps of Figure 17, we see that the distance between the reference and vehicle reduces during the cornering, allowing the vehicle to better track the turn.

The results of obstacle avoidance at medium speeds are given in Figures 19 and 20. Here the tuning gains are altered such that $k_\omega = 2.5$ and $v_{r,max} = 6$ m/s. As with the medium speed case in the tracking study, the vehicle does not follow the reference system as tightly. This behavior, however, has distinct advantages in dealing with obstacles. At the 5-second mark, the vehicle begins to diverge from the reference path in order to avoid the obstacle, as opposed to waiting until the last instant. By steering earlier, the vehicle achieves a less severe curve during avoidance, facilitating higher speeds throughout. With regard to the predator-prey analogy, this behavior is more intuitive than basic potential fields. Rather than directly mimicking the prey's path, the predator adjusts

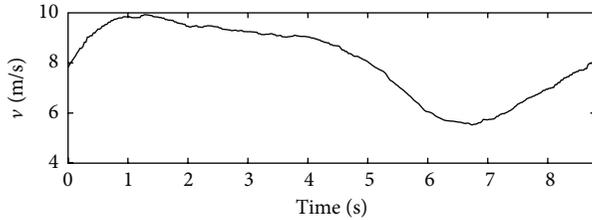


FIGURE 22: Velocity of obstacle avoidance at high speeds.

its path to easily close in on the prey. The intuition naturally follows for humans as well, as a human driver would avoid immediately upon the observation of an obstacle, rather than waiting. The human's path of avoidance would likewise be based on speed. Looking at Figure 20, the speed decreases from 6 m/s to 3 m/s during the turn. Again, this is a natural behavior as the average driver would naturally slow down in order to safely avoid the obstacle. By slowing down, the distance decreases, which then facilitates better tracking of the reference system.

Lastly, we consider the case of obstacle avoidance at high speeds, seen in Figures 21 and 22. For this case, the tuning gains are altered such that $k_\omega = 2.5$ and $v_{r,\max} = 9$ m/s. Looking at Figure 21, the algorithm clearly smooths the trajectory of the reference system. In the same way a predator would make milder turns at high speeds, the vehicle departs from the reference path at approximately 3 seconds, at a point much earlier than the reference system begins to avoid. During this outward swerve, Figure 22 indicates that the velocity begins to drop off at 5 seconds. This drop results in a decrease from 9 m/s to 6 m/s, a 33% dip opposed to the 50% change in the previous scenario. With a smoother trajectory, the algorithm allows for more preservation of speed than previously seen. Lastly, we note that the distance between the reference system and vehicle decreases, allowing for a more aggressive turn after passing the obstacle.

6. Conclusions

Many control algorithms currently in existence are largely effective but lack the simplicity of intuitive behavior native to most people. With the increasing push for driverless technologies, however, it is becoming more important to develop algorithms that exploit the trends expected by everyday users in high speed situations. To this end, we consider nature's predators for inspiration. Specifically, our bioinspired model references the cheetah on several occasions. We note that a variable distance to its prey allows it to better plan its path while maintaining a high speed. Additionally, we assume natural constraints on its motion that also hold favorable characteristics for fast maneuvering.

In an effort to reproduce this behavior, a control algorithm, based on a unicycle model, is introduced. This algorithm focuses on driving the current vehicle to within some variable following distance of a user defined reference system by commanding longitudinal and angular rates. This algorithm is proven stable through Lyapunov criteria.

Due to the limitations of the unicycle model, we are not able to directly achieve the sought after motion constraints of the predator. As a result, additional saturation constraints are implemented to enforce Ackermann steering kinematics on the system's behavior. The saturation constraints are presented as a two stage algorithm. The first stage, which is not a sufficient method within itself, focuses on preventing chatter in the final command. The second stage receives commands from the first stage and maps them within an achievable envelope. From the saturation methodologies, the reference system is then manipulated to satisfy the Lyapunov criteria. Based on simulations, the dual stage saturation algorithm exhibits the desired intuitive behavior.

Using potential fields to control the reference system, the controller is implemented on a small scale robotic vehicle. Testing results thus far indicate that the proposed controller, coupled with a potential field algorithm, exhibits strong intuitive behaviors similar to those described by the predator-prey relationship. At low speeds, the vehicle tracks the reference system within a close tolerance of the reference system's path. This trend holds in the case of obstacle avoidance. However, at high speeds, the vehicle diverges from the set path and instead smooths the overall trajectory, allowing some conservation of speed without extremely aggressive steering, much as would be expected from a human controller. Overall, the algorithm was used to successfully navigate trajectories at up to 9 m/s.

There are several instances in which the dynamics prevent the vehicle from consistently following the idealized path. Consequently, future work will focus on including these effects through backstepping or other related methods. Also, the algorithm will be refined for applicability to multiple platforms, most likely through adaptive methods.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This material is based upon the work performed while Dr. Alexander Leonessa was serving at the National Science Foundation of the United States. The authors would also like to thank Virginia Tech's Open Access Subvention Fund for their support in publishing this work.

References

- [1] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1599–1616, 2012.
- [2] S. Moon, U. Lee, and D. H. Shim, "Study on real-time obstacle avoidance for unmanned ground vehicles," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '10)*, pp. 1332–1335, October 2010.
- [3] D. Fu-guang, J. Peng, B. Xin-qian, and W. Hong-jian, "AUV local path planning based on virtual potential field," in *Proceedings*

- of the *IEEE International Conference on Mechatronics and Automation (ICMA '05)*, vol. 4, pp. 1711–1716, August 2005.
- [4] R. Daily and D. M. Bevely, “Harmonic potential field path planning for high speed vehicles,” in *Proceedings of the American Control Conference*, pp. 4609–4614, June 2008.
 - [5] P. Zhao, J. Chen, T. Mei, and H. Liang, “Dynamic motion planning for autonomous vehicle in unknown environments,” in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '11)*, pp. 284–289, Baden-Baden, Germany, June 2011.
 - [6] 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.
 - [7] M. Häselich, N. Handzhiyski, C. Winkens, and D. Paulus, “Spline templates for fast path planning in unstructured environments,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11)*, pp. 3545–3550, IEEE, San Francisco, Calif, USA, September 2011.
 - [8] C. Fulgenzi, A. Spalanzani, and C. Laugier, “Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 1610–1616, April 2007.
 - [9] A. Richardson and E. Olson, “Iterative path optimization for practical robot planning,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11)*, pp. 3881–3886, September 2011.
 - [10] P. Bhattacharya and M. L. Gavrilova, “Voronoi diagram in optimal path planning,” in *Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD '07)*, pp. 38–47, July 2007.
 - [11] K. P. Valavanis, T. Hebert, R. Kolluru, and N. Tsourveloudis, “Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field,” *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 30, no. 2, pp. 187–196, 2000.
 - [12] M. Park and M. Lee, “Experimental evaluation of robot path planning by artificial potential field approach with simulated annealing,” in *Proceedings of the 41st SICE Annual Conference*, vol. 4, pp. 2190–2195, Osaka, Japan, August 2002.
 - [13] X. Hu, “Clonal selection based mobile robot path planning,” in *Proceedings of the IEEE International Conference on Automation and Logistics (ICAL '08)*, pp. 437–442, September 2008.
 - [14] A. M. Wilson, J. C. Lowe, K. Roskilly, P. E. Hudson, K. A. Golabek, and J. W. Mc-Nutt, “Locomotion dynamics of hunting in wild cheetahs,” *Nature*, vol. 498, no. 7453, pp. 185–189, 2013.
 - [15] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot: backstepping kinematics into dynamics,” *Journal of Robotic Systems*, vol. 14, no. 3, pp. 149–163, 1997.
 - [16] D. DeVon and T. Bretl, “Kinematic and dynamic control of a wheeled mobile robot,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 4065–4070, San Diego, Calif, USA, November 2007.
 - [17] K. Pathak and S. K. Agrawal, “Planning and control of a non-holonomic unicycle using ring shaped local potential fields,” in *Proceedings of the American Control Conference (AAC '04)*, pp. 2368–2373, July 2004.
 - [18] Q. Wang and Y.-P. Tian, “Formation control of multiple non-holonomic unicycles using adaptive perturbation method,” in *Proceedings of the 51st IEEE Conference on Decision and Control (CDC '12)*, pp. 2453–2458, December 2012.
 - [19] L. C. Figueiredo and F. G. Jota, “Nonholonomic systems stabilization and tracking control using discontinuous control,” in *Proceedings of the IEEE Conference on Automation Science and Engineering (CASE '05)*, pp. 142–147, August 2005.
 - [20] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, “Closed loop steering of unicycle like vehicles via Lyapunov techniques,” *IEEE Robotics & Automation Magazine*, vol. 2, no. 1, pp. 27–35, 1995.
 - [21] R. Carona, A. P. Aguiar, and J. Gaspar, “Control of unicycle type robots,” in *IV Jornadas de Engenharia Electronica e Telecomunicacoes e de Computadores*, pp. 180–185, November 2008.
 - [22] V. Medina-Garciadiago and A. Leonessa, “Tracking control of a nonholonomic ground vehicle,” in *Proceedings of the American Control Conference (ACC '11)*, pp. 1710–1713, IEEE, San Francisco, Calif, USA, June-July 2011.
 - [23] J. Borenstein and Y. Koren, “High-speed obstacle avoidance for mobile robots,” in *Proceedings of the 3rd International Symposium on Intelligent Control*, pp. 382–384, August 1988.
 - [24] J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989.
 - [25] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 1398–1404, April 1991.

Research Article

Unified Switching between Active Flying and Perching of a Bioinspired Robot Using Impedance Control

Shanshan Du,¹ Heping Chen,² Yong Liu,¹ and Runting Hu¹

¹*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*

²*Ingram School of Engineering, Texas State University, San Marcos, TX 78666, USA*

Correspondence should be addressed to Yong Liu; liuy1602@njjust.edu.cn

Received 23 January 2015; Revised 19 June 2015; Accepted 20 August 2015

Academic Editor: Maki K. Habib

Copyright © 2015 Shanshan Du 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.

Currently, a bottleneck problem for battery-powered microflying robots is time of endurance. Inspired by flying animal behavior in nature, an innovative mechanism with active flying and perching in the three-dimensional space was proposed to greatly increase mission life and more importantly execute tasks perching on an object in the stationary way. In prior work, we have developed some prototypes of flying and perching robots. However, when the robots switch between flying and perching, it is a challenging issue to deal with the contact between the robot and environment under the traditional position control without considering the stationary obstacle and external force. Therefore, we propose a unified impedance control approach for bioinspired flying and perching robots to smoothly contact with the environment. The dynamic model of the bioinspired robot is deduced, and the proposed impedance control method is employed to control the contact force and displacement with the environment. Simulations including the top perching and side perching and the preliminary experiments were conducted to validate the proposed method. Both simulation and experimental results validate the feasibility of the proposed control methods for controlling a bioinspired flying and perching robot.

1. Introduction

With the increasing missions requirements in the air space, the flying robots have been widely used in all kinds of fields. In the military fields, flying robots are mainly developed for reconnaissance and surveillance, communication relay, and electronic interference, as well as biochemical detection. In the civil aspects, they are employed to accomplish atmospheric monitoring, resource exploration, traffic monitoring, aerial photography, and so on [1–3]. For such applications, flying robots are difficult to meet the requirement because of their low power efficiency. For example, the requirement of the surveillance missions mostly needs to take more than one hour; however, current flying robots have to keep flying which decreases the endurance of battery dramatically. Typical battery-powered microflying robots can only fly in tens of minutes. Thus, power consumption for flying robots is a challenging issue.

An attractive approach is to enable the flying robots to do what flying animals do: taking frequent short flights with

periods of perching in between. The energy efficiency of such animals is much higher than that of flying robots.

In particular, it is useful for small fixed-wing planes to perch on vertical surfaces such as cliffs or the walls of buildings [4]. Clinging to such surfaces, they consume little power, allowing them to stay there for hours or possibly days as a stable platform for surveillance, inspection, or environmental monitoring. But the application is greatly limited on the brick walls and the process is hardly controllable.

Anderson et al. have proposed various ideas for perching, such as perching with adhesives and weighted tail line [5]. Kovač et al. have designed a simple and practical perching mechanism based on the concept of the target surface using needles [6]. Chi et al. have proposed a design of bioinspired adaptive perching mechanism, and it mainly focused on the static characteristics [7]. Still other works have focused on performing perching maneuvers using a morphing airplane [8, 9].

However, existing methods focus on passive way to perch on the surface. Although they consume little or no power,



FIGURE 1: The developed robot prototypes. (a) is top perching robot with an adsorption device consisting of motor, spinning rotor, fixture, and flexible sponge skirt seal. (b) is side perching with suction cup driven by micropump.

switching processes are not stable and uncontrollable, the payload is too small, or specified objects are required so that the applications are greatly limited in laboratory. The flying and perching robot, which can fly in the sky and anchor on a wall surface alternately, has the advantages of both long distance motion with the ability to overcome obstacles and long time stop with low energy consumption. In our prior work, prototypes of active flying and perching robots we proposed [10, 11], as shown in Figure 1, are promising in many applications. Figure 1(a) is the prototype of our developed generation one top perching robot adsorbed on the ceiling and mechanism of the adsorption device is based on vortex attraction technique. This mechanism allows the robot to adsorb on various smooth and rough surfaces. Figure 1(b) is the side perching robot perching on the window glass.

The existing challenging issue is to control the switching process between flying and perching on an object surface that may cause robots breakdown. Traditional position control methods [12–14] of the flying robot do not consider the force and the displacement constraint from the environment. When the robot contacts with the environment, because a force balance has been broken and a stationary obstacle appears, the original dynamic model, the planned trajectory, and the position controller will be completely ineffective. The approach we employed before is to make the robot approach the environment at certain velocity; thus, the external force will be compensated. But the problem is that it is difficult to decide how much velocity should be given to the robot. If it is too large, it could hit the environment and damage the robot; but if it is too small, its attitude will be changed by the external force, and robots cannot generate the enough force for adsorption. In this paper, we propose a model based impedance control method to address the issue for the bioinspired active flying and perching robot. The control system performs not only position and attitude control but also force control in order to interact with the environment. Force control of robotic manipulators has been an active area of research for many reasons [15–18]. We extend the impedance

control to our developed active flying and adhesion robot control system, which has an active adsorption device and the adhesion force and switching process are controllable so that it is suitable for all kinds of objects surfaces and applications.

This paper is structured as follows. In Section 2, the strategy of the bioinspired flying and perching robot is introduced and the dynamic model of the robot is described, especially considering the situation that the mass center and the geometric center may not be coincident. In Section 3, impedance force control system of the robot is designed and analyzed. In Section 4, simulations for top perching and side perching and experiments based on the prototype we developed were performed. The results are analyzed. Section 5 concludes the work and discusses future directions of our research.

2. Robot Dynamics

2.1. The Strategy. The motivation for our design is that the robot could execute the missions with the ability of flying and the ability of actively perching on most objects surfaces in the three-dimension space with more payloads and obstacle-free under lower power consumption. The general sequence of the bioinspired active flying and perching robot, illustrated in Figure 2, is as follows: (1) fly in the sky, (2) approach the target perching object surface, (3) contact the environment and regulate the contact force to achieve desired contact force, (4) adsorption device in the robot starts to work and make it adhere to the object surface steadily, (5) stop flying and anchor on the object surface with low power consumption, (6) execute the missions such as reconnaissance and surveillance, and (7) switching from the perching to flying is an inverse process.

The focus of our work in the paper is the interaction with environment in step (3) and the goal of the step is to control the bioinspired robot switch between the flying and the perching actively and smoothly in natural way like a flying animal behavior.

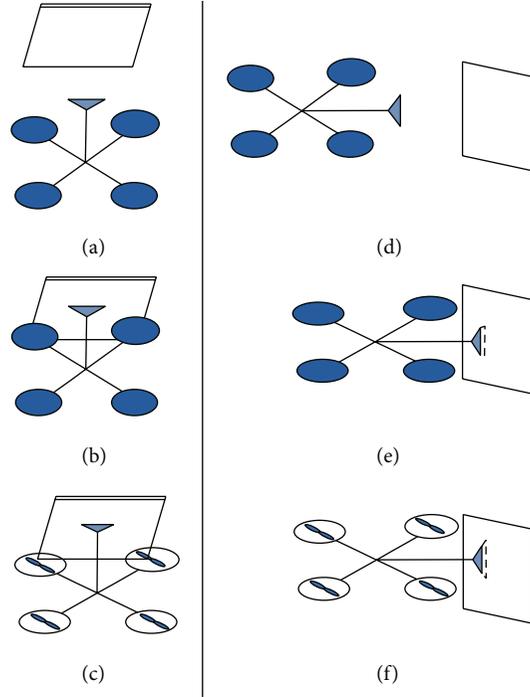


FIGURE 2: Top perching and side perching: the robot performs several phases when switching from flying to perching. The top perching robot equipped with a quadrotor and suction cup flies to the ceiling and hangs on the ceiling. (a) Top perching approaching step, (b) top contact step, and (c) top perching rest step. The side perching robot with suction cup flies and perches on the wall. (d) Side perching approaching step, (e) side contact step, and (f) side perching rest step.

2.2. Dynamics Model. The generalized coordinates of the robot system are $(x, y, z, \psi, \theta, \varphi)$ where (x, y, z) represents the relative position with respect to an inertial frame and (ψ, θ, φ) represents the orientation of the robot using Z-Y-X Euler angles, namely, yaw, pitch, and roll. Let $\{A\}$ denote the inertial frame and $\{B\}$ the body frame. The frame $\{B\}$ is related to $\{A\}$ by a 3 by 3 coordinate rotation matrix $R : B \rightarrow A$, which is defined to map a vector expressed in the frame $\{B\}$ to the equivalent vector expressed in the frame $\{A\}$. The translational and rotational variables are defined as $\varepsilon = \{x \ y \ z\}^T \in \{A\}$ and $\eta = \{\varphi \ \theta \ \psi\}^T \in \{A\}$, respectively, shown in Figure 3. The rotation matrix $R : B \rightarrow A$ is given by

$$R = \begin{bmatrix} c_\theta c_\psi & s_\theta c_\psi s_\varphi - s_\psi c_\varphi & s_\theta c_\psi c_\varphi + s_\psi s_\varphi \\ c_\theta s_\psi & s_\theta s_\psi s_\varphi + c_\psi c_\varphi & s_\theta s_\psi c_\varphi - c_\psi s_\varphi \\ s_\theta & c_\theta s_\varphi & c_\theta c_\varphi \end{bmatrix}. \quad (1)$$

The i th rotor has an angular speed ω_i and generates a force F_i and moment M_i according to

$$\begin{aligned} F_i &= k_F \omega_i^2, \\ M_i &= k_M \omega_i^2, \end{aligned} \quad (2)$$

where k_F is force stiffness and k_M is moment stiffness.

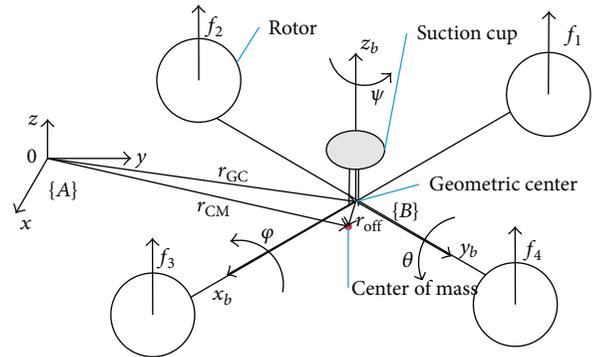


FIGURE 3: Free body diagram and coordinate systems.

The position vectors of the center of mass r_{CM} in frame A and geometric center r_{GC} in frame A are related to the geometric relationship according to

$$r_{CM} = r_{GC} + R r_{off}, \quad (3)$$

where $r_{off} = [x_{off}, y_{off}, z_{off}]^T$ are offsets of the center of mass in frame B , which is calculated by the system design. The

linear velocity of geometric center of the robot in the frame A is expressed as follows:

$$V = \dot{r}_{GC} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}. \quad (4)$$

And the angular velocity of the robot body frame B in the inertial frame A is denoted by Ω , with the components p , q , and r in the body frame B . The dynamic model of the bioinspired robot can be written in the position subsystem by Newton's equation of motion acting on the geometric center

$$\begin{aligned} (m + m_0) (\dot{V} + R(\dot{\Omega} \times r_{\text{off}}) + R(\Omega \times (\Omega \times r_{\text{off}}))) \\ = (0, 0, -(m + m_0)g)^T + F_m + F_e \end{aligned} \quad (5)$$

and in the rotational subsystem by Euler equation

$$I_{CM} \dot{\Omega} = -\Omega \times I_{CM} \Omega + \begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} - r_{\text{off}} \times \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix}. \quad (6)$$

The force vector F_m can be calculated:

$$F_m = R \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix}, \quad (7)$$

where u is the system control input which can be defined using the total thrust $u = \sum_{i=1}^4 F_i$ and the torques $\tau = (\tau_\varphi \ \tau_\theta \ \tau_\psi)^T$; m is the total mass of the rotor system; m_0 is the mass of the battery and perching device; I_{CM} is the moment of inertia matrix referenced to the center of mass along the x -, y -, and z -axis in the frame B . F_e is the external force.

The two terms $(m + m_0)(\dot{\Omega} \times r_{\text{off}})$ and $(m + m_0)(\Omega \times (\Omega \times r_{\text{off}}))$ in (5) can be compensated by the feedback. This is because these terms are generally small relative to other terms in practice [19]. Meanwhile, for the side perching robot, the asymmetry is mainly caused by the installation of the perching device. Therefore, the disequilibrium is primarily made along the X and Y directions.

In summary, the dynamic equation in the inertial frame under the assumption of neglecting Coriolis terms and gyroscopic effects yields

$$\begin{aligned} (m + m_0) \ddot{x} &= -us_\theta - F_{ex}, \\ (m + m_0) \ddot{y} &= uc_\theta s_\varphi - F_{ey}, \\ (m + m_0) \ddot{z} &= uc_\theta c_\varphi - F_{ez} - (m + m_0)g, \end{aligned} \quad (8)$$

$$\begin{pmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = I_{CM} \begin{pmatrix} \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} + \begin{pmatrix} y_{\text{off}} u \\ -x_{\text{off}} u \\ 0 \end{pmatrix}, \quad (9)$$

where $s_{(\cdot)} \triangleq \sin(\cdot)$, $c_{(\cdot)} \triangleq \cos(\cdot)$, $F_{ex} = F_e c_\alpha c_\psi c_\theta - F_e s_\alpha c_\theta$,

$$\begin{aligned} F_{ey} &= F_e c_\alpha (c_\psi s_\theta s_\varphi - s_\psi s_\varphi) - F_e s_\alpha (s_\psi s_\theta s_\varphi - c_\psi c_\varphi), \\ F_{ez} &= F_e c_\alpha (c_\psi s_\theta c_\varphi + s_\psi s_\varphi) - F_e s_\alpha (s_\psi s_\theta c_\varphi - c_\psi s_\varphi). \end{aligned} \quad (10)$$

There are six variables to represent the system; we have only four inputs. Therefore, the system is an underactuated system. Since the motor response is quick compared to that of the robotic system, we assume that rotor speed response can be instantly achieved during the controller development. Therefore, the control inputs can be expressed in terms of the rotor speeds as follows:

$$\begin{pmatrix} u \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = C_R \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix}, \quad (11)$$

$$C_R = \begin{pmatrix} k_F & k_F & k_F & k_F \\ k_F l & 0 & -k_F l & 0 \\ 0 & -k_F l & 0 & k_F l \\ k_M & -k_M & k_M & -k_M \end{pmatrix}, \quad (12)$$

where l is the distance from the rotational axes of the rotors to the center of the robot. C_R is the translational matrix. The angular speed of each rotor can be calculated using (11) and (12).

3. Impedance Control

In this section, in order to let the robot fly to a desired perching object, contact the object surface with a desired force, and adsorb on a surface steadily, the impedance controller for this underactuated bioinspired flying and perching robot is introduced.

Direct application of traditional position controller designed for free space robot motion usually results in instability and eventually lets the robot break down when the robot is in contact with environment, because this controller ignores the interaction between the robot and the environment, which may generate external force and displacement block. Impedance control allows us to deal with the external force and displacement simultaneously under interaction with the environment. The inclusion of force information in the control of this kind of bioinspired flying and perching robots will increase their adaptability to uncertain environments and provide safety against damage due to excessive contact force.

Suppose ε_d and F_d are the desired robot motion and interaction force, and ε_d is assumed to be twice continuously differentiable. Let

$$\begin{aligned} e &= \varepsilon_d - \varepsilon, \\ e_f &= F_d - F_e \end{aligned} \quad (13)$$

be the tracking errors of motion and the interaction force, respectively; the impedance control law can be described as

$$M\ddot{e} + D\dot{e} + Ke = -K_f e_f, \quad (14)$$

where the generalized impedance parameters M , D , and K are specified as positive definite diagonal matrices and K_f is the force stiffness matrix.

Following (14), the controller can yield based on some different situations.

Case 1. In the free space control, the control law can be obtained as follows:

$$M\ddot{e} + D\dot{e} + Ke = -K_f F_d. \quad (15)$$

Note that when the desired force F_d is set to zero and $M = \text{diag}(1, 1, 1)$, then the robot is under position control:

$$\ddot{e} = \ddot{e}_d + D\dot{e} + Ke. \quad (16)$$

Then, the desired total thrust and desired attitude u_d, θ_d, φ_d can be also obtained using (8) and (9).

Case 2. When the robot comes close to the environment, we hope it can produce a steady force against the external environment; the control law can be obtained as follows:

$$\ddot{e} = \ddot{e}_d + \frac{1}{M} (D\dot{e} + Ke + K_f (F_d - F_e)). \quad (17)$$

Case 3. The special case of Case 2 is used for the top perching; in this case, the direction of z is controlled by impedance control and the other two directions are controlled by position control. Combining the control laws (20) and (12), impedance controller in the direction of z can be obtained:

$$\ddot{z} = \ddot{z}_d + \frac{1}{M_z} (D_z \dot{e}_z + K_z e_z + K_{fz} (F_{dz} - F_{ez})). \quad (18)$$

4. Simulation and Experiment

4.1. System Setup. To accomplish some missions such as reconnaissance and surveillance, the robot is required to carry a certain payload, adsorb on a surface actively, and spend low power consumption. Figure 4 shows the real bioinspired side perching robot we have designed and developed. The prototype of the bioinspired flying and perching robot is based on the quadrotor structure [20], which also

includes an adsorption device, a supporting structure, and an embedded controller. The perching device we employed here is based on vacuum generated by micropumps which mainly depends on a membrane, which is characterized with low power consumption, low noise, and light weight. Meanwhile, it can be very convenient to be installed due to its small size. The supporting system, on the one hand, is used to improve the payload when the robot anchors on a wall surface; meanwhile, it plays a role in protection; on the other hand, is equipped with a spring and potentiometer, which can be used to measure the contact force as feedback.

Figure 5 shows the control block diagram of the torque-based force control system for simulation. Two kinds of simulations were performed: top perching and side perching. Table 1 lists the parameters of the bioinspired robot used for simulation. The impedance parameters are shown in Table 2, where K_f is for the situation of top perching.

In the simulation, the impedance control block uses the relationship of displacement error and force error (14) to calculate the new control variable. In the top perching, only the vertical force and displacement are considered; there are no displacement and movement in the other directions. And the total thrust and angles in the second block are calculated by using (8) and (18). Torque from attitude angle is calculated using (9) and then the four rotor speeds are calculated according to the torque and the four thrust forces in (11) and are output to robot system block.

In the real system, the feedback data comes from the measurement of sensors, but in the simulation, inverse kinematics are used to simulate the movement of the robot. The force generated by four rotors is calculated based on the proportional relationship. The displacement and attitude angle are computed by integration. Because the data is obtained by virtual computing instead of collecting from real system, without taking into account the system error, the ideal result should be completely tracking the desired data. In simulation system, according to the real system, the sampling time is set to 0.01 s.

To verify the reliability and stability, without loss of generality, we add noise into simulation, which exists in real system. In the sensor block, we put white Gaussian noise into the block.

4.2. Top Perching. The robot is controlled to adhere to the ceiling of a room. The 5% deviation is added to the nominal values of all parameters shown in Table 1. White noises are also deployed in the sensor measurement to simulate the real situation. The desired motion trajectory along the Z direction is given as follows (the unit is m):

$$z_d = \begin{cases} 0.096t^3 - 0.01435t^4 + 0.000573t & t < 10 \\ 9.8 + 0.05(t - 10) & t < 11 \\ 9.823 + 0.02667(2.5(t - 10)^3 - 1.875(t - 10)^4 + 0.375(t - 10)^5) & t < 12 \\ 9.876 & t \geq 12, \end{cases} \quad (19)$$

$$x_d = y_d = 0.$$

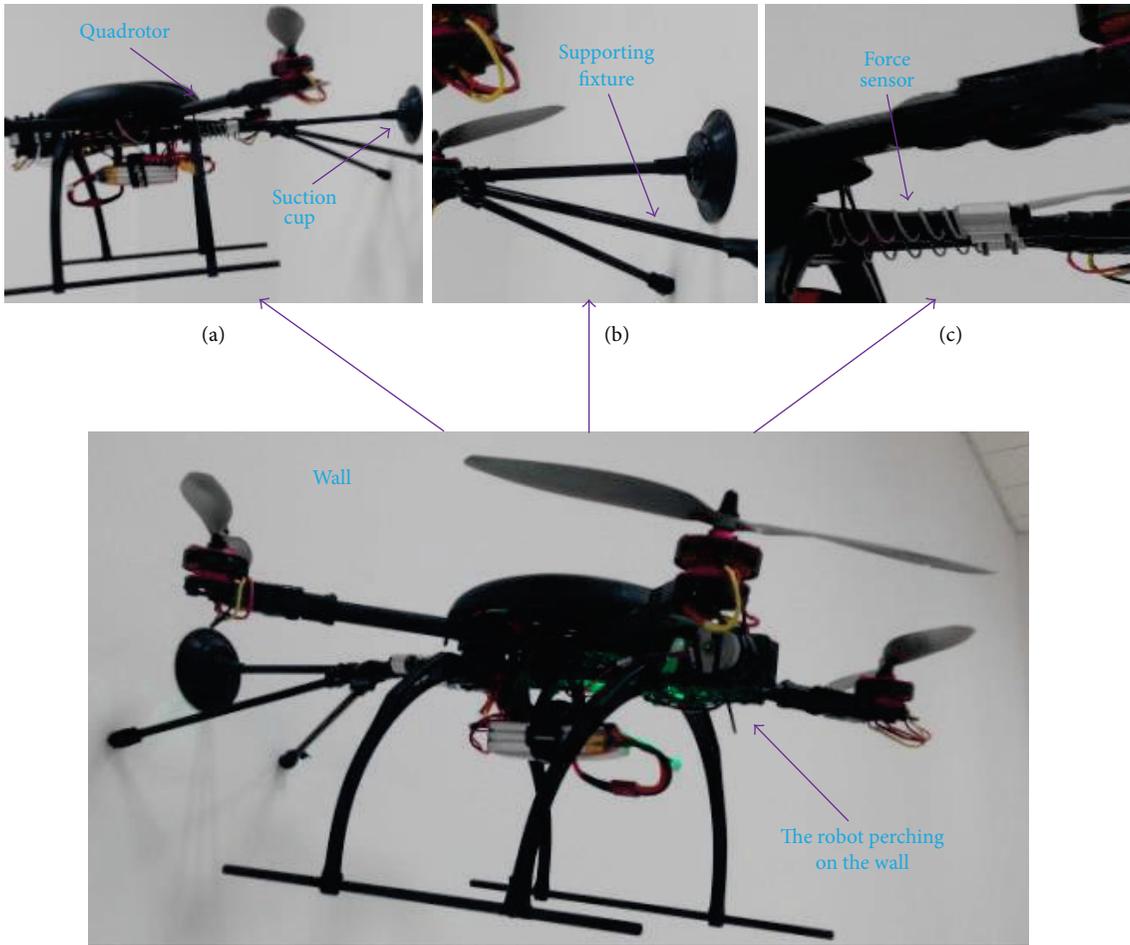


FIGURE 4: The prototype of the side perching robot. (a), (b), and (c) are views of local components of the robot showing the details when it rests on the wall. A suction cup seals with the wall which generates the vacuum chamber. The supporting system shown in (b) is well designed to provide more payload. A slider structure with spring and potentiometer is well designed to measure the external force.

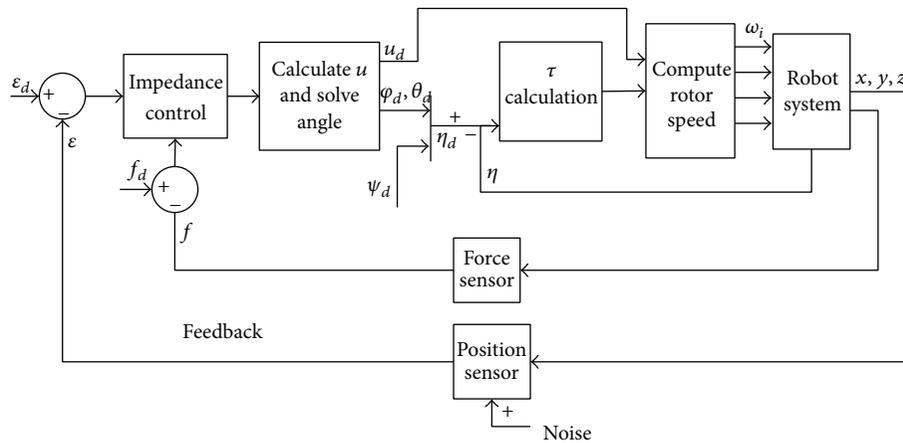


FIGURE 5: System schematic for impedance control system.

TABLE 1: Parameters of the robot system.

Variables	Values
Total mass m [Kg]	0.65
Micropump mass m_0 [Kg]	0.15
Frame size [cm*cm]	40 × 50
Mass center offsets [cm] [grams]	(4, 4, 0)
Angle between suction cup and lateral rotors α [rad]	$\pi/4$

TABLE 2: Impedance parameters.

M	D	K	K_f
{8, 9, 10}	{50, 50, 30}	{200, 200, 400}	{0, 0, 20}

And the desired force is given:

$$F_{dz} = \begin{cases} 0 & z_d \leq 9.8 \\ K_e (z_d - 9.8) & z_d > 9.8, \end{cases} \quad (20)$$

$$f_{dx} = f_{dy} = 0,$$

where K_e is preset by the stiffness of the environment.

Figures 6 and 7 show the impedance control simulation results for the top perching. During the free flight, the robot tracks the desired position to approach the environment.

Because the parameter uncertainties are added to the system, the maximum tracking error is about 2.3 cm. In the free space, the desired force is set to zero. The robot starts to contact with the ceiling at around 10 s and follows the desired force reference very well from Figure 7. The maximum force error is about 0.4 N and stabilized at 0.2 N. Figure 6 shows the position tracking result along the Z direction. After 12 s, the system settled down and the tracking error is about 0.2 cm.

4.3. Evaluation. The traditional one of control techniques tracking a desired motion trajectory for quadrotors is PD controller. An investigation is conducted to evaluate the performance of the developed impedance control and PD controller while performing the top perching which is the same as Section 4.2. During step (3) of contact with the environment, the flying robot will encounter the ceiling of high stiffness at a position about 9.8 m from the ground. The position tracking error and the interaction force along the z direction are shown in Figures 6, 7, 8(a), and 8(b), respectively. The larger interaction force is generated with the pure position PD control. The measured force continues to increase at the time of 11 seconds where $F_e = 33$ N which will cause the flying robot to fall down. The results demonstrate that the impedance control can handle the external contact without generating excessive interaction forces.

4.4. Side Perching. The desired trajectory is as follows:

$$x_d = y_d = \begin{cases} 0.096t^3 - 0.01435t^4 + 0.000573t & t < 10 \\ 9.8 + 0.05(t - 10) & t < 11 \\ 9.823 + 0.02667(2.5(t - 10)^3 - 1.875(t - 10)^4 + 0.375(t - 10)^5) & t < 12 \\ 9.876 & t \geq 12, \end{cases} \quad (21)$$

$$z_d = 10.$$

And the desired force is given:

$$F_d = \begin{cases} 0 & x_d \leq 9.8 \\ K_e (x_d - 9.8) & x_d > 9.8. \end{cases} \quad (22)$$

The robot is controlled to perform side perching following the planned trajectory and force reference.

The parameter deviation and white noise are also added to the system. The perching device is placed in the center of x -axis and y -axis, so the desired trajectory along the X direction and Y direction is the same as that along the Z direction in the top perching. The desired trajectory along the Z directions is 0. The desired force is the same as that used in the top perching. In the side perching, the

tracking errors along both the x -axis and y -axis are generated because the robot has to tilt a little bit to approach a side wall in a room. There are also orientation tracking errors. Figure 9 shows the tracking errors along the X and Y directions.

Because it is more challenging to make the robot perching on the side wall, the force tracking error is bigger than that in the top perching. Figure 10 shows the force tracking error. The maximum tracking error is about 1 N. The orientation errors are also recorded and shown in Figures 11 and 12. The maximum orientation tracking error is about 0.05 rad.

4.5. Sensors Testing. The ultrasonic sensors are employed to measure the distance between the quadrotor and the

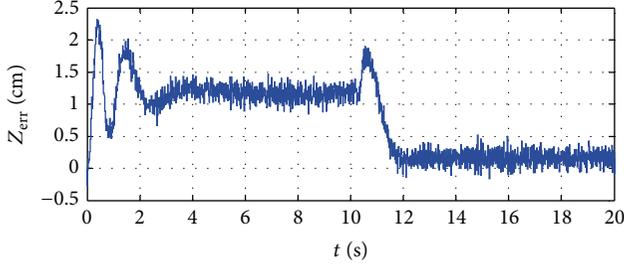


FIGURE 6: Position tracking error along the Z direction.

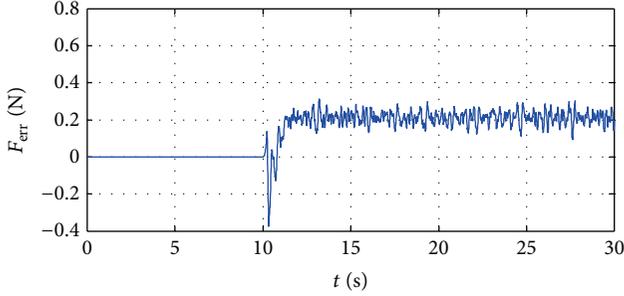
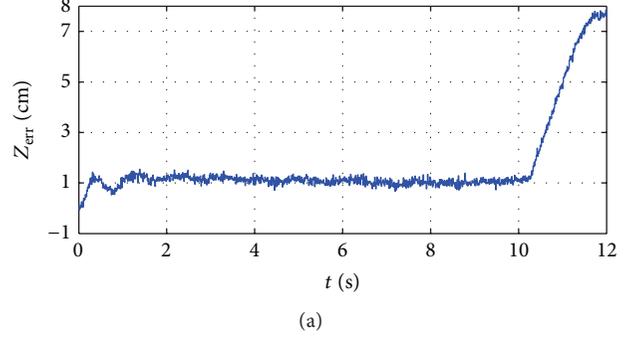
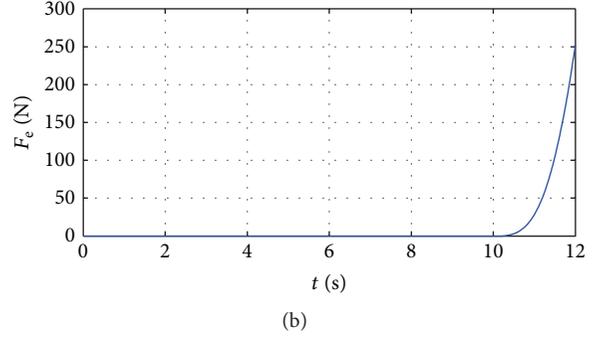


FIGURE 7: Force tracking error along the Z direction.



(a)



(b)

FIGURE 8: (a) Position tracking errors along the z-axis. (b) The interaction force along the z-axis.

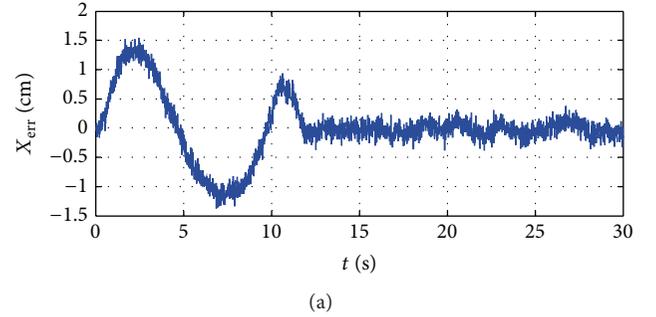
TABLE 3: US-100 testing results for different materials.

Distance (mm)	Wood	Glass	Foam	Plastic	Sponge	Iron
50	55	56	60	60	550	55
100	106	106	105	102	890	106
150	156	157	156	151	260	156
200	205	157	156	151	260	156
300	300	302	299	299	320	301
500	500	504	501	501	520	498

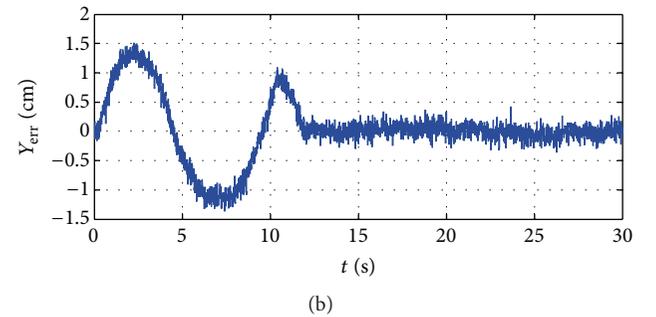
target object surface and some experiments are conducted on different materials to test the performance of the ultrasonic sensor; the experimental results of different materials using US-100 are shown in Table 3.

Although there is a certain deviation between the measured results and the actual distance, but the deviation is within a small range (except that a material has a good ability to absorb ultrasonic wave such as sponge); the cause of the deviation is that an ultrasonic transmitter and receiver sensors are in the interior, while the distance is measured by the front end of the ultrasonic sensor.

In order to get the feedback of force, a dynamometer is well designed and developed by ourselves, which contains a spring and sliding fixture. The contact force between the suction cup and the wall is measured by mapping the relationship between the force and the voltage changed with the displacement. Figure 13 shows the mapping relationship between the voltage value and the force value.



(a)



(b)

FIGURE 9: Position tracking errors: (a) along the x-axis, (b) along the y-axis.

Through Figure 13, we can conclude that the force and the displacement are a linear relationship. When the spring wire diameter is 0.8 cm in diameter and 3.5 cm in length, the mapping formulation is as follows:

$$y = 10.069x + 862.72, \quad (23)$$

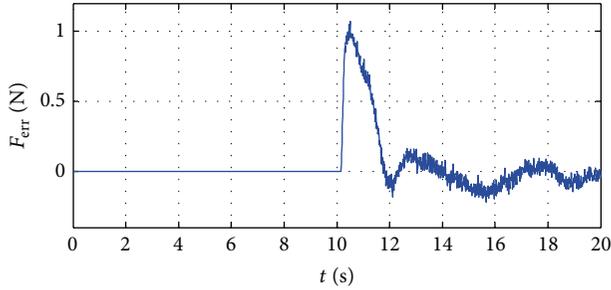


FIGURE 10: Force tracking error.

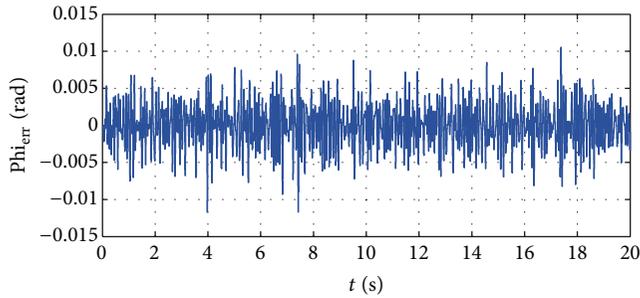


FIGURE 11: The orientation tracking error phi.

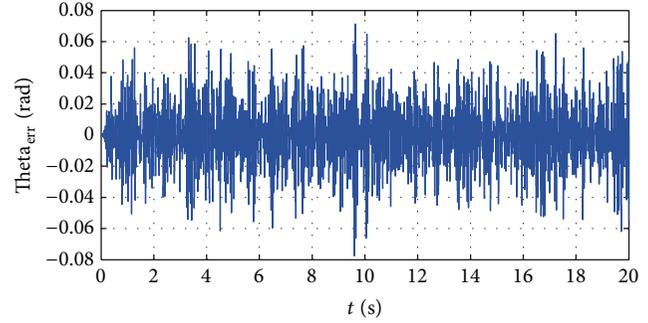


FIGURE 12: The orientation tracking error theta.

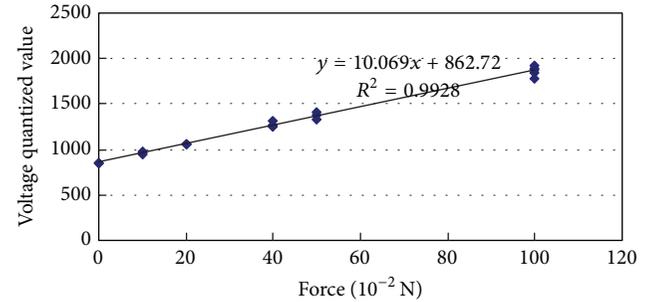


FIGURE 13: The mapping results between voltage and force.

where y is the measured force and x is the measured voltage which is linear with the displacement of the dynamometer.

4.6. Switching Control Testing. To test the switching strategy and also simplify the testing preparation, the structure with a spring and slider fixture, called dynamometer, is developed to link the quadrotor and the suction cup. From the mapping results for the dynamometer, the relationship between the displacement and the force is linear during the contact process. So the impedance control could be yielded as position control. We let the robot take off from the ground and perch on a ceiling which is about 3 meters high. The trajectory is planned in the manual way. The experimental results are shown in Figure 14. The robot can fly in the free space and approach the target position of the ceiling. The system can handle the external force when the robot pushes the ceiling. The rotors stop running after the robot perches on the ceiling; during this stage, the power consumption is very lower because the vacuum pump starts to work instead of the quad rotors. The whole switching process for top perching is very smooth and stable so that it verifies the effectiveness of the method and the developed actively flying and perching robot. When the robot leaves the ceiling, note that the robot needs to start to run the rotors to produce the force to support the gravity of the body before the adhesion force will be released. Currently, the robot needs struggle with the adhesion force because of the residual negative pressure even if the vacuum pump stopped working. A valve which connects the inner cup with outer air will be added to solve the issue. One could also think about a further work here: implement the

control algorithm and plan the trajectory in the autonomous way. But this would require new hardware support and is undergoing.

5. Conclusions and Future Work

The bioinspired active flying and perching robot has many applications and market potentials because it can increase the time of endurance greatly, about ten times and even hundreds of times longer than typical battery-powered flying robots. However, it is difficult to control the switching of the robot we have built in natural way like a flying animal because of the unexpected external force. The impedance force control method is developed to enable the robot to approach a desired position and handle the contact force with the environment smoothly until the adsorption device perches on the object surface steadily. Simulations and preliminary experiments were performed to evaluate the performance of the position and force tracking, which demonstrate the effectiveness of the proposed methods. In the future work, further experiments will be conducted on the developed platform using the proposed method and comparing the performance to realize an autonomous and robust switching between flying and perching. Meanwhile, the micro wireless camera may be mounted on the robot to do some tasks with the use of image processing. Furthermore, other sensors will be integrated to control the system to accomplish more complex missions.

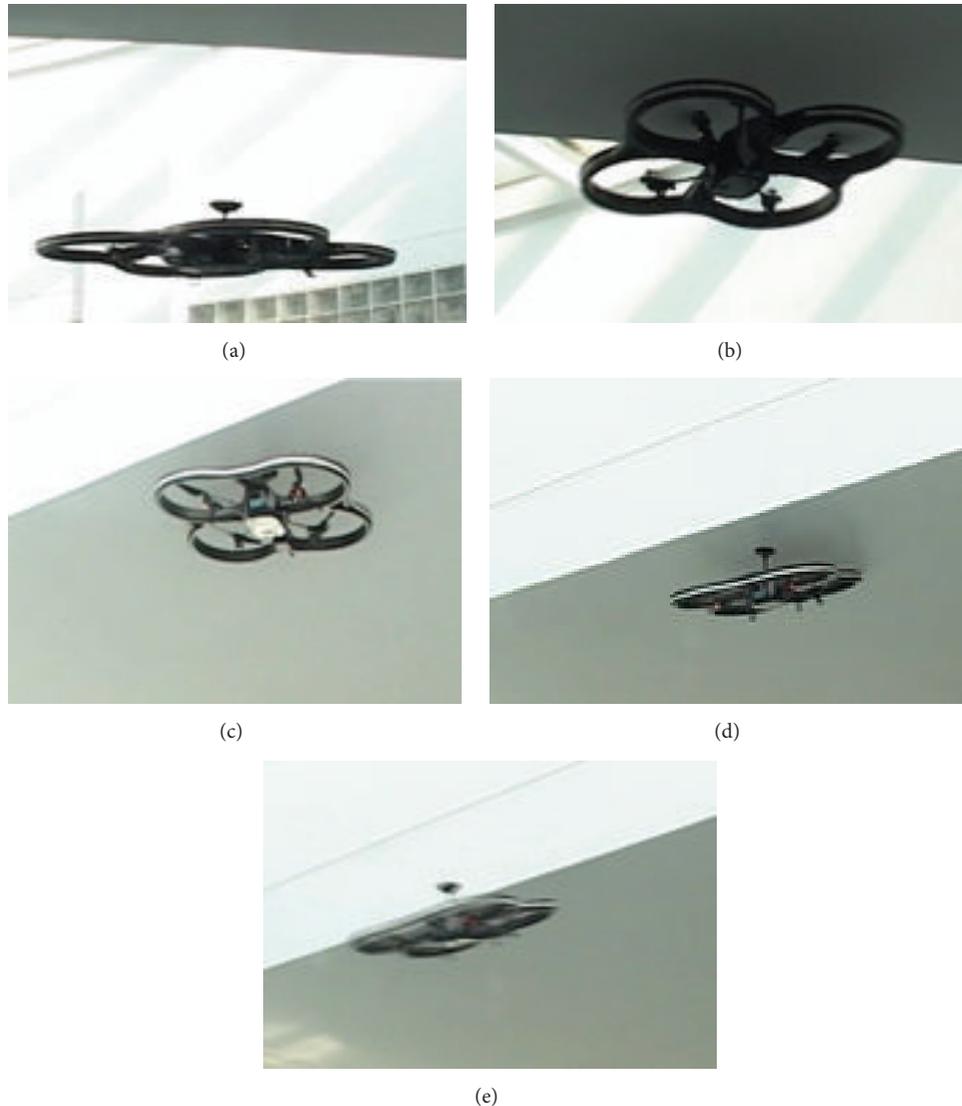


FIGURE 14: A group of sequences of experimental results of the top perching robot adsorbing on a ceiling. (a) Fly in the sky, (b) approach the ceiling and adsorb on the ceiling, (c) stop flying and perch on the ceiling, (d) start to fly from the nest, and (e) fly again in the sky.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants 61473155 and 61175082, Jiangsu Prospective Joint Research Project under Grants BY2013046, and Lianyungang Joint Research Project under Grants CXY1310.

References

- [1] P. E. Rybski, N. P. Papanikolopoulos, S. A. Stoeter et al., "Enlisting rangers and scouts for reconnaissance and surveillance," *IEEE Robotics & Automation Magazine*, vol. 7, no. 4, pp. 14–24, 2000.
- [2] E. Krotkov and J. Blicth, "The defense advanced research projects agency (DARPA) tactical mobile robotics program," *International Journal of Robotics Research*, vol. 18, no. 7, pp. 769–776, 1999.
- [3] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 510–516, 2004.
- [4] A. L. Desbiens and M. R. Cutkosky, "Landing and perching on vertical surfaces with microspines for small unmanned air vehicles," in *Selected Papers from the 2nd International Symposium on UAVs, Reno, Nevada, U.S.A. June 8–10, 2009*, pp. 313–327, Springer, Dordrecht, The Netherlands, 2010.
- [5] M. L. Anderson, C. J. Perry, B. M. Hua et al., "The sticky-pad plane and other innovative concepts for perching UAVs," in *Proceedings of the 47th AIAA Aerospace Sciences Meeting*, Orlando, Fla, USA, January 2009.
- [6] M. Kovač, J. Germann, C. Hürzeler, R. Y. Siegwart, and D. Floreano, "A perching mechanism for micro aerial vehicles,"

- Journal of Micro-Nano Mechatronics*, vol. 5, no. 3-4, pp. 77–91, 2009.
- [7] W. Chi, K. H. Low, K. H. Hoon, J. Tang, and T. H. Go, “A bio-inspired adaptive perching mechanism for unmanned Aerial Vehicles,” *Journal of Robotics and Mechatronics*, vol. 24, no. 4, pp. 642–648, 2012.
- [8] P. G. Backes, Y. Bar-Cohen, and B. Joffe, “The multifunction automated crawling system (MACS),” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 335–340, Albuquerque, NM, USA, April 1997.
- [9] A. Wickenheiser and E. Garcia, “Perching aerodynamics and trajectory optimization,” in *14th International Symposium on Smart Structures and Materials & Nondestructive Evaluation and Health Monitoring*, vol. 6525 of *Proceedings of SPIE*, International Society for Optics and Photonics, April 2007.
- [10] Y. Liu, G. Sun, and H. Chen, “A micro robot with the ability of fly and adhesion: development and experiment,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '11)*, pp. 2413–2414, December 2011.
- [11] Y. Liu, H. Chen, Z. Tang, and G. Sun, “A bat-like switched flying and adhesive robot,” in *Proceedings of the IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER '12)*, pp. 92–97, IEEE, Bangkok, Thailand, May 2012.
- [12] B. Samir, N. Andk, and S. Roland, “PID vs LQ control techniques applied to an weight augmentation high energy conruption indoor micro quadrotor,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2451–2456, 2004.
- [13] T. Madani and A. Benallegue, “Backstepping control for a quadrotor helicopter,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 3255–3260, IEEE, Beijing, China, October 2006.
- [14] A. Mokhtari, A. Benallegue, and B. Daachi, “Robust feedback linearization and $G_{H_{\infty}}$ controller for a quadrotor unmanned aerial vehicle,” in *Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 1009–1014, IEEE, August 2005.
- [15] K. Kiguchi and T. Fukuda, “Position/force control of robot manipulators for geometrically unknown objects using fuzzy neural networks,” *IEEE Transactions on Industrial Electronics*, vol. 47, no. 3, pp. 641–649, 2000.
- [16] S. Jung and T. C. Hsia, “Robust neural force control scheme under uncertainties in robot dynamics and unknown environment,” *IEEE Transactions on Industrial Electronics*, vol. 47, no. 2, pp. 403–412, 2000.
- [17] R. J. Anderson and M. W. Spong, “Hybrid impedance control of robotic manipulators,” *IEEE Journal of Robotics and Automation*, vol. 4, no. 5, pp. 549–556, 1988.
- [18] S. Jung, T. C. Hsia, and R. G. Bonitz, “Force tracking impedance control of robot manipulators under unknown environment,” *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 474–483, 2004.
- [19] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, “Design, modeling, estimation and control for aerial grasping and manipulation,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11)*, pp. 2668–2673, September 2011.
- [20] K. J. Yoon and N. S. Goo, “Development of a small autonomous flying robot with four-rotor system,” in *Proceedings of the IEEE 15th International Conference on Advanced Robotics (ICAR '11)*, pp. 150–154, IEEE, Tallinn, Estonia, June 2011.

Research Article

Development and Evaluation of Compact Robot Imitating a Hermit Crab for Inspecting the Outer Surface of Pipes

Naoto Imajo, Yogo Takada, and Mikiji Kashinoki

Osaka City University, 3-3-138 Sugimoto, Sumiyoshi-ku, Osaka-shi, Osaka 558-8585, Japan

Correspondence should be addressed to Yogo Takada; takada@mech.eng.osaka-cu.ac.jp

Received 14 January 2015; Revised 20 April 2015; Accepted 13 May 2015

Academic Editor: Liwei Shi

Copyright © 2015 Naoto Imajo 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.

Terrestrial hermit crabs which are a type of hermit crabs live on land, whereas typical hermit crabs inhabit the sea. They have an ability of climbing a tree vertically. Their claws allow them to hang on the tree. In this study, an outer-pipe inspection robot was developed. Its locomotion mechanism was developed in imitation of the terrestrial hermit crab's claws. It is equipped with two rimless wheels. Each of the spokes is tipped with a neodymium magnet, which allows the robot to remain attached to even a vertical steel pipe. Moreover, the robot has a mechanism for adjusting the camber angle of the right and left wheels, allowing it to tightly grip pipes with different diameters. Experiments were conducted to check the performance of the robot using steel pipes with different diameters, placed horizontally, vertically, or obliquely. The robot attempted to move a certain distance along a pipe, and its success rate was measured. It was found that the robot could successfully travel along pipes with vertical orientations, although it sometimes fell from oblique or horizontal pipes. The most likely reason for this is identified and discussed. Certain results were obtained in laboratory. Further experiments in actual environment are required.

1. Introduction

There are many aging petrochemical plants, which makes maintenance an important issue. Such plants contain a large number of steel pipes running horizontally, vertically, and obliquely, and inspecting them is both costly and time-consuming. It can also be a difficult process in areas where pipes are densely packed. For these reasons, there is a strong demand for practical pipe inspection robots to aid in this task.

Pipe inspection robots are divided into two types. One type is an inner-pipe inspection robot which moves inside pipe [1–4]. Another type is outer-pipe inspection robot which moves outside pipe. In this study, we focused on the latter because it is expected that the pipe inspection robot would be applied to other structure inspection such as a bridge cable as well as pipe inspection. Several outer-pipe and bridge cable inspection robots have already been developed, and they are capable of moving along pipes and the cables. However, one type is incapable of overcoming obstacles on the surface of the pipe and the cable [5–7]. Another has a specialized mechanism for overcoming obstacles, but the mechanism is too bulky to allow them to be used with narrow pipes [8]. There

is therefore a need for an outer-pipe inspection robot that can move along such narrow pipes and overcome obstacles. If it is not limited to the inspection robot for pipes, permanent magnetic adhesion mechanism is one of the methods for moving on surface of the structure [9, 10].

Terrestrial hermit crabs live on land, whereas typical hermit crabs inhabit the sea. They have an ability of climbing a tree vertically. Their claws allow them to hang on the tree. In imitation of the terrestrial hermit crab's locomotion mechanism, we have previously developed a four-wheel-drive bridge inspection robot with rimless wheels [11]. The terrestrial hermit crabs have some legs. The pair of front legs has large claws. The hermit crabs alternately hang the two claws on tree surface and climb up the tree. The climbing motion of the hermit crabs inspired us to develop the bridge inspection robot having the rimless wheels. The spoke of the rimless wheels was constructed to mimic the hermit crab's claws. Each of the spokes is tipped with a permanent magnet, allowing the robot to move over flat steel surfaces, including walls and ceilings. When the robot climbs up steel wall with rolling rimless wheel, each of the spokes attaches to the wall in

sequence. The sequence of motion is similar to the climbing motion of the hermit crab using claws. The robot is also capable of overcoming obstacles along its path, due to its rimless wheels. However, it cannot be used on pipes because it does not have the ability to remain attached to cylindrical surfaces, except if the diameter is considerably larger than the robot itself. Nevertheless, the rimless wheel concept is considered to be promising for a pipe-climbing robot also.

In the present study, a prototype two-wheeled outer-pipe inspection robot was developed, which can move vertically along narrow pipes or wires. It is equipped with the same rimless wheels imitating the hermit crab's claws. The performance of the robot was evaluated using pipes with different diameters and orientations.

2. General Description of Outer-Pipe Inspection Robot

2.1. Requirements for Pipe-Inspection Robot. The following requirements are considered essential for a robot that can perform pipe inspections.

- (1) The robot should be capable of moving along the outer surface of pipes, regardless of their diameter or orientation.
- (2) It should be able to overcome obstacles on the surface of a pipe and negotiate bends.
- (3) It should be capable of remaining attached to a pipe for extended periods of time in order to perform an inspection. It should therefore be light and use little electric power. It should have a large capacity battery and consume less power when it is motionless on the pipe.
- (4) It should be capable of supporting the weight of sensors for detecting corrosion or cracks.

2.2. Conceptual Design of Outer-Pipe Inspection Robot. In the present study, a compact two-wheel-drive design was chosen, since this was considered to be capable of satisfying the above requirements. As shown in the schematic illustration in Figure 1, the robot uses right and left rimless, spoked wheels. If there are small obstacles on the path, the robot can stride them by spokes. Each of the spokes is tipped with a neodymium permanent magnet, which allows the robot to remain permanently attached to a steel pipe. One common problem with such two-wheeled robots is that their body becomes rotated due to the reaction torque from the motor when the wheels are turned, preventing them from moving straight. To solve this issue, the robot in the present study is equipped with a rear wheel to eliminate pitching motion. Figure 2 schematically shows the robot attached to the outer surface of a pipe. It has a link mechanism for adjusting the camber angle of its wheels so that it can adapt to different pipe diameters. Initial attempts by the robot to move on a pipe were unsuccessful. One of the wheels did move and produced a yawing motion, causing the robot to fall from the pipe. To

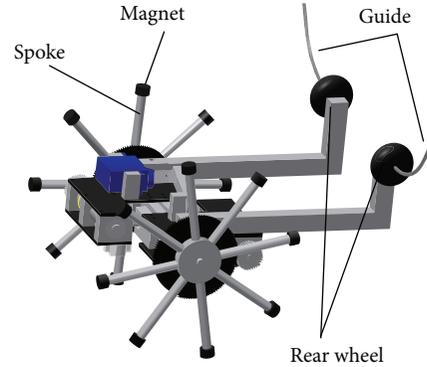


FIGURE 1: Conceptual design of outer-pipe inspection robot.

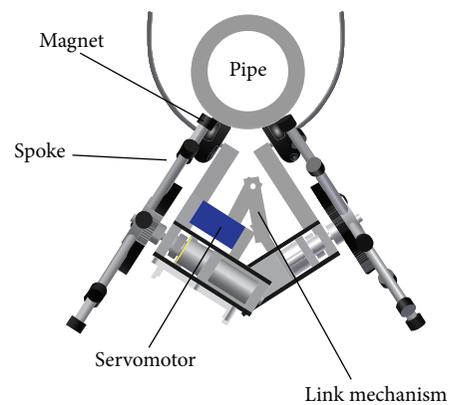


FIGURE 2: Robot attached to pipe.

overcome the problem of yawing, the robot is equipped with guides to hold the pipe as shown in Figure 1.

2.3. Magnetic Force between Robot and Steel Pipe. As illustrated in Figure 3, each of the rimless wheels has eight spokes, on the end of which is a permanent magnet with a length of 5 mm and a diameter of 8 mm. In order to ensure that the robot can remain attached to a pipe regardless of the orientation of either the robot or the pipe, the attractive magnetic force between the two must exceed the gravitational force acting on the robot.

Figure 4 illustrates the magnetic force F_m acting between one of the wheel spokes and a flat steel surface. It varies as a function of the distance L between the permanent magnet and the steel surface. Figure 5 shows the dependence of F_m on L , calculated using the computation tool for magnetic-flux density and absorption force from NeoMag Co., Ltd.

Since it is difficult to perform a theoretical analysis of the magnetic force when L is too small, the F_m value of 14.7 N for $L = 0.4$ mm is used for all distances $L < 0.4$ mm. When the spoke is not vertical but instead makes an angle α to the vertical axis, the distance between the magnet and the steel surface is not uniform, as shown in Figure 6, so that the magnetic force varies with α . One of the authors derives the following

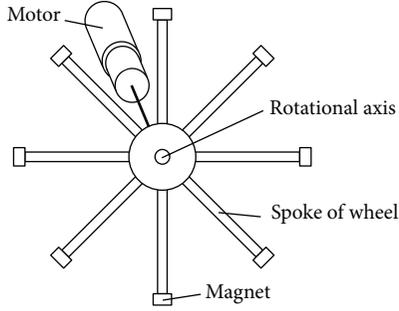


FIGURE 3: Schematic of rimless wheel and magnets.

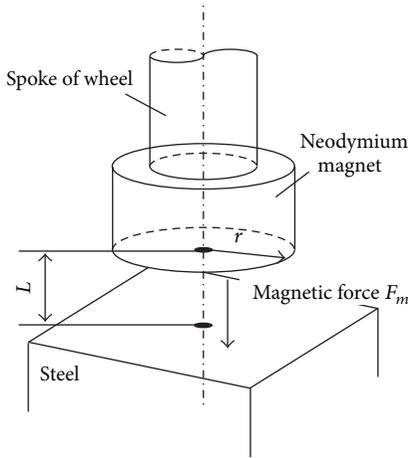


FIGURE 4: Magnetic force between spoke and steel surface.

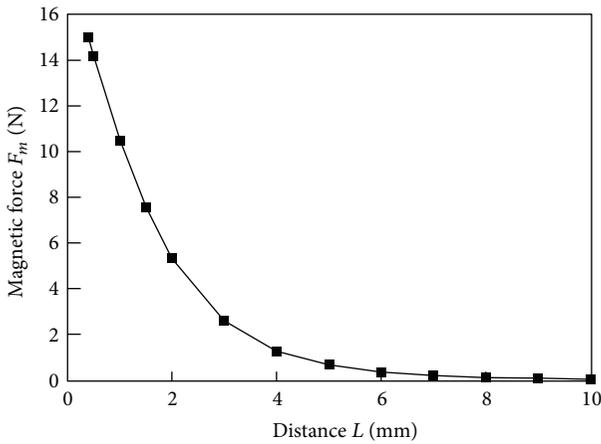


FIGURE 5: Dependence of magnetic force on distance.

equation of the magnetic force [11]. The magnetic force is given by

$$F_m = \int_0^{2r} \frac{2\sqrt{2rx - x^2}}{\pi r^2 (bx - L)^2} dx \quad (1)$$

$$b = \sin \alpha,$$

where r is the radius of the magnet. Figure 7 shows the relationship between F_m and α . The wheel has eight spokes.

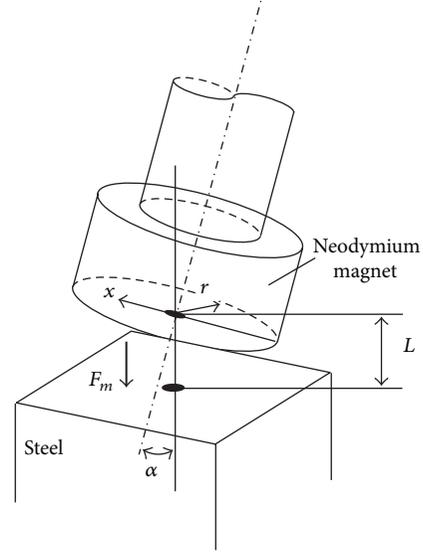


FIGURE 6: Magnetic force for tilted spoke.

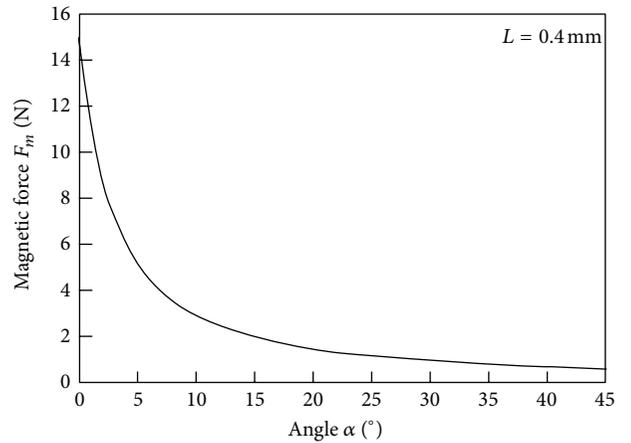


FIGURE 7: Relationship between magnetic force and tilt angle.

Therefore, when the wheel is rolling on a steel pipe, its spoke is detached from the pipe at $\alpha = 22.5^\circ$. When $\alpha = 22.5^\circ$, $F_m \cong 1.2$ N, which is about 8% of the value for $\alpha = 0^\circ$. Therefore, the magnet can be detached from the steel pipe by a much smaller force. Based on these results, a prototype robot was constructed, as described in the following section.

3. Construction of Prototype Robot

Figure 8 shows a schematic diagram of the prototype robot. It weighs 380 g and is 180 mm in length, 124 mm in width, and 110 mm in height. In order to minimize the weight of the robot, carbon fiber-reinforced plastic and duralumin are used to construct the frame. The permanent magnets are coated with rubber cement to prevent slipping on the steel pipe. The robot is driven using DC reduction motors equipped with a gearbox with planetary gears. These motors are light (15 g) and can produce a high torque. To evaluate its performance, it

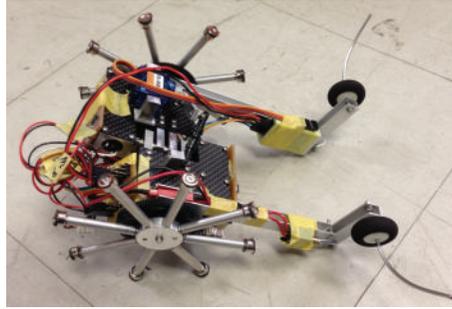


FIGURE 8: Prototype robot.

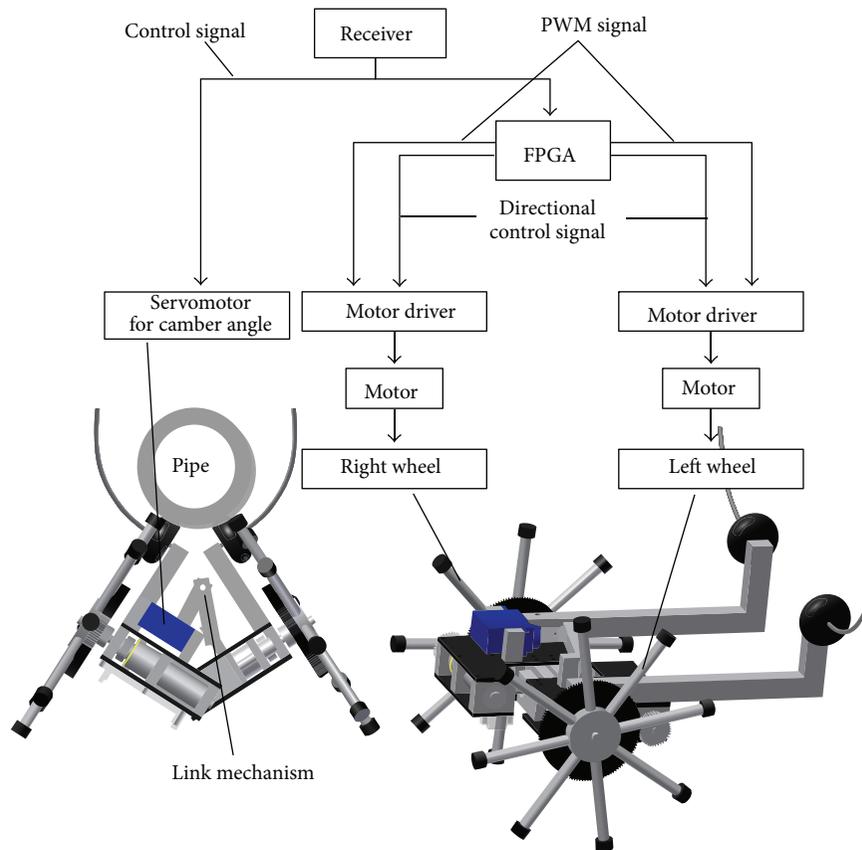


FIGURE 9: Signal flow for robot control.

was gradually subjected to a load torque for an applied voltage of 1.5 or 3.0 V. When the motor stopped rotating, the measured torque was 0.0908 Nm at 1.5 V and 0.181 Nm at 3.0 V [12]. Thus, the motor torque is roughly proportional to the input voltage. For an applied voltage of 7.4 V, the motor torque can therefore be calculated to be 0.446 Nm. Moreover, two reducing gears were attached to the robot, with 37 and 116 gear teeth, giving a gear ratio of 3.14. If the transmission efficiency of the reducing gears is 0.9, the torque of the robot is then $0.446 \times 3.14 \times 0.90 = 1.26$ Nm. Figure 9 shows the signal flow used to control the robot. The signals are transmitted (T6J, Futaba Co., Ltd.) wirelessly to a receiver (R2106GF, Futaba Co., Ltd.) on the robot. A field-programmable gate array

(FPGA; XC6SLX9-2TQG144C, Xilinx Co., Ltd.) is used to process the PWM signals from the receiver and issues signals including directional controls to the motor drivers, which cause the motors to rotate. The robot is also equipped with a servomotor for adjusting the camber angle of the wheels. The guides are made of curved lengths of wire. Two small wheels are attached to the rear of the robot for repressing pitching motion. Those wheels do not deface the surface of the pipe.

4. Performance of Prototype Robot

Since an actual outer-pipe inspection robot will be expected to carry sensors and other devices, the maximum movable

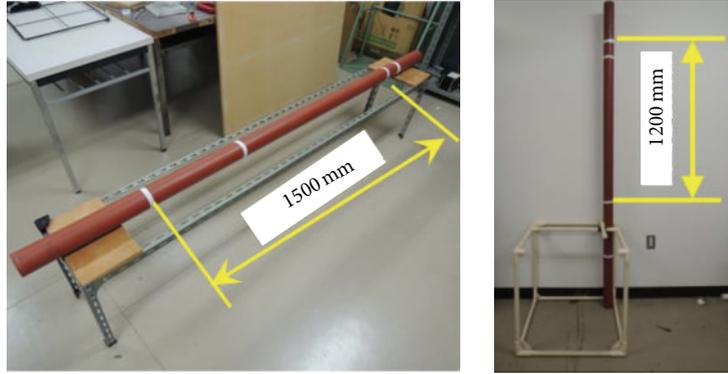


FIGURE 10: Experimental setup for testing robot movement.

TABLE 1: Movable load during vertical movement.

Maximum movable load (without slip)	155 g
Maximum movable load (with slip)	245 g

TABLE 2: Electric power consumption.

Climbing with 0 g load	4.56 W
Climbing with 155 g load	6.59 W

load during vertical movement on the wall was first determined, and the results are shown in Table 1. The robot will be expected to move vertically on the wall without slip. The movable loads without slip and with slip are measured. Table 2 shows the electric power consumption of the robot for different loads.

To evaluate the performance of the robot, experiments were carried out using steel pipes with different diameters. Several experiments were conducted with the outer-pipe inspection robot. We prepared SGP pipes with various diameters. The SGP is a carbon-steel pipe used for transportation of gas, oil, and other materials in Japan. The nominal external diameters of the steel pipes were 15 A (21.7 mm), 50 A (60.5 mm), 65 A (76.3 mm), and 90 A (101.6 mm). The pipes were coated with corrosion-resistant paint. Figure 10 shows the experimental setup used for testing the ability of the robot to travel a fixed distance. This shows the cases only for horizontal and vertical movement, but the full range of tilt angles was 0° (horizontal), 30° , 45° , 60° , and 90° (vertical). For the horizontal case, the required travelling distance for the robot was 1500 mm, and for all other angles it was 1200 mm. Experiments were conducted twenty times by each route. Figure 11 shows the relationship between the movement speed and the inclination angle. Figures 12 and 13 show example photographs of the robot undergoing vertical and horizontal movement, respectively.

Table 3 shows the success rates for the different inclination angles and pipe diameters. As can be seen, the robot failed to complete the entire distance for the case of the pipe with a diameter of 21.7 mm, regardless of the inclination angle. This was because for this narrow pipe, both of the rimless wheels could not make contact with the pipe at the same time.

TABLE 3: Success rate for movement on the pipe.

Tilt angle of steel pipes	Success rate			
	External diameter			
	21.7 mm	60.5 mm	76.3 mm	101.6 mm
0°		65%	95%	90%
30°	Cannot run the whole distance	95%	100%	100%
45°		100%	85%	95%
60°		100%	100%	95%
90°		100%	100%	100%

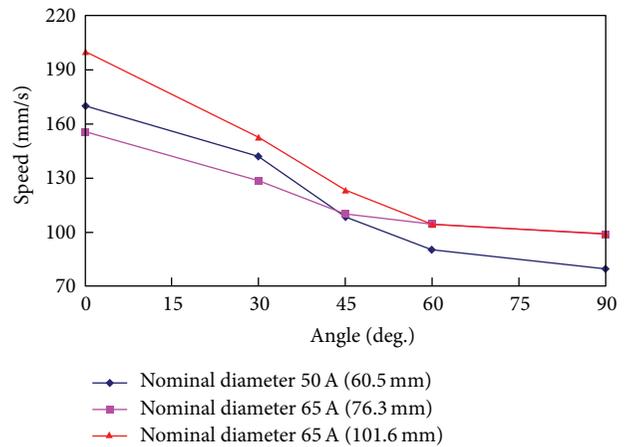


FIGURE 11: Relationship between movement speed and inclination angle of steel pipe.

Although further investigation is needed, it is likely that if the spokes are made longer, the robot would be able to travel even on such a narrow pipe. For other pipe diameters, the success rate was above 85%, except for the case of the 60.5 mm diameter horizontal pipe, where the success rate was 65%. Normally, it would be thought that it would be more difficult for the robot to travel at an inclined angle rather than horizontally, but this was not the case in the present study. Therefore, we carefully observed the behavior of the robot when travelling horizontally on the 60.5 mm diameter pipe.

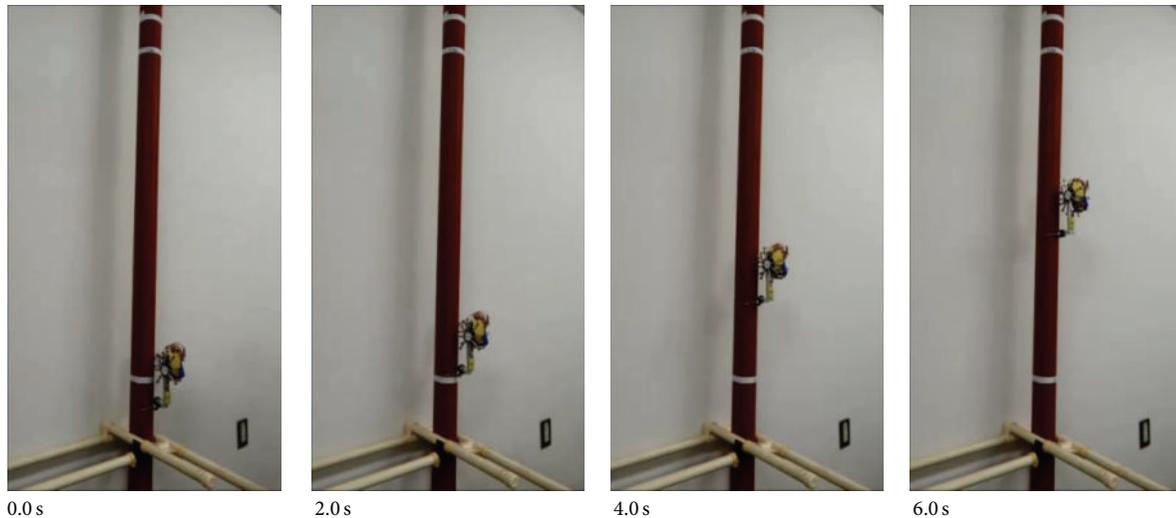


FIGURE 12: Robot moving vertically along a steel pipe.

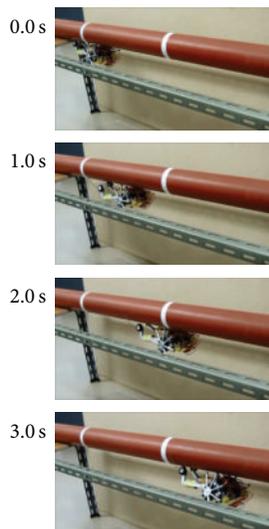


FIGURE 13: Robot moving horizontally along a steel pipe.

Figure 14 shows schematic diagrams of the attitude of the robot when moving horizontally and obliquely on the pipe. It can be seen that when the pipe is inclined, both the front and back wheels are in contact with the pipe, which makes it stable and unlikely to fall off. This is not the case when the pipe is horizontal, because the robot tilts about its front wheels, causing its rear wheels to lose contact with the pipe. The situation might be avoided by redesigning the robot so that its center of gravity is in front of its rimless wheels.

The robot occasionally fell from not only the horizontal pipe but also the inclined pipe. When the robot runs on the inclined pipe, it keeps its stability because the rear wheels touch the pipe. The robot fell from the pipe all of a sudden. This result suggests that the magnetic force of the robot is not enough to retain the robot to the steel pipe. To verify the hypothesis, we used a simulation model of the rimless wheel

for the dynamic analysis of magnetic force. Figure 15 shows relationship between rotation angle and the magnetic force in the whole of a rimless wheel. The magnetic force in the entire wheel changes periodically whenever rotating by 45° because the rimless wheel has eight magnets. Dynamic analysis result showed that the minimum attaching force of the rimless wheel is 1.69 N. The weight of pipe inspection robot is 380 g. When both of the right and left rimless wheels produce minimum attaching force, the attaching force is only 3.38 N. Therefore, the robot would fall from the steel pipe. The cause of the robot falling depends on magnetic force shortage of the magnets. The permanent magnets should be exchanged for ones which can produce stronger attaching force. However, magnetic force must be as low as the motor can rotate. Further work is required to select magnets which produce the suitable force.

5. Conclusions

A prototype robot was developed for inspecting the outer surface of steel pipes. The robot is equipped with rimless wheels, to the spokes of which neodymium magnets are attached. It also has a link mechanism that allows the camber angle of its right and left wheels to be adjusted. Guides are attached to the rear of the robot so that it does not undergo yawing motion on the steel pipe. Experimental tests showed that it was capable of travelling along pipes with diameters of 60.5 to 101.6 mm, although it fell from a pipe with a diameter of 21.7 mm. However, it is thought to be likely that if the spokes are made longer, the robot would also be capable of travelling on such a narrow pipe. The magnetic force of the robot is not enough to retain the robot to the steel pipe. The permanent magnets should be exchanged for ones which can produce suitable attaching force. The movable load in the absence of slippage was 155 g, which would allow it to carry a small camera to inspect pipes. Certain results were obtained in laboratory. Further experiments in actual environment are required.

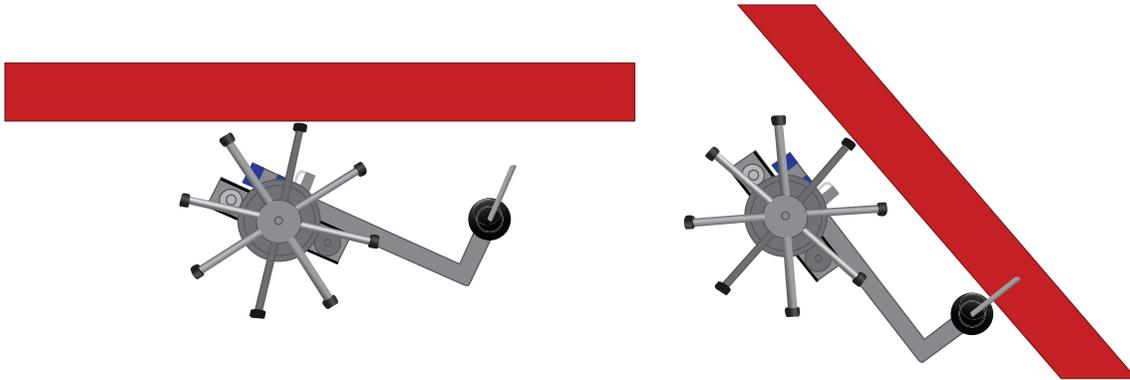


FIGURE 14: Attitude of robot on steel pipe.

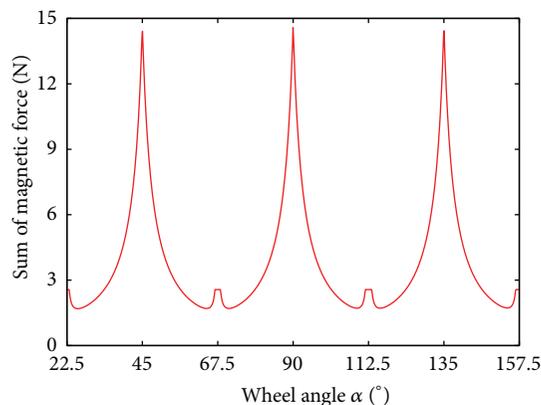


FIGURE 15: Relationship between rotation angle and the magnetic force of the whole of a rimless wheel.

If there are small obstacles on the pipe, the pipe inspection robot can stride them by rimless wheels. On the other hand, the robot cannot stride the large ones. We should stop the robot from crashing the obstacle. However, rear guides of the robot prevent it from changing the moving direction on the pipe. We will redesign the robot for practical use to change the moving direction.

Conflict of Interests

The authors declare that there is no conflict of interests.

References

- [1] O. Tătar, D. Mândru, and I. Ardelean, "Development of mobile minirobots for in pipe inspection tasks," *Mechanika*, vol. 68, no. 6, pp. 60–64, 2007.
- [2] D. Wu, H. Ogai, Y. Yeh, K. Hirai, T. Abe, and G. Sato, "Pipe inspection robot using a wireless communication system," *Artificial Life and Robotics*, vol. 14, no. 2, pp. 154–159, 2009.
- [3] N. A. B. H. Yahya, N. Ashrafi, and A. H. Humod, "Development and adaptability of in-pipe inspection robots," *IOSR Journal of Mechanical and Civil Engineering*, vol. 11, no. 4, pp. 1–8, 2014.
- [4] M. Horodincu, I. Doroftei, E. Mignon, and A. Preumont, "A simple architecture for in-pipe inspection robots," in *Proceedings of the International Colloquium on Autonomous and Mobile Systems*, Magdeburg, Germany, 2002.
- [5] M. Hamada, M. Sasaki, K. Koizumi, T. Sasaki, and K. Kinoshita, "Development of out-pipe robot," in *Proceedings of the 43rd JSME Hokuriku-Shinetsu Branch Annual Meeting*, 2006.
- [6] H.-B. Yun, S.-H. Kim, L. Wu, and J.-J. Lee, "Development of inspection robots for bridge cables," *The Scientific World Journal*, vol. 2013, Article ID 967508, 17 pages, 2013.
- [7] C. Choi, S. Jung, and S. Kim, "Feeder pipe inspection robot with an inch-worm mechanism using pneumatic actuators," *International Journal of Control Automation and Systems*, vol. 4, no. 1, pp. 87–95, 2006.
- [8] T. Yukawa and H. Okano, "Development of a mechanism of pipe-surface inspection robot," *Journal of the Japan Society for Precision Engineering*, vol. 74, no. 6, 2008.
- [9] M. F. Silva, R. S. Barbosa, and A. L. Oliveira, "Climbing robot for ferromagnetic surfaces with dynamic adjustment of the adhesion system," *Journal of Robotics*, vol. 2012, Article ID 906545, 16 pages, 2012.
- [10] W. Shen, J. Gu, and Y. Shen, "Permanent magnetic system design for the wall-climbing robot," *Applied Bionics and Biomechanics*, vol. 3, no. 3, pp. 151–159, 2006.
- [11] M. Kashinoki, Y. Takada, T. Kawai, and K. Koyama, "Running ability evaluation of bridge inspection robot with permanent magnets," in *Proceedings of the 12th Symposium on Evaluation and Diagnosis*, CD-ROM, 2013.
- [12] Y. Takada, K. Kirimoto, T. Tajiri, and T. Kawai, "Development of a bridge inspection robot working in three-dimensional environment," *Transactions of the Japan Society of Mechanical Engineers*, vol. 79, no. 805, pp. 3135–3146, 2013.

Research Article

Sparse Approximation for Nonrigid Structure from Motion

Yaming Wang, Xiaomeng Yan, Junbao Zheng, and Mingfeng Jiang

School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China

Correspondence should be addressed to Xiaomeng Yan; 1171803460@qq.com

Received 5 December 2014; Revised 19 March 2015; Accepted 1 April 2015

Academic Editor: Nan Xiao

Copyright © 2015 Yaming 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.

This paper introduces applying a novel sparse approximation method into solving nonrigid structure from motion problem in trajectory space. Instead of generating a truncated traditional trajectory basis, this method uses an atom dictionary which includes a set of overcomplete bases to estimate the real shape of the deformable object. Yet, it still runs reliably and can get an optimal result. On the other hand, it does not need to consider the size of predefined trajectory bases; that is to say, there is no need to truncate the trajectory basis. The mentioned method is very easy to implement and the only trouble which needs to be solved is an L_1 -regularized least squares problem. This paper not only presents a new thought, but also gives out a simple but effective solution for the nonrigid structure from motion problem.

1. Introduction

Nonrigid structure from motion (NRSfM) refers to the process of recovering 3D coordinates of a nonrigid object from its 2D projections. This technology plays an important role in computer vision applications [1–4]. One prevalent solution for solving this process was considering the deformable object as a linear combination of shape basis. This method worked well for simple motion containing a single action. However, it would be hard to recover the structure of the object when dealing with complex sequences. For this reason, Akhter et al. [5] introduced a common method that reconstructs the 3D object in trajectory space instead of shape space. This approach did well with long complex motion sequences because these measurement points were not dependent on each other. What is more, since the trajectories of the points of most 3D motion in the real world were some curved lines which were naturally smooth, these trajectories could be modeled with a linear combination of some known curved lines, known as trajectory basis. The appearance of trajectory basis model improved the solutions for recovering nonrigid shape from motion well.

Moreover, experiments also proved that the efficiency of the trajectory basis method relied on two factors: the type of the trajectory basis and the number of bases. And in the selection of trajectory basis, it turned out that the discrete

cosine transform (DCT) for Markov was more suitable to be defined as a general basis [6, 7]. However, we should know that, though DCT basis has been proved to be better than the others on the whole, the former could not be suitable for every motion sequence. Another thing, the number of the selected trajectory bases was thought-provoking. Selecting a smaller trajectory basis size might lead to big ignorance of much important information of the motion sequences, while a larger basis size might lead to a large number of unknown factors and the system of equations would be ill-posed. And the former situation might get a bad result, and the latter would be a large waste of time or even could not get the solution of the equations [8].

(1) *Contributions.* In general, a solution to 3D point trajectories could be obtained more accurately by reducing the restrictions of the trajectory basis. This paper introduces applying sparse coding algorithms to the NRSfM problem. In this process, a set of overcomplete bases called atonic dictionary was predefined to represent the deformable object with sparse coefficients.

An advantage of the sparse coding approach is that it is not restricted to only one trajectory basis function and may be generated by two or more incoherent basis functions. It is very useful to recover the trajectory curves which consist of some different types of bases functions. What is more, since

the goal of sparse approximation is to represent trajectory sequences as a sparse combination of all atoms, there is no need to predefine the number of trajectory bases.

(2) *Related Work.* Bregler et al. first proposed to use factorization approaches to recover the nonrigid deformable objects from motion [9]. The main thought of Bregler et al. was to obtain a low-rank shape basis which can satisfy the 2D point projections. They argued that the structure of motion object could be regarded as an approximate linear combination of basis shapes. This shape model was widely used in this field, though it appeared to be very hard because the inherent basis ambiguity of the nonrigid problem must be overcome. To resolve this ambiguity, Torresani et al. [10] introduced a Gaussian prior to constraining the solution by reducing the coefficients. For the same reason, Xiao et al. [11] proposed to add extra “basis constraints” besides the orthonormality constraints. Later, Dai et al. [12] proposed a simple prior-free method to solve the NRSfM problem for the first time in 2010. Although they did not assume any additional prior knowledge except the low-rank constraint, the method recovered the nonrigid shape reliably. The shape based model became more mature. The methods mentioned above are based on shape basis, until Akhter et al. [6] made a great improvement, who introduced trajectories based model for solving NRSfM problem instead of shape basis. This approach, regarding the 3D point trajectories as a model in the domain of the discrete cosine transform (DCT) basis vectors, provided better results on complex shapes. Zhu and Lucey proposed to use a penalty to minimize the size of active trajectory basis [13].

2. Problem Formulations

The measured projective trajectories are contained in a $2F \times P$ matrix W as follows:

$$W = \begin{pmatrix} X_{11} & \cdots & X_{1P} \\ \vdots & \ddots & \vdots \\ X_{F1} & \cdots & X_{FP} \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \in R^{2F \times P}. \quad (1)$$

W is the measurement matrix generated from the 2D coordinates (X, Y) of feature points, where F indicates the number of motion frames and P is the number of feature points of the deformable object [9]. Then deal with W and let W be a registered measurement matrix connecting the camera center and the projection on the image plane, so that in the trajectory space W can be represented as

$$W = MS, \quad (2)$$

$$M = \begin{pmatrix} M_1 & & \\ & \ddots & \\ & & M_F \end{pmatrix}, \quad S = \begin{pmatrix} S_1 \\ \vdots \\ S_F \end{pmatrix}.$$

In this formula, M is the camera motion (projection) matrix and S indicates the nonrigid shape matrix [10]. Moreover, the shape coordinates of the object can be decomposed into a set of truncation bases and corresponding coefficients:

$$S_{3F \times P} = \Theta_{3F \times 3k} A_{3k \times P}. \quad (3)$$

Matrix Θ is the predefined trajectory bases, and A indicates the corresponding coefficients.

We all know that most of the energy in natural signal concentrates in low-frequency area and discrete cosine transform (DCT) has a huge advantage in the specificity of energy concentration [11]. For the above reason, DCT is used in NRSfM processing so that the loss compression of data can be conducted. What is more, it has been proved that in general DCT basis is a better basis in the whole. So this paper mainly considers the situation in the DCT trajectory space. The discrete cosine transform is generally formulated as follows:

$$y(k, d) = u(k) \sum_{n=1}^N u(n, d) \cos \frac{\pi(2N-1)(k-1)}{2N}, \quad (4)$$

where

$$k = 1, \dots, N; \quad u(k) = \begin{cases} \frac{1}{\sqrt{M}}; & k = 1 \\ \sqrt{\frac{2}{M}}; & 2 \leq k \leq M. \end{cases} \quad (5)$$

In Akhter’s method [6], the rank of matrix Θ was truncated to k . That is to say, they selected $3k$ columns from a $3F \times 3F$ matrix and k was far smaller than F . But because of the limitation of bases number in use, it might be widely inaccurate when the scenes were complex.

3. Sparse Approximation Method

Traditional trajectory bases approaches of reconstruction mainly involve the use of a certain kind of orthogonal bases, such as the Fourier basis, various DCT bases, and other orthogonal wavelet bases. In these situations, one wishes to represent all the trajectory curves of feature points as a linear combination of a certain waveform. But there is a problem that if the number of trajectory bases was too big, the equations system would be a NP-hard problem. So this paper introduces a new method which uses sparse approximation method to represent the trajectory curves instead of traditional trajectory bases method. And the sparse approximation provides a class of algorithms that learn basis functions only when they capture higher-level features in the input data [14]. Moreover, an overcomplete atom dictionary will be used in this method other than trajectory bases. An overcomplete atom dictionary which contains different kinds of bases functions will help to get a better result.

3.1. Atom Dictionary. In reality, wavelets perform poorly on high-frequency sinusoids and, on the contrary, sinusoids perform poorly on impulsive events [15]. So it would not be

accurate to recover the trajectory curves just with one certain basis. Dirac function and trigonometric function are two kinds of fundamental orthonormal bases, and most of the real trajectory curves can be represented by these two functions. If Dirac basis and DCT basis are put together in one matrix, this will generate an atom dictionary which performs better than any independent trajectory basis. And the coefficients of the dictionary must be sparse (i.e., the coefficients include many zero items). Each column of this dictionary is called atom, and the concatenation of Dirac basis and DCT basis has turned out to be suitable for most situations [16]. In the paper this dictionary is used in experiments with experimental data instead of trajectory bases.

3.2. Sparse Coding. A dictionary is a concatenation of some orthonormal bases. So the expression is not unique, and many combination methods arise. But the goals of these methods are all to get a highly sparse decomposition which contains very few nonzero terms. This puts forward an optimization problem:

$$\begin{aligned} \alpha' &= \operatorname{argmin}_{\alpha} \|\alpha\|_0 \\ \text{s.t. } W &= \Pi\alpha, \end{aligned} \quad (6)$$

where $\Pi = M\Theta$ and the symbol $\|\cdot\|_0$ indicates the ℓ_0 norm which constrains the nonsparse items. It has been turned out to be right that if the isometry constant Δ satisfied the following equations:

$$\begin{aligned} (1 - \Delta) \|\alpha\|_2^2 &\leq \|\Pi\alpha\|_2^2 \leq (1 + \Delta) \|\alpha\|_2^2 \\ \Delta &< 1, \end{aligned} \quad (7)$$

then a unique K -sparse solution of the vector α will be found. But a drawback to (4) is that it may be NP-hard. Fortunately, one can get a convex relaxation on (4) with a ℓ_1 norm as follows:

$$\begin{aligned} \alpha' &= \operatorname{argmin}_{\alpha} \|\alpha\|_1 \\ \text{s.t. } W &= \Pi\alpha. \end{aligned} \quad (8)$$

The ℓ_1 norm is a very common solution to solve sparse estimation problems and it has been turned out to be effective. A variety of different solution packages can be found to solve the ℓ_1 norm problem. This paper will apply ℓ_1 -regularized least squares, namely, the feature-sign search algorithm, to achieve the sparse representation of the object motion trajectory. This algorithm is mentioned in [13, 17, 18].

4. Algorithm Solution

Having mastered the above theories, applications of sparse approximation algorithm in the NRSfM problem will be presented in the following words.

The goal of the NRSfM is to estimate the camera motion matrix M and then recover the true nonrigid 3D coordinate matrix S from the measurement matrix W . For this purpose,

this paper needs to estimate the matrix M at first. Fortunately, there has been a well solution for this objective. Dai et al. [12] have put forward a prior-free method for shape basis model. And this method could also be used in trajectory space. Having solved the motion matrix M , then an overcomplete atom dictionary can be predefined as the trajectory basis. The sparse approximation algorithm will help to solve the corresponding coefficients. So the 3D shape matrix would be obtained with the trajectory basis and the corresponding coefficients.

4.1. Estimate Camera Motion Matrix. One can compute the rank-3K decomposition of a measurement matrix via singular value decomposition (SVD) [18] and obtain the equation as $W = \widetilde{M}\widetilde{A}$. For the reason that there exist any rank-3K matrices G which satisfied the equation that $W = \widetilde{M}\widetilde{A} = \widetilde{M}GG^{-1}\widetilde{A} = MA$, so the decomposition is not unique. In this paper, a semidefinite programming (SDP) of small and fixed size mentioned in [15] is applied to solve the above problem. Dai's method is also effective in estimating G in trajectory space by trace minimization. The linear equations system can be obtained as follows.

$\min \operatorname{trace}(Q_k)$, such that

$$\begin{aligned} Q_k &= G_k G_k^T, \\ Q_k &\geq 0, \end{aligned} \quad (9)$$

$$\begin{bmatrix} (\widetilde{M}_i \otimes \widetilde{M}_i)(1, :) - (\widetilde{M}_i \otimes \widetilde{M}_i)(4, :) \\ (\widetilde{M}_i \otimes \widetilde{M}_i)(2, :) \end{bmatrix} \operatorname{vec}(Q_k) = 0$$

where G_k denotes the k th column-triplet of G , \widetilde{M}_i denotes the i th double rows of \widetilde{M} , $(\widetilde{M}_i \otimes \widetilde{M}_i)(j, :)$ denotes the j th row of $(\widetilde{M}_i \otimes \widetilde{M}_i)$, and $\operatorname{vec}(\cdot)$ denotes the vectorization operator. The above linear system will get a unique solution via a standard SDP. Then G_k can be found by using SVD. Once the G is obtained, one is allowed to compute the camera motion matrix M by the following equation:

$$\widetilde{M}_i G_k = \theta_{ik} M_i, \quad i = 1, \dots, F. \quad (10)$$

Note that θ_{ik} is included in the $\Theta_{3F \times 3k}$. Finally, the motion matrix M is represented as $M = \operatorname{blkdiag}([M_1, M_2, \dots, M_F])$.

4.2. Estimate the Coefficients Matrix A. The coefficients matrix A is a sparse matrix; that is to say, matrix A consists of many nonzero items. And from the previous statement, A can be solved by a sparse approximation method. Having known the camera motion matrix M , Π can be obtained from the equation $\Pi = M\Theta$. Then an alternate strategy can be described as follows:

$$\alpha'_i = \operatorname{argmin}_{\alpha_i} \|\alpha_i\|_1 \quad (11)$$

$$\text{s.t. } w_i = \Pi\alpha_i,$$

where $W = [w_1, w_2, \dots, w_i], i = 1, \dots, P$.

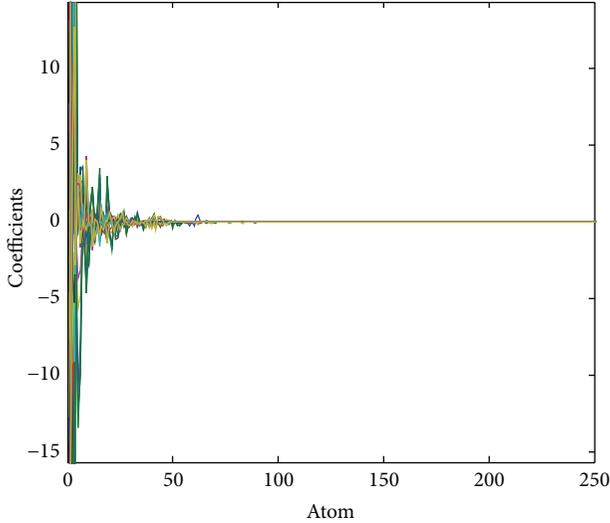


FIGURE 1: Coefficients distribution from the 1st to the 250th atoms on the “Yoga” sequence (better viewed in color).

To simplify the expression, it can be represented by the following optimization problem:

$$\min_x f(x) \equiv \|w_i - \Pi\alpha_i\|^2 + \gamma \|\alpha_i\|_1, \quad (12)$$

where γ is the penalty coefficient and it must be a constant.

An iterative feature-sign search algorithm, which can solve for α in the Fourier domain, can be used to solve this objective efficiently. The details of the algorithm have been introduced in [17]. The coefficients matrix A can be generated from the vector α_i . At last, the shape matrix S will be obtained via the equation $S = \Theta A$.

5. Experiments

The training dataset used in this paper is from CMU Motion Capture dataset, which covers a variety of human actions. The random synthetic data are used only in the condition of algorithm validation and the sparse approximation method performs clearly better than others. So the result will be not mentioned and this paper reports the results on real sequences only. The real scenes tested in this paper mainly include the commonly sequences of “Yoga” (41/307), “drinking” (41/1102), “pickup” (41/357), “shark” (91/240), “Stretch” (41/740), and “walking” (55/260), where (F/P) denotes the number of points (P) and frames (F).

5.1. Sparse Coefficients Estimations. At first, this paper does one experiment on “Yoga” sequence. A union atom dictionary of DCT and Dirac function is used in this experiment. Having recovered the 3D coordinates of the deformable object, the corresponding coefficients of the atom dictionary are obtained naturally. And the experimental result is shown in Figure 1. Figure 1 shows the corresponding coefficients range from the 1st to the 250th atoms.

Coordinate “atom” refers to the basis order of the dictionary generated by trajectory basis functions. That is to

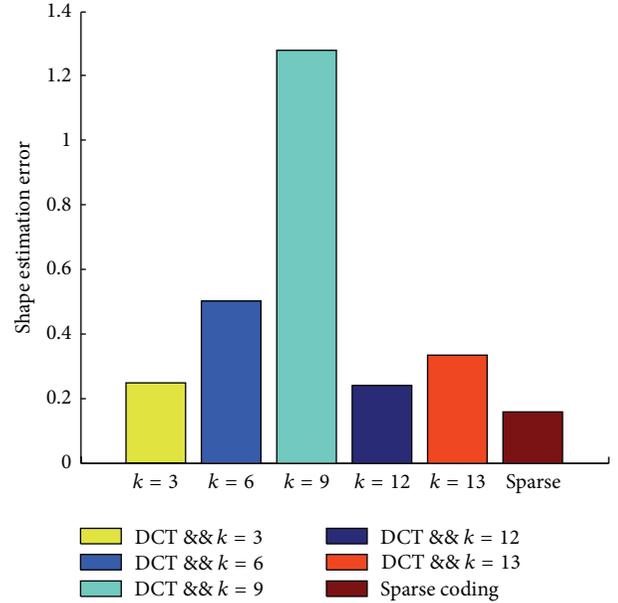


FIGURE 2: Shape estimation error on the “Yoga” motion sequence. From left to right, each column indicates different performances when the sizes of DCT basis are 3, 6, 9, 12, and 13, respectively, and the most right column gives the result of sparse methods (better viewed in color).

say, the “atom” corresponds to the columns of one dictionary matrix. Coordinate “coefficients” refer to the corresponding coefficients of each atom. From the above graph, one can easily find that most of the atom coefficients are zeros. It proves that the assumption is suitable that the coefficients matrix is a sparse matrix. At the same time, the result also verifies the feasibility of the sparse approximation method.

5.2. Experimental Result on One Real Sequence. It has been proved that DCT was better than the others in the application of trajectory basis model and it is the most common method. This paper compares the sparse method against the trajectory basis methods in the situation of selecting DCT bases with different size. The atom dictionary used in sparse approximation method is also generated by a union of DCT basis and Dirac basis. This subsection presents the different performances on the “Yoga” motion sequence of these methods and gives the shape reconstruction error on one diagram as shown in Figure 2. The calculation formula of the mean 3D error is denoted as follows:

$$\text{err } S = \|S - \tilde{S}\|_2, \quad (13)$$

where S indicates the real shape sequence and \tilde{S} indicates the estimated one.

Clearly, selecting 12 as the size of DCT basis is the best solution and the shape estimation error is the smallest when recovering the object shape with DCT basis model. But when considering with the sparse coding method, the sparse method is obviously better than the trajectory basis methods even in any size of DCT basis. It is very easy to understand

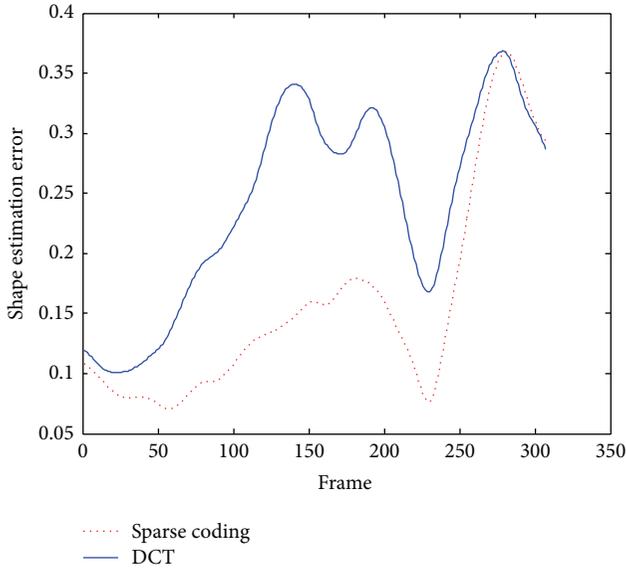


FIGURE 3: Estimation errors of each frame on the “Yoga” sequence. The top line denotes the results curve with DCT method when the size of DCT bases is 12, and another line denotes the results curve with sparse coding method (better viewed in color).

that the atom dictionary contains all predefined trajectory bases even those not used in traditional trajectory basis. The difference between these two methods is that the sparse method only sets the coefficients of the unused trajectory bases to zero and these unused trajectory bases may still be used in other trajectory curves. But, in the trajectory basis model, once the trajectory bases were truncated in a fixed size, the trajectory curves could only be represented by these truncated bases.

5.3. Estimation Errors. This subsection gives a statistical comparison between the well-size DCT trajectory basis method [19, 20] and the sparse approximation method. The truncated size of the trajectory bases is learned by the previous work done by other researchers and it has been proved to be reliable. By using the obtained data from the “Yoga” sequences, the mean 3D error of each frame is computed with these two methods, respectively. From the above subsection, the best solution of the DCT basis model could be found. So the size of the DCT trajectory basis in this scene is 12 and the penalty coefficient of the sparse method is 0.1. The shape estimation errors of each frame are shown in Figure 3.

From Figure 3, it is very clear to know that, despite using the traditional DCT trajectory basis method in the best situation, the shape estimation errors of each frame obtained by sparse approximation method are smaller than the estimation errors obtained by DCT basis model. That is to say, the sparse approximation method performs better than the DCT trajectory basis method in dealing with the NRSfM problems in the “Yoga” scene.

5.4. Tests on Different Scenes. To verify the effectiveness of the sparse coding method, this paper repeats the experiment

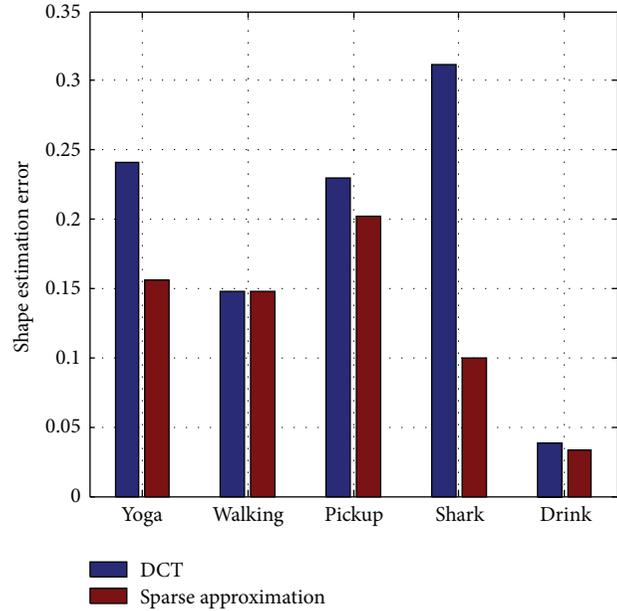


FIGURE 4: Shape estimation error on different scenes (better viewed in color).

in different scenes. The experiment is done with some real sequences which include “Yoga,” “walking,” “pickup,” “shark,” and “drink.” The best size of DCT trajectory bases is obtained from the previous experiments which have been proved to be effective. This experiment applies the best solution of DCT basis model to every scene and compares the results of DCT basis model with the sparse approximation method. From previous works, the best DCT basis size of “Yoga” is 12, “walking” is 8, “pickup” is 12, “shark” is 2, and “drinking” is 10. The penalty coefficient of the sparse approximation method is 0.1. The experimental result is shown in Figure 4.

From Figure 4, one can find that the shape estimation errors of every scene obtained by DCT trajectory basis model are higher than the estimation errors obtained by sparse approximation method. That is to say, sparse approximation method is better than DCT trajectory basis method in most of the scenes; even the size of DCT trajectory basis is selected in the best situation. At worst, the former will get a similar result with the latter in the situation that the truncated DCT bases can represent the trajectory curves of the feature points largely. A large number of experiments have been done to prove that the sparse approximation method is a better application in the NRSfM problem.

5.5. Sample Shape Reconstruction Results. To verify the effectiveness of the sparse approximation method that this paper mentioned, a lot of experiments have been done. At this subsection of the paper, some experimental results were shown clearly to enhance the persuasion. This subsection gives out the shape reconstruction results of the “Yoga,” “shark,” and “Stretch” sequences using the DCT method and the sparse approximation method, respectively. The truncated size of DCT basis in the “Yoga” experiment is 12, and the penalty

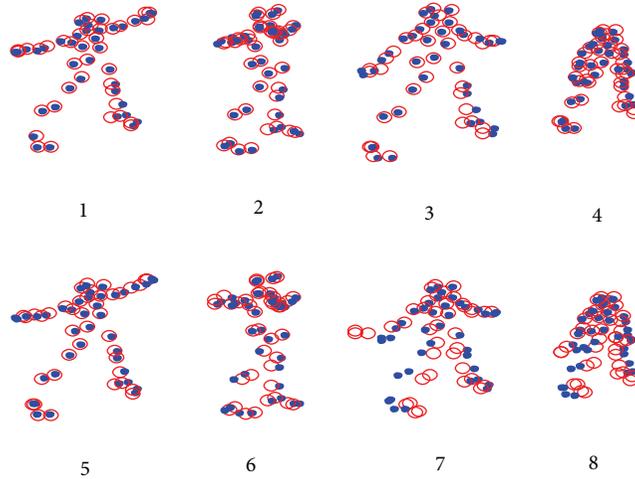


FIGURE 5: 3D reconstruction results on the “Yoga” sequence with two methods, respectively. The red dots denote the ground truth coordinates and the blue circles denote the estimation coordinates. The top row is the reconstruction result with sparse approximation and the bottom row is with the DCT trajectory basis method. Both of the results of these two methods are shown with the 1_{th} , 60_{th} , 120_{th} , and 180_{th} frames (better viewed in color).

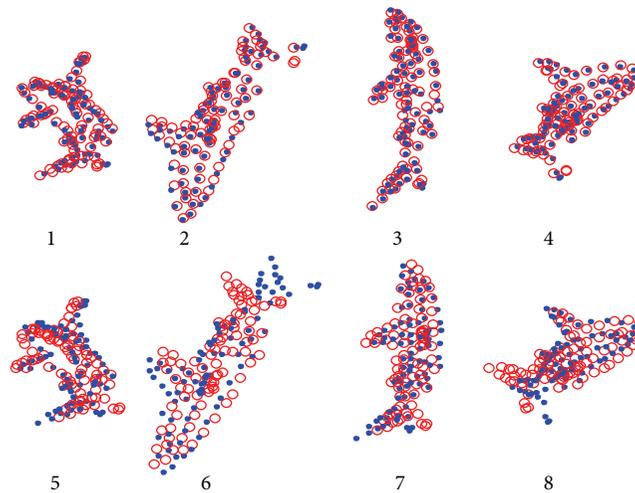


FIGURE 6: 3D reconstruction results on the “shark” sequence with two methods, respectively. The red dots denote the ground truth coordinates and the blue circles denote the estimation coordinates. The top row is the reconstruction result with sparse approximation and the bottom row is with the DCT trajectory basis method. Both of the results of these two methods are shown with the 1_{th} , 60_{th} , 120_{th} , and 180_{th} frames through different visualizations (better viewed in color).

coefficient of the sparse method is 0.1. The experimental results are seen in Figure 5. The truncated size of DCT basis in the “shark” experiment is 2, and the penalty coefficient of the sparse method is 0.1. The experimental results are seen in Figure 6.

To verify the effectiveness of sparse approximation method, this subsection also gives out the experimental result of “Stretch” sequence; Figure 7 shows the shape reconstruction result of the “Stretch” sequence with two methods, respectively. The truncated size of DCT basis in the “Stretch” experiment is 12 and the DCT basis method will get the best solution in the DCT basis model situation, and the penalty coefficient of the sparse method is 0.1.

From Figures 5, 6, and 7, one can easily find that the reconstruction points with sparse approximation method are more close to the original 3D feature points. The experimental results have shown the advantage of the sparse approximation method. The approach mentioned in this paper is obviously more effective than the traditional DCT trajectory basis methods.

6. Conclusions

This paper introduces a novel sparse approximation method to resolve the NRSfM problem. It is quite easy to understand and is guaranteed to get an optimal solution. It shows that

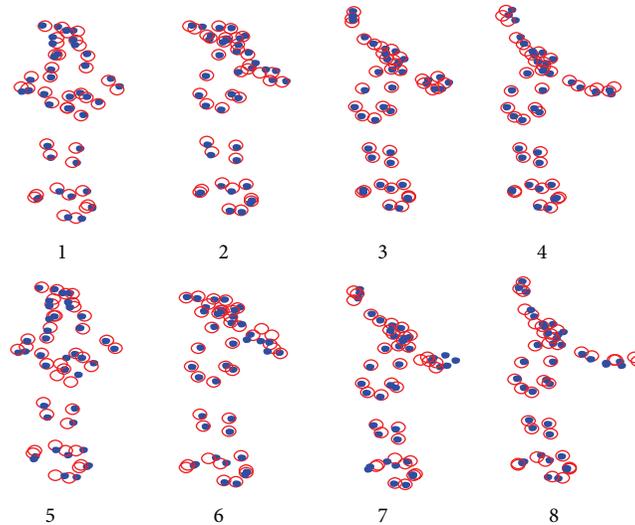


FIGURE 7: 3D reconstruction results on the “Stretch” sequence with two methods, respectively. The red dots denote the ground truth coordinates and the blue circles denote the estimation coordinates. The top row is the reconstruction result with sparse approximation and the bottom row is with the DCT trajectory basis method. Both of the results of these two methods are shown with the 65_{th}, 105_{th}, 145_{th}, and 185_{th} frames through different visualizations (better viewed in color).

one needs not to consider the size of truncated DCT bases via using the sparse approximation method. In this paper, only the union of DCT and Dirac function is applied into the reconstruction of 3D motion object. It is expected that camera rotation matrix estimation error can be obtained more accurately.

In this paper, the present experiments are all in a situation of orthographic camera models. And thanks to recent progress in signal sparse coding, the proposed solution can be easily applied into solving the NRSfM problem. This paper just gives out one thought to solve the size of the trajectory bases. Some other ideas can also be used in this problem. At the same time, obtaining a rotation matrix accurately will help to get a satisfied result.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61272311) and is also supported in part by the Natural Science Foundation of Zhejiang Province (LY13F020042, LY14F010022, and Z15F020017) and 521 Project of Zhejiang Sci-Tech University.

References

- [1] J. K. Aggarwal and Q. Cai, “Human motion analysis: a review,” in *Proceedings of the IEEE Nonrigid and Articulated Motion Workshop*, pp. 90–102, June 1997.
- [2] C. Nastar and N. Ayache, “Frequency-based nonrigid motion analysis: application to four dimensional medical images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 11, pp. 1067–1079, 1996.
- [3] C. Studholme, D. L. G. Hill, and D. J. Hawkes, “An overlap invariant entropy measure of 3D medical image alignment,” *Pattern Recognition*, vol. 32, no. 1, pp. 71–86, 1999.
- [4] T. B. Moeslund and E. Granum, “A survey of computer vision-based human motion capture,” *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.
- [5] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, “Nonrigid structure from motion in trajectory space,” in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS '08)*, pp. 42–48, December 2008.
- [6] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, “Trajectory space: a dual representation for nonrigid structure from motion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1442–1456, 2011.
- [7] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. Kriegman, and S. Belongie, “Generalized non-metric multidimensional scaling,” *Journal of Machine Learning Research*, vol. 2, pp. 11–18, 2007.
- [8] Y. Wang, J. Cheng, J. Zheng, Y. Xiong, and H. Zhang, “Analysis of wavelet basis selection in optimal trajectory space finding for 3D non-rigid structure from motion,” *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 12, no. 3, Article ID 1450023, 14 pages, 2014.
- [9] C. Bregler, A. Hertzmann, and H. Biermann, “Recovering nonrigid 3D shape from image streams,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, pp. 690–696, June 2000.
- [10] L. Torresani, A. Hertzmann, and C. Bregler, “Nonrigid structure-from-motion: estimating shape and motion with hierarchical priors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 878–892, 2008.

- [11] J. Xiao, J.-X. Chai, and T. Kanade, "A closed-form solution to non-rigid shape and motion recovery," in *Computer Vision—ECCV 2004*, pp. 573–587, Springer, Berlin, Germany, 2004.
- [12] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 101–122, 2014.
- [13] Y. Zhu and S. Lucey, "Convolutional sparse coding for trajectory reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 529–540, 2013.
- [14] Q. Barthélemy, A. Larue, and J. I. Mars, "Decomposition and dictionary learning for 3D trajectories," *Signal Processing*, vol. 98, pp. 423–437, 2014.
- [15] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.
- [16] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Neural Information Processing Systems*, pp. 801–808, 2007.
- [17] E. J. Candes and Y. Plan, "A probabilistic and RIPless theory of compressed sensing," *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7235–7254, 2011.
- [18] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6–18, 2006.
- [19] J. Valmadre and S. Lucey, "General trajectory prior for non-rigid reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1394–1401, IEEE, Providence, RI, USA, June 2012.
- [20] M. Chen, G. AlRegib, and B.-H. Juang, "Trajectory triangulation: 3D motion reconstruction with ℓ_1 optimization," in *Proceedings of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '11)*, pp. 4020–4023, Prague, Czech Republic, May 2011.

Research Article

Unmanned Aerial Vehicle Navigation Using Wide-Field Optical Flow and Inertial Sensors

Matthew B. Rhudy,¹ Yu Gu,² Haiyang Chao,³ and Jason N. Gross⁴

¹Division of Engineering, Pennsylvania State University, Reading, PA 19610, USA

²Department of Mechanical and Aerospace Engineering and Lane Department of Computer Science and Electrical Engineering at WVU, Morgantown, WV 26506, USA

³Aerospace Engineering Department, University of Kansas, Lawrence, KS 66045, USA

⁴Department of Mechanical and Aerospace Engineering at West Virginia University, Morgantown, WV 26506, USA

Correspondence should be addressed to Matthew B. Rhudy; matthew.rhudy@gmail.com

Received 16 January 2015; Revised 12 March 2015; Accepted 30 March 2015

Academic Editor: Liwei Shi

Copyright © 2015 Matthew B. Rhudy 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 offers a set of novel navigation techniques that rely on the use of inertial sensors and wide-field optical flow information. The aircraft ground velocity and attitude states are estimated with an Unscented Information Filter (UIF) and are evaluated with respect to two sets of experimental flight data collected from an Unmanned Aerial Vehicle (UAV). Two different formulations are proposed, a full state formulation including velocity and attitude and a simplified formulation which assumes that the lateral and vertical velocity of the aircraft are negligible. An additional state is also considered within each formulation to recover the image distance which can be measured using a laser rangefinder. The results demonstrate that the full state formulation is able to estimate the aircraft ground velocity to within 1.3 m/s of a GPS receiver solution used as reference “truth” and regulate attitude angles within 1.4 degrees standard deviation of error for both sets of flight data.

1. Introduction

Information about the velocity and attitude of an aircraft is important for purposes such as remote sensing [1], navigation, and control [2]. Traditional low-cost aircraft navigation relies on the use of both inertial sensors and Global Positioning System (GPS) [3–5]. While GPS can provide useful information to an aircraft system, this information is not always available or reliable in certain situations, such as flying in urban environments or other GPS-denied areas (e.g., under radio-frequency jamming or strong solar storm). GPS is not self-contained within the aircraft system; rather the information comes from external satellites. Insects, such as the honeybee, have demonstrated impressive capabilities in flight navigation without receiving external communications [6]. One significant information source that is used by insects as well as birds is vision [6–8]. This information can also be made available to an aircraft through the use of onboard video cameras. The challenge with this information rich data

is correctly processing and integrating the vision data with the other onboard sensor measurements [9].

Vision data can be processed using feature detection algorithms such as the Scale-Invariant Feature Transform (SIFT) [10] to obtain optical flow vectors, as well as other techniques. Optical flow is useful for aircraft systems because it is rich in navigation information, simple to represent, and easy to compute [11]. One of the benefits of this information is that it can be used in order to extract velocity information about the aircraft, which in turn can be used for aircraft positioning. This optical flow information has been used for autonomous navigation applications such as relative heading and lateral position estimation of a quadrotor helicopter [12, 13]. Another work has considered the use of optical flow for UAV take-off and landing [14] and landmark navigation [15]. Another potential benefit of optical flow is that it implicitly contains information about the aircraft attitude angles. This implicit information has been used in related work for UAV attitude estimation using horizon detection and optical flow

along the horizon line [16, 17] and pose estimation for a hexacopter [18], a lunar rover [19], and spacecraft [20]. While this work is useful, these vehicles contain significantly different dynamic characteristics than a typical airplane. Due to this, more analysis of the application of optical flow for airplane applications is necessary.

This work presents a combined velocity and attitude estimation algorithm using wide-field optical flow for airplanes that does not require horizon detection, which is useful because the horizon does not need to be visible in the image frame in order to obtain attitude information. The algorithm relies on the optical flow computed using a downward facing video camera, measurements from a laser range finder and an Inertial Measurement Unit (IMU) that are mounted in parallel to the camera axis, and a flat ground assumption to determine information about the aircraft velocity and attitude. Many of the existing experiments for optical flow and inertial sensor fusion are done using helicopter platforms and focus on position and velocity estimation [21, 22]. This work considers an airplane system rather than a helicopter, which contains a significantly different flight envelope and dynamics. Additionally, the regulation of attitude information through the use of optical flow is considered, which is not typically done in existing applications. This work takes advantage of all detected optical flow points in the image plane, including wide-field optical flow points which were often omitted in previous works [23–25]. These wide-field optical flow points are of significant importance for attitude estimation, since they contain roll and pitch information that is not observable from the image center. Although this work considers the use of a laser range finder to recover the distance between the image scene and the camera, it is possible to determine this information using other techniques [26]. In fact, it has been demonstrated that the scale is an observable mode for the vision and IMU data fusion problem [27]. The presented formulation was originally offered in its early stages of development in [28]. Since this original publication, the implementation and tuning of the formulation have been refined, and additional results have been generated. In particular, a simplified formulation is offered which reduces the filter states, and the inclusion of a range state is considered. The main contribution of this paper is the analysis of a stable vision-aided solution for the velocity and attitude determination without the use of GPS. This solution is verified with respect to two sets of actual UAV flight testing data.

The rest of this paper is organized as follows. Section 2 presents the different considered formulations and framework for this problem. Section 3 describes the experimental setup which was used to collect data for this study. The results are offered in Section 4 followed by a conclusion in Section 5.

2. Problem Formulation

2.1. Optical Flow Equations. Optical flow is the projection of 3D relative motion into a 2D image plane. Using the pinhole camera model, the 3D position (η_x, η_y, η_z) in the 3D camera

body frame can be mapped into the 2D image plane with coordinates (μ, ν) using

$$\begin{aligned}\mu &= f \frac{\eta_x}{\eta_z}, \\ \nu &= f \frac{\eta_y}{\eta_z},\end{aligned}\tag{1}$$

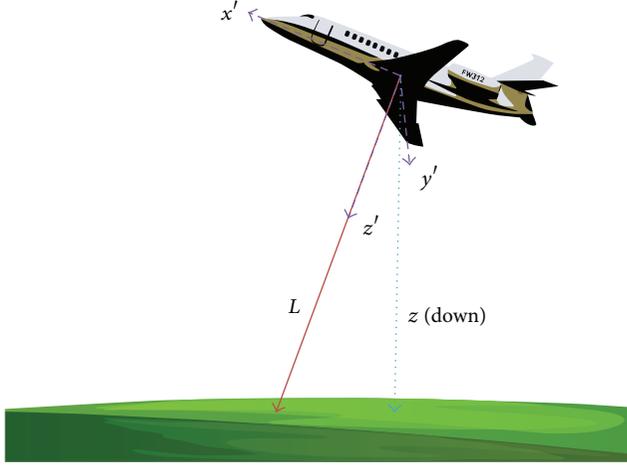
where μ , ν , and f are given in pixels and f is the focal length. For a downward looking camera that is parallel to the aircraft z -axis, and with a level and flat ground assumption, the optical flow equations have been derived [29]:

$$\begin{aligned}\begin{bmatrix} \dot{\mu} \\ \dot{\nu} \end{bmatrix} &= \frac{f + \nu \tan \phi - \mu \tan \theta / \cos \phi}{\eta_z} \begin{bmatrix} -u + \frac{\mu w}{f} \\ -v + \frac{\nu w}{f} \end{bmatrix} \\ &- \begin{bmatrix} fq - r\nu - \frac{p\mu\nu}{f} + \frac{q\mu^2}{f} \\ -fp + r\mu - \frac{p\nu^2}{f} + \frac{q\nu\nu}{f} \end{bmatrix},\end{aligned}\tag{2}$$

where ϕ , θ are the roll and pitch angles, p , q , r are the roll, pitch, and yaw body-axis angular rates, u , v , w , are the body-axis ground velocity components of the aircraft, and $\dot{\mu}$, $\dot{\nu}$ are the components of optical flow in the 2D image plane, given in pixels/sec. This equation captures the relationship between optical flow at various parts of the image plane with other pieces of navigation information. By considering only the area close to the image center ($\mu \approx 0$, $\nu \approx 0$), the narrow-field optical flow model can be simplified [23–25]; however, this removes the roll and pitch dependence of the equation and is therefore not desirable for attitude estimation purposes.

2.2. State Space Formulation and Stochastic Modeling. This work considers the simultaneous estimation of body-axis ground velocity components (u, v, w) and Euler attitude angles (ϕ, θ, ψ) . This estimation is performed through the fusion of Inertial Measurement Unit (IMU) measurements of body-axis accelerations (a_x, a_y, a_z) and angular rates (p, q, r) , laser rangefinder range measurements (L) , and n sets of optical flow measurements $(\dot{\mu}, \dot{\nu})_i$, where $i = 1, 2, \dots, n$. The value of n varies with each time step based on how many features in the frame can be used for optical flow calculation. Using these values, the state space model of the system is formulated with the following state vector, \mathbf{x} , bias state vector, \mathbf{b} , input vector, \mathbf{u} , optical flow input vectors, \mathbf{d}_i , and output vectors, \mathbf{z}_i :

$$\begin{aligned}\mathbf{x} &= [u \ v \ w \ \phi \ \theta \ \psi \ \mathbf{b}^T]^T, \\ \mathbf{b} &= [b_{a_x} \ b_{a_y} \ b_{a_z} \ b_p \ b_q \ b_r \ b_L]^T, \\ \mathbf{u} &= [a_x \ a_y \ a_z \ p \ q \ r \ L]^T, \\ \mathbf{d}_i &= [\mu \ \nu]_i^T, \\ \mathbf{z}_i &= [\dot{\mu} \ \dot{\nu}]_i^T.\end{aligned}\tag{3}$$



- ←-- Body coordinates
- ←--- NED coordinates
- ←- Range coordinate

FIGURE 1: Diagram of the range coordinate.

A diagram describing the definition of the range coordinate, L , is provided in Figure 1. Note that the range coordinate, L , is equivalent to the camera z coordinate, η_z .

In order to determine the dynamics of the velocity states, the time derivative of the velocity vector observed from the fixed navigation frame is equal to the time rate of change as observed from the moving body axis frame plus the change caused by rotation of the frame [30]:

$$\frac{d}{dt} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix}. \quad (4)$$

The IMU measures the acceleration with respect to the fixed gravity vector, as in

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \frac{d}{dt} \begin{pmatrix} u \\ v \\ w \end{pmatrix} + C_n^b \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}, \quad (5)$$

where C_n^b is the rotation matrix from the navigation frame to the body frame:

$$C_n^b = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & \sin \phi \cos \theta \\ \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi & \cos \phi \cos \theta \end{bmatrix}. \quad (6)$$

Combining these results gives the dynamics for the velocity states [31]:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw + a_x - g \sin \theta \\ pw - ru + a_y + g \sin \phi \cos \theta \\ qu - pv + a_z + g \cos \phi \cos \theta \end{bmatrix}. \quad (7)$$

The dynamics of the attitude states are defined using [32]

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \quad (8)$$

To define the dynamics for the bias parameters, a first-order Gauss-Markov noise model was used. In a related work [33], the Allan deviation [34] approach presented in [35, 36] was used to determine the parameters of the first-order Gauss-Markov noise model for the dynamics of the bias on each IMU channel. The Gauss-Markov noise model for each sensor measurement involves two parameters: a time constant and a variance of the wide-band sensor noise. Using this model, the dynamics for the bias parameters are given by

$$\mathbf{b}_k = \mathbf{b}_{k-1} e^{-T_s/\tau} + \mathbf{n}_{k-1}, \quad (9)$$

where τ is a vector of time constants and \mathbf{n} is a zero-mean noise vector with variance given by a diagonal matrix of

the variance terms for each sensor. The time constant and variance terms were calculated in [33] for each channel of the same IMU that was considered for this study.

The state dynamic equations have been defined in continuous-time using the following format:

$$\dot{\mathbf{x}} = \mathbf{f}_c(\mathbf{x}, \mathbf{u}), \quad (10)$$

where \mathbf{f}_c is the nonlinear continuous-time state transition function. In order to implement these equations in a discrete-time filter, a first-order discretization is used [37]:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + T_s \mathbf{f}_c(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \triangleq \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}), \quad (11)$$

where k is the discrete time index, \mathbf{f} is the nonlinear discrete-time state transition function, and T_s is the sampling time of the system.

To formulate the observation equations, optical flow information is utilized. In particular, each optical flow point identified from vision data consists of four values: μ , ν , $\dot{\mu}$, $\dot{\nu}$. These values are obtained using a point matching method [38] and the Scale-Invariant Feature Transform (SIFT) algorithm [10]. Note that the method for optical flow generation is not the emphasis of this research [38]; therefore, any other optical flow algorithm can be used similarly within the proposed estimator, without any loss of generality.

During the state estimation process, the image plane coordinates (μ, ν) are taken as inputs to the observation

equation, allowing the optical flow $(\dot{\mu}, \dot{\nu})$ to be predicted at that point in the image plane using (2), where η_z is provided by the laser rangefinder measurement, L . These computed observables are then compared with the optical flow measurements of $(\dot{\mu}, \dot{\nu})$ from the video in order to determine how to update the states. Since multiple optical flow points can be identified within a single time step, this creates a set of n_k observation equations, where n_k is the number of optical flow points at time step k .

Since (7) and (8) are derived from kinematics, the only uncertainty that must be modeled is due to the input measurements. Therefore, the input vector is given by

$$\mathbf{u}_{k-1} = \hat{\mathbf{u}}_{k-1} + \mathbf{b}_{k-1}, \quad (12)$$

where $\hat{\mathbf{u}}$ is the measured input vector and \mathbf{b} is the vector of sensor biases which follow a first order Gauss-Markov noise model as determined in [33].

The uncertainty in the measurements is due to the errors in the optical flow estimation from the video. It is assumed that each optical flow measurement \mathbf{y}_i has an additive measurement noise vector, \mathbf{v}_i , with corresponding covariance matrix, \mathbf{R}_i . For this study, it is also assumed that each optical flow measurement carries equal uncertainty and that errors along the two component directions of the image plane also have equal uncertainty and are uncorrelated; that is,

$$\mathbf{R}_i = \mathbf{R} = R\mathbf{I}, \quad (13)$$

where R is the scalar uncertainty of the optical flow measurements and \mathbf{I} is a 2×2 identity matrix.

2.3. Simplified Formulation. The motion of a typical airplane is mostly in the forward direction, that is, the speed of the aircraft is primarily contained in the component, u , while v and w are small. With this idea, assuming that v and w are zero, the formulation is simplified to the following state vector, \mathbf{x} , bias state vector, \mathbf{b} , input vector, \mathbf{u} , optical flow input vectors, \mathbf{d}_i , and output vectors, \mathbf{z}_i :

$$\begin{aligned} \mathbf{x} &= [u \ \phi \ \theta \ \mathbf{b}^T]^T, \\ \mathbf{b} &= [b_{a_x} \ b_p \ b_q \ b_r \ b_L]^T, \\ \mathbf{u} &= [a_x \ p \ q \ r \ L]^T, \\ \mathbf{d}_i &= [\mu \ \nu]_i^T, \\ \mathbf{z}_i &= \dot{\mu}_i. \end{aligned} \quad (14)$$

Note that this simplified formulation removes the v and w states which removes the need for y -axis and z -axis acceleration measurements. Since the yaw state is not contained in any of the state or observation equations it has also been removed. Due to the assumption that v and w are zero,

only the x -direction of optical flow is relevant. With these simplifications, the state dynamics become

$$\begin{aligned} \dot{u} &= a_x - g \sin \theta, \\ \dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta, \\ \dot{\theta} &= q \cos \phi - r \sin \phi. \end{aligned} \quad (15)$$

The dynamics of the bias states remain the same as in the full formulation except the corresponding bias states for a_y and a_z have been removed. The observation equations from (2) are simplified to be

$$\begin{aligned} \dot{\mu} &= \frac{-u(f + v \tan \phi - \mu \tan \theta / \cos \phi)}{\eta_z} - fq + rv \\ &+ \frac{p\mu\nu}{f} - \frac{q\mu^2}{f}. \end{aligned} \quad (16)$$

The advantage of considering this simplified formulation is primarily to reduce the computational complexity of the system. The processing of vision data leading to a relatively large number of measurement updates can significantly drive up the computation time of the system, particularly for higher sampling rates. This simplified formulation not only reduces the computation time through a reduction of states, but also significantly reduces the processing and update time for optical flow measurements since only the forward component of flow is used. This formulation could be more practical than the full state formulation for real-time implementation, especially on systems which are limited in onboard computational power due, for example, to cost or size constraints.

2.4. Inclusion of a Range State. It is possible to include a state to estimate the range in order to recover the scale of the optical flow images. To determine the dynamics of the range state, the flat ground assumption is used. With this assumption, consider the projection of the range vector onto the Earth-fixed z -axis, that is, “down,” as shown in Figure 1, by taking the projection through both the roll and pitch angles of the aircraft:

$$z = -L \cos \phi \cos \theta. \quad (17)$$

Here, the negative sign is used because the L coordinate is always positive, while the z coordinate will be negative when the aircraft is above the ground (due to the “down” convention). Taking the derivative with respect to time yields

$$\dot{z} = -\dot{L} \cos \phi \cos \theta + \dot{\phi} L \sin \phi \cos \theta + \dot{\theta} L \cos \phi \sin \theta. \quad (18)$$

Compare this z -velocity equation with that obtained from rotating aircraft body velocity components into the Earth-fixed frame:

$$\dot{z} = -u \sin \theta + v \sin \phi \cos \theta + w \cos \phi \cos \theta. \quad (19)$$

Equating these two expressions for z -velocity gives

$$\begin{aligned} -\dot{L} \cos \phi \cos \theta + \dot{\phi} L \sin \phi \cos \theta + \dot{\theta} L \cos \phi \sin \theta \\ = -u \sin \theta + v \sin \phi \cos \theta + w \cos \phi \cos \theta. \end{aligned} \quad (20)$$

Simplifying this relationship leads to

$$\dot{L} = \dot{\phi}L \tan \phi + \dot{\theta}L \tan \theta + u \sec \phi \tan \theta - v \tan \phi - w. \quad (21)$$

Substituting in the dynamics for the roll and pitch angles and simplifying leads to the following expression for the range state dynamics:

$$\begin{aligned} \dot{L} = & u \sec \phi \tan \theta - v \tan \phi - w \\ & + L [p \tan \phi + q \sec \phi \tan \theta]. \end{aligned} \quad (22)$$

Note that, for level conditions, that is, roll and pitch angles are zero, the equation reduces to

$$\dot{L} = -w \quad (23)$$

which agrees with physical intuition. In order to implement the range state in the simplified formulation, the following expression can be used:

$$\dot{L} = u \sec \phi \tan \theta + L [p \tan \phi + q \sec \phi \tan \theta]. \quad (24)$$

2.5. Information Fusion Algorithm. Due to the nonlinearity, nonadditive noise and numbers of multiple optical flow measurements ranging from 0 to 300 per frame with a mean of 250, the Unscented Information Filter (UIF) [39–41] was selected for the implementation of this algorithm [42]. The advantage of the information filtering framework over Kalman filtering is that redundant information vectors are additive [39–41]; therefore, the time-varying number of outputs obtained from optical flow can easily be handled with relatively low computation, since the coupling between the errors in different optical flow measurements is neglected. The UIF algorithm is summarized as follows [41].

Consider a discrete time nonlinear dynamic system of the form

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}, \quad (25)$$

with measurement equations of the form

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{d}_k) + \mathbf{v}_k, \quad (26)$$

where \mathbf{h} is the observation function and \mathbf{w} and \mathbf{v} are the zero-mean Gaussian process and measurement noise vectors. At each time step, sigma-points are generated from the prior distribution using

$$\begin{aligned} \chi_{k-1} \\ = [\hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} + \sqrt{N + \lambda} \sqrt{\mathbf{P}_{k-1}} \quad \hat{\mathbf{x}}_{k-1} - \sqrt{N + \lambda} \sqrt{\mathbf{P}_{k-1}}], \end{aligned} \quad (27)$$

where N is the total number of states and λ is a scaling parameter [42]. Now, the sigma-points are predicted using

$$\chi_{k|k-1}^{(i)} = \mathbf{f}(\chi_{k-1}^{(i)}, \mathbf{u}_{k-1}), \quad i = 0, 1, \dots, 2N, \quad (28)$$

where (i) denotes the i th column of a matrix. The *a priori* statistics are then recovered:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \sum_{i=0}^{2N} \boldsymbol{\eta}_i^m \chi_{k|k-1}^{(i)}, \\ \mathbf{P}_{k|k-1} &= \mathbf{Q}_{k-1} \\ &+ \sum_{i=0}^{2N} \boldsymbol{\eta}_i^c (\chi_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1}) (\chi_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1})^T, \end{aligned} \quad (29)$$

where \mathbf{Q} is the process noise covariance matrix, and $\boldsymbol{\eta}_i^m$ and $\boldsymbol{\eta}_i^c$ are weight vectors [42]. Using these predicted values, the information vector, \mathbf{y} , and matrix, \mathbf{Y} , are determined:

$$\begin{aligned} \hat{\mathbf{y}}_{k|k-1} &= \mathbf{P}_{k|k-1}^{-1} \hat{\mathbf{x}}_{k|k-1}, \\ \mathbf{Y}_{k|k-1} &= \mathbf{P}_{k|k-1}^{-1}. \end{aligned} \quad (30)$$

For each measurement, that is, each optical flow pair, the output equations are evaluated for each sigma-point, as in

$$\begin{aligned} \boldsymbol{\psi}_{k|k-1}^{(i,j)} &= \mathbf{h}(\chi_{k|k-1}^{(i,j)}, \mathbf{d}_k), \\ i &= 0, 1, \dots, 2N, \quad j = 1, \dots, n_k, \end{aligned} \quad (31)$$

where $\boldsymbol{\psi}$ denotes an output sigma-point and the superscript (i, j) denotes the i th sigma-point and the j th measurement. The computed observation is then recovered using

$$\hat{\mathbf{z}}_{k|k-1}^{(j)} = \sum_{i=0}^{2N} \boldsymbol{\eta}_i^m \boldsymbol{\psi}_{k|k-1}^{(i)}, \quad j = 1, \dots, n_k. \quad (32)$$

Using the computed observation, the cross-covariance is calculated:

$$\mathbf{P}_{k|k-1}^{xy(j)} = \sum_{i=0}^{2N} \boldsymbol{\eta}_i^c (\chi_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1}) (\boldsymbol{\psi}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1}^{(j)})^T. \quad (33)$$

Then the observation sensitivity matrix, \mathbf{H} , is determined:

$$\mathbf{H}_k^{(j)} = [\mathbf{P}_{k|k-1}^{-1} \mathbf{P}_{k|k-1}^{xy(j)}]^T, \quad j = 1, \dots, n_k. \quad (34)$$

The information contributions can then be calculated:

$$\begin{aligned} \hat{\mathbf{y}}_k &= \hat{\mathbf{y}}_{k|k-1} \\ &+ \sum_{j=1}^{n_k} (\mathbf{H}_k^{(j)})^T \mathbf{R}_k^{-1} [\mathbf{z}_k^{(j)} - \mathbf{z}_{k|k-1}^{(j)} + \mathbf{H}_k^{(j)} \hat{\mathbf{x}}_{k|k-1}], \\ \mathbf{Y}_k &= \mathbf{Y}_{k|k-1} + \sum_{j=1}^{n_k} (\mathbf{H}_k^{(j)})^T \mathbf{R}_k^{-1} \mathbf{H}_k^{(j)}. \end{aligned} \quad (35)$$

3. Experimental Setup

The research platform used for this study is the West Virginia University (WVU) ‘‘Red Phastball’’ UAV, shown in Figure 2, with a custom GPS/INS data logger mounted inside



FIGURE 2: Picture of WVU “Red Phastball” aircraft.

TABLE 1: Details for WVU Phastball aircraft.

Property	Value
Length	2.2 m
Wingspan	2.4 m
Takeoff mass	11 kg
Payload mass	3 kg
Propulsion system	Dual 90 mm Ducted Fan Motor
Static thrust	60 N
Fuel capacity	Two 5 Ah LiPo batteries
Cruise speed	30 m/s
Mission duration	6 min

the aircraft [28, 43]. Some details for this aircraft are provided in Table 1.

The IMU used in this study is an Analog Devices ADIS-16405 MEMS-based IMU, which includes triaxial accelerometers and rate gyroscopes. Each suite of sensors on the IMU is acquired at 18-bit resolution at 50 Hz over ranges of $\pm 18 g$'s and $\pm 150 \text{ deg/s}$, respectively. The GPS receiver used in the data logger is a Novatel OEM-V1, which was configured to provide Cartesian position and velocity measurements and solution standard deviations at a rate of 50 Hz, with 1.5 m RMS horizontal position accuracy and 0.03 m/s RMS velocity accuracy. An Optic-Logic RS400 laser range finder was used for range measurement with an approximate accuracy of 1 m and range of 366 m, pointing downward. In addition, a high-quality Goodrich mechanical vertical gyroscope is mounted onboard the UAV to provide pitch and roll measurements to be used as sensor fusion “truth” data, with reported accuracy of within 0.25° of true vertical. The vertical gyroscope measurements were acquired at 16-bit resolution with measurement ranges of $\pm 80 \text{ deg}$ for roll and $\pm 60 \text{ deg}$ for pitch.

A GoPro Hero video camera is mounted at the center of gravity of the UAV for flight video collection, pointing downwards. The camera was previously calibrated to a focal length of 1141 pixels [29]. Two different sets of flight data were used for this study, each using different camera settings. The first flight used a pixel size of 1920×1080 and a sampling rate of 29.97 Hz. The second flight used a pixel size of 1280×720 and a sampling rate of 59.94 Hz. All the other sensor data were collected at 50 Hz and resampled to the camera time for postflight validation after manual synchronization.

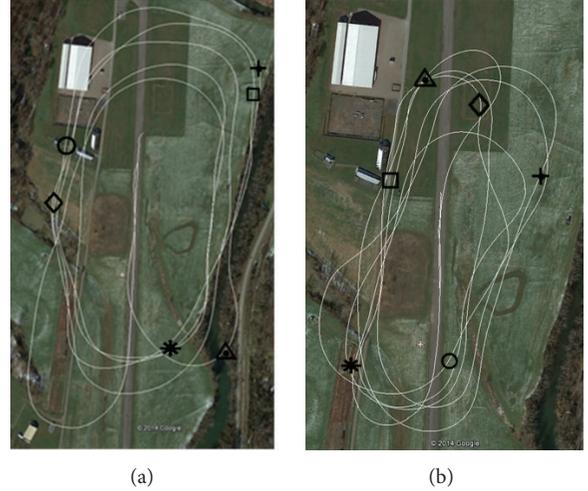


FIGURE 3: Flight trajectories for Flight #1 (a) and Flight #2 (b), © 2014 Google.

4. Experimental Results

4.1. Flight Data. Two sets of flight data from the WVU “Red Phastball” aircraft were used in this study. Each flight consists of approximately 5 minutes of flight. The top-down flight trajectories from these two data sets are overlaid on a Google Earth image of the flight test location in Figure 3. Six different unique markers have been placed in Figure 3 in order to identify specific points along the trajectory. These markers will be used in future figures in order to synchronize the presentation of data.

4.2. Selection of Noise Assumptions for Optical Flow Measurements. Since the noise properties of the IMU have been established from previous work [33], only the characteristics of the uncertainty in the laser range and optical flow measurements need to be determined. The uncertainty in the laser range finder measurement is modeled as 1 m zero-mean Gaussian noise, based on the manufacturer’s reported accuracy of the sensor. The optical flow errors are a bit more difficult to model. Due to this difficulty, different assumptions of the optical flow uncertainty were considered. Using both sets of the flight data, the full state UIF was executed for each assumption of optical flow uncertainty. To evaluate the performance of the filter, the speed measurements were compared with reference measurements from GPS which have been mapped into the aircraft frame using roll and pitch measurements from the vertical gyroscope and approximating the yaw from the heading as determined by GPS. The roll and pitch estimates were compared with the measurements from the vertical gyroscope. Due to the possibility of alignment errors, only standard deviation of error was considered. Each of these errors was calculated for each set of flight data, and the results are offered in Figure 4.

Figure 4 shows how changing the assumption on the optical flow uncertainty affects the estimation performance of the total ground speed, roll angle, and pitch angle. The relatively

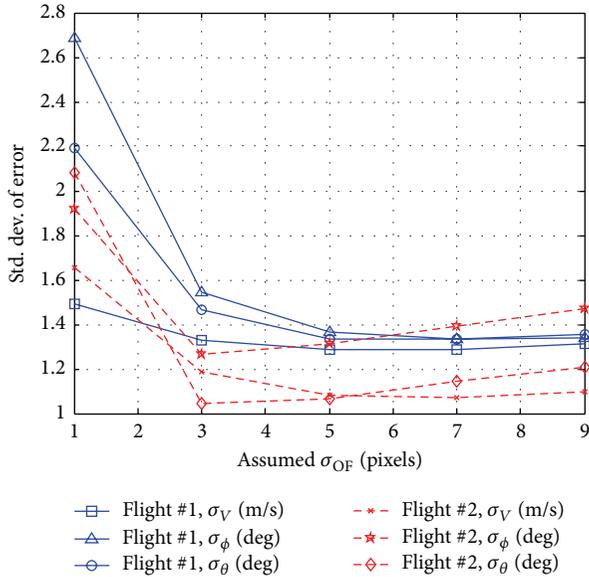


FIGURE 4: Comparison of errors for different assumed optical flow uncertainties.

flat region in Figure 4 for assumed optical flow standard deviations from approximately 3 to 9 pixels indicates that this formulation is relatively insensitive to tuning of these optical flow errors. It is also interesting to note in Figure 4 that Flight #1 and Flight #2 have optimum performance at different values of R . This however makes sense, as Flight #2 has twice the frame rate as Flight #1; therefore, the assumed noise characteristics should be one half that of Flight #1. From Figure 4, the optical flow uncertainties were selected to be $R = 5^2$ pixels² for Flight #1 and $R = 2.5^2$ pixels² for Flight #2.

4.3. Full State Formulation Estimation Results. Using each set of flight data, the full state formulation using UIF was executed. The estimated components of velocity are shown for Flight #1 in Figure 5 and for Flight #2 in Figure 6. These estimates from the UIF are offered with respect to comparable reference values from GPS, which were mapped into the aircraft frame using roll and pitch measurements from the vertical gyroscope and approximating the yaw angle with the heading angle obtained from GPS. From each of these figures, the following observations can be made. The forward velocity, u , is reasonably captured by the estimation. The lateral velocity, v , and vertical velocity, w , however, demonstrate somewhat poor results. This does however make sense, as the primary direction of flight is forward, thus resulting in good observability characteristics in the optical flow in the forward direction, while the signal-to-noise ratio (SNR) for the lateral and vertical directions remains small for most typical flight conditions. However, since these lateral and vertical components are only a small portion of the total velocity, the total speed can be reasonably approximated by this technique. The total speed estimates are shown in Figure 7 for Flight #1 with GPS reference.

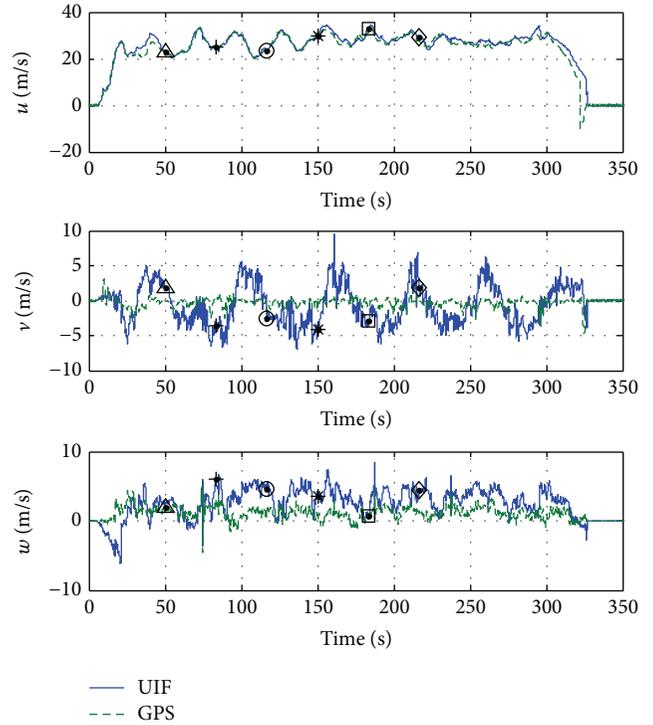


FIGURE 5: Estimated velocity components for Flight #1.

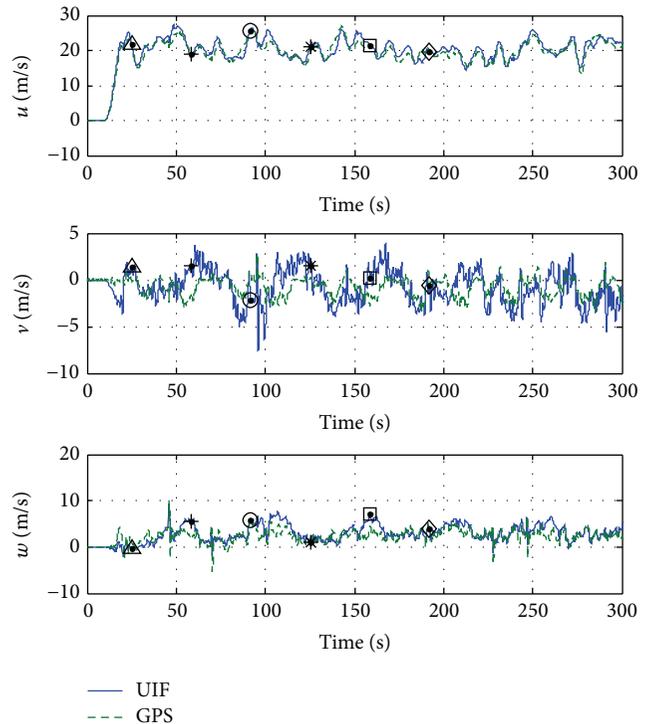


FIGURE 6: Estimated velocity components for Flight #2.

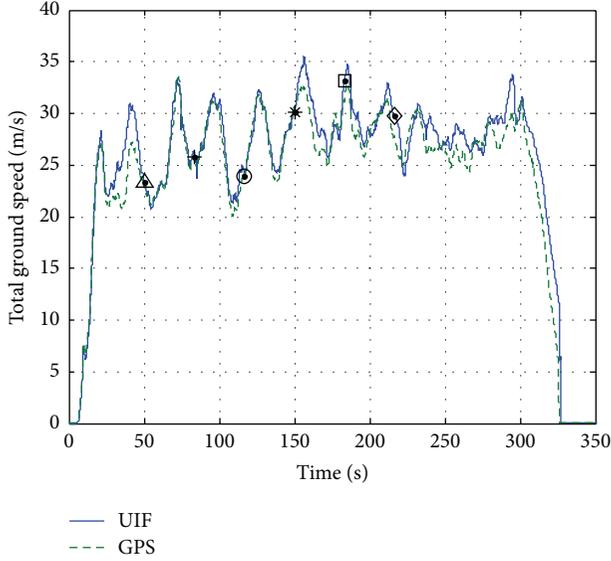


FIGURE 7: Estimated total speed for Flight #1.

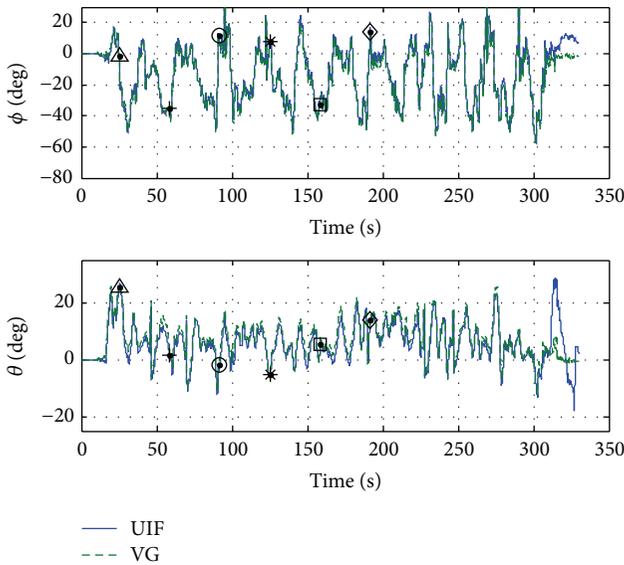


FIGURE 8: Roll and pitch estimation results for Flight #2.

The attitude estimates for the roll and pitch angles are compared with the vertical gyroscope measurements as a reference, as shown in Figure 8. In order to demonstrate the effectiveness of this method in regulating the drift in attitude estimates that occurs with dead reckoning, the estimation errors from the UIF are compared with the errors obtained from dead reckoning attitude estimation. These roll and pitch errors are offered in Figure 9 for Flight #2. Figure 9 demonstrates the effectiveness of the UIF in regulating the attitude errors from dead reckoning.

In order to quantify the estimation results, the mean absolute error and standard deviation of error of the estimates are calculated for the velocity components with respect to the GPS reference and also for the roll and pitch angles with

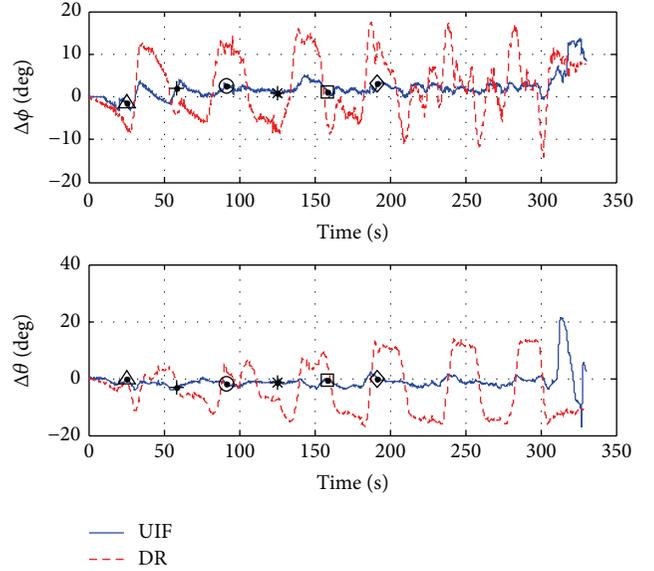


FIGURE 9: Roll and pitch estimation errors as compared to dead reckoning (DR) for Flight #2.

TABLE 2: Flight #1 error statistics for estimated states.

Estimated state	Mean abs.	Standard deviation	Units
u	1.0644	1.2667	m/s
v	2.4699	2.7719	m/s
w	2.1554	1.6554	m/s
V	1.2466	1.2858	m/s
ϕ	1.1553	1.3668	deg
θ	2.1752	1.3339	deg

respect to the vertical gyroscope reference. These statistical results are provided in Table 2 for Flight #1 and Table 3 for Flight #2, where V is the total airspeed as determined by

$$V = \sqrt{u^2 + v^2 + w^2}. \quad (36)$$

It is shown in Tables 2 and 3 that reasonable errors are obtained in both sets of flight data for the velocity and attitude of the aircraft. Larger errors are noted in particular for the lateral velocity state, v , which is due to observability issues in the optical flow. Note that mean errors in the roll and pitch estimation could be due to misalignment between the vertical gyroscope, IMU, and video camera. The attitude estimation accuracy is reported in Tables 2 and 3 similar to the reported accuracy of loosely coupled GPS/INS attitude estimation using similar flight data [43].

4.4. Simplified Formulation Estimation Results. Since it was observed in the full state formulation results that the lateral and vertical estimates were small, the simplified formulation was implemented in order to investigate the feasibility of a simplified version of the filter that estimates only the forward velocity component and assumes the lateral and vertical components are zero. The forward velocity, u , for Flight #1 is offered in Figure 10, while the roll and pitch errors with

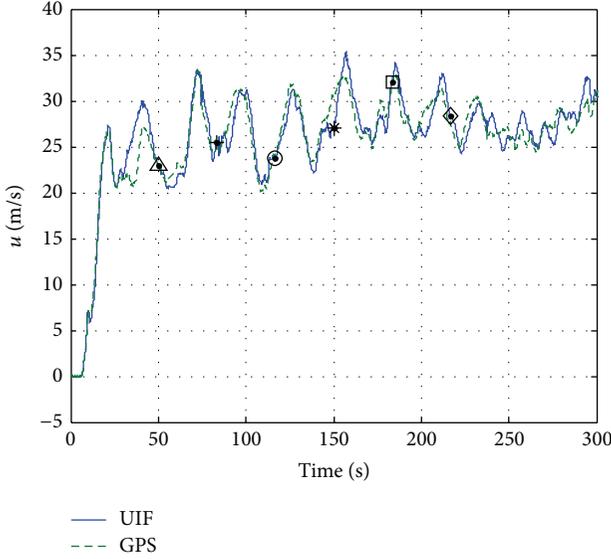


FIGURE 10: Simplified formulation speed estimation results for Flight #1.

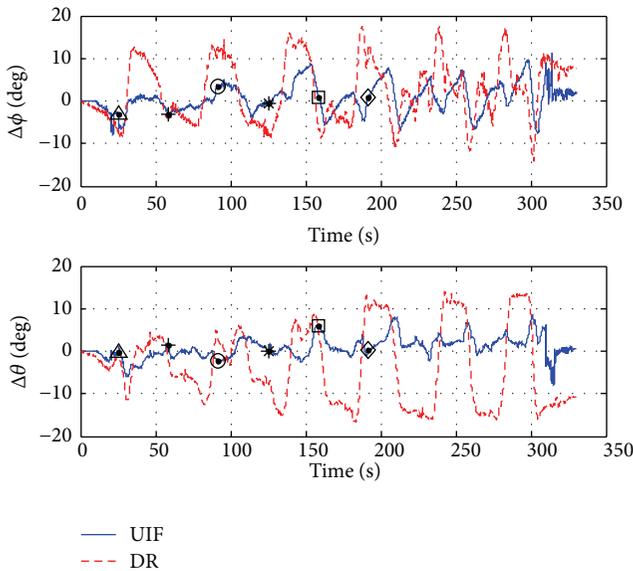


FIGURE 11: Simplified formulation attitude estimates with dead reckoning (DR) for Flight #2.

TABLE 3: Flight #2 error statistics for estimated states.

Estimated state	Mean abs.	Standard deviation	Units
u	1.1299	1.2663	m/s
v	1.5184	1.8649	m/s
w	1.1324	1.3453	m/s
V	1.2087	1.2535	m/s
ϕ	1.8818	1.2782	deg
θ	1.6646	1.1049	deg

respect to the vertical gyroscope measurement are offered in Figure 11 for the UIF and dead reckoning (DR). Additionally,

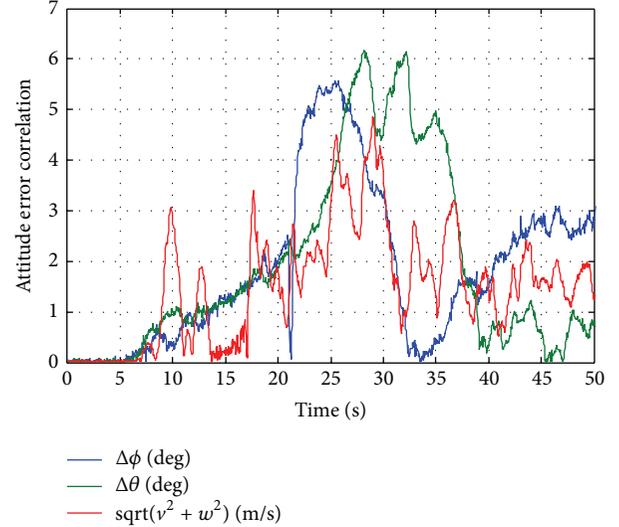


FIGURE 12: Comparison of attitude estimation errors with respect to lateral and vertical velocity.

TABLE 4: Simplified formulation error statistics for Flight #1.

Estimated state	Mean abs.	Standard deviation	Units
u	1.2283	1.5730	m/s
ϕ	1.9901	2.2922	deg
θ	1.9002	2.1881	deg

TABLE 5: Simplified formulation error statistics for Flight #2.

Estimated state	Mean abs.	Standard deviation	Units
u	1.3073	1.5775	m/s
ϕ	2.8402	3.5426	deg
θ	2.0286	2.4691	deg

the mean absolute error and standard deviation of error for these terms are provided in Table 4 for Flight #1 and Table 5 for Flight #2.

It is shown in Tables 4 and 5 that the simplified formulation results in significantly higher attitude estimation errors with respect to the full state formulation. These increased attitude errors are likely due to the assumption that lateral and vertical velocity components are zero. To investigate this possible correlation, the roll and pitch errors are shown in Figure 12 with the magnitude of the lateral and vertical velocity as determined from GPS for a 50-second segment of flight data which includes takeoff. Figure 12 shows that there is some correlation between the attitude estimation errors and the lateral and vertical velocity, though it is not the only source of error for these estimates.

4.5. Results Using Range State. The results for each flight for both the full state formulation and simplified formulation were recalculated with the addition of the range state. The statistical results for these tests are offered in Tables 6–9.

TABLE 6: Flight #1 error statistics for estimated states with range state.

Estimated state	Mean abs.	Standard deviation	Units
u	1.1330	1.3699	m/s
v	2.4387	2.7369	m/s
w	2.1210	1.5998	m/s
V	1.3041	1.3852	m/s
ϕ	1.1599	1.4084	deg
θ	2.1767	1.4064	deg

TABLE 7: Flight #2 error statistics for estimated states with range state.

Estimated state	Mean abs.	Standard deviation	Units
u	1.1444	1.2937	m/s
v	1.5122	1.8529	m/s
w	1.1154	1.3268	m/s
V	1.2112	1.2761	m/s
ϕ	1.8897	1.2845	deg
θ	1.6609	1.1152	deg

TABLE 8: Simplified formulation error statistics for Flight #1 with range state.

Estimated state	Mean abs.	Standard deviation	Units
u	1.2919	1.6256	m/s
ϕ	2.0621	2.3746	deg
θ	1.9277	2.2713	deg

TABLE 9: Simplified formulation error statistics for Flight #2 with range state.

Estimated state	Mean abs.	Standard deviation	Units
u	1.3356	1.6016	m/s
ϕ	2.8748	3.5462	deg
θ	2.0303	2.4876	deg

In order to compare the results from the different cases, the standard deviation of error is shown graphically for Flight #1 in Figure 13 and Flight #2 in Figure 14. It is shown in Figures 13 and 14 that the simplified formulation offers poorer estimation performance as expected, particularly for the attitude estimates. The addition of the range state does not affect the performance significantly.

5. Conclusions

This paper presented vision-aided inertial navigation techniques which do not rely upon GPS using UAV flight data. Two different formulations were presented, a full state estimation formulation which captures the aircraft ground velocity vector and attitude and a simplified formulation which assumes all of the aircraft velocity is in the forward direction. Both formulations were shown to be effective in regulating the INS drift. Additionally, a state was included in each formulation in order to estimate the distance between the image center and the aircraft. The full state formulation

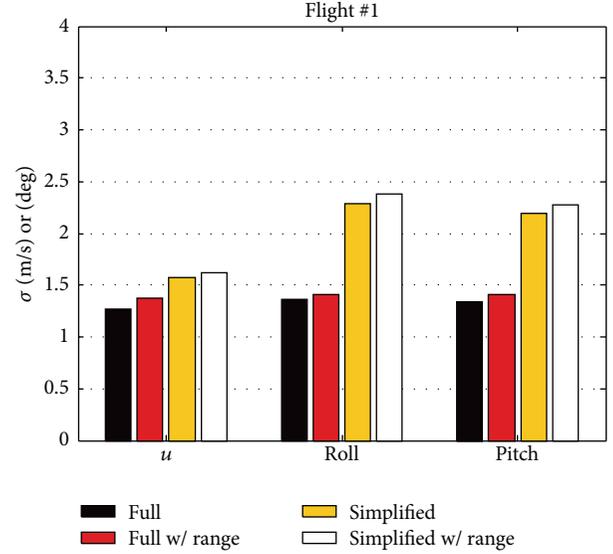


FIGURE 13: Graphical comparison of standard deviation of error for Flight #1.

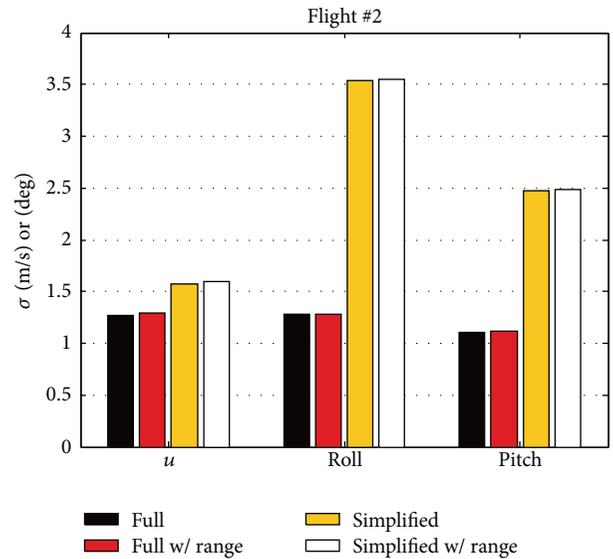


FIGURE 14: Graphical comparison of standard deviation of error for Flight #2.

was shown to be effective in estimating aircraft ground velocity to within 1.3 m/s and regulating attitude angles within 1.4 degrees standard deviation of error for both sets of flight data.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by NASA Grant no. NNX12AM56A, Kansas NASA EPSCoR PDG grant, and SRI grant.

References

- [1] C. Li, L. Shen, H.-B. Wang, and T. Lei, "The research on unmanned aerial vehicle remote sensing and its applications," in *Proceedings of the IEEE International Conference on Advanced Computer Control (ICACC '10)*, pp. 644–647, Shenyang, China, March 2010.
- [2] A. J. Calise and R. T. Rysdyk, "Nonlinear adaptive flight control using neural networks," *IEEE Control Systems Magazine*, vol. 18, no. 6, pp. 14–24, 1998.
- [3] M. S. Grewal, L. R. Weill, and A. P. Andrew, *Global Positioning, Inertial Navigation & Integration*, Wiley, New York, NY, USA, 2nd edition, 2007.
- [4] J. L. Crassidis, "Sigma-point Kalman filtering for integrated GPS and inertial navigation," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 1981–2004, San Francisco, Calif, USA, August 2005.
- [5] J. N. Gross, Y. Gu, M. B. Rhudy, S. Gururajan, and M. R. Napolitano, "Flight-test evaluation of sensor fusion algorithms for attitude estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2128–2139, 2012.
- [6] M. V. Srinivasan, "Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics," *Physiological Reviews*, vol. 91, no. 2, pp. 413–460, 2011.
- [7] P. S. Bhagavatula, C. Claudianos, M. R. Ibbotson, and M. V. Srinivasan, "Optic flow cues guide flight in birds," *Current Biology*, vol. 21, no. 21, pp. 1794–1799, 2011.
- [8] N. Franceschini, "Visual guidance based on optic flow: a biorobotic approach," *Journal of Physiology Paris*, vol. 98, no. 1-3, pp. 281–292, 2004.
- [9] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 519–535, 2007.
- [10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [11] M. V. Srinivasan, S. Thurrowgood, and D. Soccol, "Competent vision and navigation systems," *IEEE Robotics & Automation Magazine*, vol. 16, no. 3, pp. 59–71, 2009.
- [12] J. Conroy, G. Gremillion, B. Ranganathan, and J. S. Humbert, "Implementation of wide-field integration of optic flow for autonomous quadrotor navigation," *Autonomous Robots*, vol. 27, no. 3, pp. 189–198, 2009.
- [13] F. Kendoul, I. Fantoni, and K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Robotics and Autonomous Systems*, vol. 57, no. 6-7, pp. 591–602, 2009.
- [14] A. Beyeler, J.-C. Zufferey, and D. Floreano, "optiPilot: control of take-off and landing using optic flow," in *Proceedings of the European Micro Air Vehicle Conference and Competition (EMAV '09)*, Delft, The Netherlands, September 2009.
- [15] M. M. Veth and J. Raquet, "Fusion of low-cost imaging and inertial sensors for navigation," in *Proceedings of the 19th International Technical Meeting of the Satellite Division Institute of Navigation (ION GNSS '06)*, pp. 1093–1103, Fort Worth, Tex, USA, September 2006.
- [16] D. Dusha, W. Boles, and R. Walker, "Fixed-wing attitude estimation using computer vision based horizon detection," in *Proceedings of the International Unmanned Air Vehicle Systems Conference*, April 2007.
- [17] D. Dusha, W. Boles, and R. Walker, "Attitude estimation for a fixed-wing aircraft using horizon detection and optical flow," in *Proceedings of the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA '07)*, pp. 485–492, IEEE, Glenelg, Australia, December 2007.
- [18] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, May 2012.
- [19] M. Dille, B. Grocholsky, and S. Singh, "Outdoor downward-facing optical flow odometry with commodity sensors," in *Field and Service Robotics: Results of the 7th International Conference*, vol. 62 of *Springer Tracts in Advanced Robotics*, pp. 183–193, 2010.
- [20] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery, "Augmenting inertial navigation with image-based motion estimation," in *Proceedings of the IEEE International Conference on Robotics & Automation*, pp. 4326–4333, Washington, DC, USA, May 2002.
- [21] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, "Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 3056–3063, Shanghai, China, May 2011.
- [22] P.-J. Bristeau, F. Callou, D. Vissière, and N. Petit, "The navigation and control technology inside the AR.Drone micro UAV," in *Proceedings of the 18th IFAC World Congress*, pp. 1477–1484, Milano, Italy, September 2011.
- [23] H. Chao, Y. Gu, and M. Napolitano, "A survey of optical flow techniques for robotics navigation applications," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 73, no. 1–4, pp. 361–372, 2014.
- [24] W. Ding, J. Wang, S. Han et al., "Adding optical flow into the GPS/INS integration for UAV navigation," in *Proceedings of the International Global Navigation Satellite Systems Society IGNSS Symposium*, Surfers Paradise, Australia, 2009.
- [25] H. Romero, S. Salazar, and R. Lozano, "Real-time stabilization of an eight-rotor UAV using optical flow," *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1752–1762, 2012.
- [26] V. Grabe, H. H. Bulthoff, and P. R. Giordano, "A comparison of scale estimation schemes for a quadrotor UAV based on optical flow and IMU measurements," in *Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 5193–5200, IEEE, Tokyo, Japan, November 2013.
- [27] A. Martinelli, "Vision and IMU data fusion: closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.
- [28] M. B. Rhudy, H. Chao, and Y. Gu, "Wide-field optical flow aided inertial navigation for unmanned aerial vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '14)*, pp. 674–679, Chicago, Ill, USA, September 2014.
- [29] H. Chao, Y. Gu, J. Gross, G. Guo, M. L. Fravolini, and M. R. Napolitano, "A comparative study of optical flow and traditional sensors in UAV navigation," in *Proceedings of the 1st American*

- Control Conference (ACC '13)*, pp. 3858–3863, Washington, DC, USA, June 2013.
- [30] R. C. Hibbeler, *Engineering Mechanics: Dynamics*, Prentice Hall, 10th edition, 2004.
- [31] V. Klein and E. A. Morelli, *Aircraft System Identification: Theory and Practice*, American Institute of Aeronautics and Astronautics, Reston, Va, USA, 2006.
- [32] J. Roskam, *Airplane Flight Dynamics and Automatic Flight Controls*, DARcorporation, Lawrence, Kan, USA, 2003.
- [33] J. N. Gross, Y. Gu, M. Rhudy, F. J. Barchesky, and M. R. Napolitano, “On-line modeling and calibration of low-cost navigation sensors,” in *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, pp. 298–311, Portland, Ore, USA, August 2011.
- [34] D. W. Allan, “Statistics of atomic frequency standards,” *Proceedings of the IEEE*, vol. 54, no. 2, pp. 221–230, 1966.
- [35] X. Zhiqiang and D. Gebre-Egziabher, “Modeling and bounding low cost inertial sensor errors,” in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS '08)*, pp. 1122–1132, IEEE, Monterey, Calif, USA, May 2008.
- [36] Z. Xing, *Over-bounding integrated INS/GNSS output errors [Ph.D. dissertation]*, The University of Minnesota, Minneapolis, Minn, USA, 2010.
- [37] F. L. Lewis and V. L. Syrmos, *Optimal Control*, Wiley, New York, NY, USA, 2nd edition, 1995.
- [38] M. Mammarella, G. Campa, M. L. Fravolini, Y. Gu, B. Seanor, and M. R. Napolitano, “A comparison of optical flow algorithms for real time aircraft guidance and navigation,” in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, USA, August 2008.
- [39] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*, CRC Press, Washington, DC, USA, 1998.
- [40] T. Vercauteren and X. Wang, “Decentralized sigma-point information filters for target tracking in collaborative sensor networks,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, part 2, pp. 2997–3009, 2005.
- [41] D.-J. Lee, “Unscented information filtering for distributed estimation and multiple sensor fusion,” in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, USA, 2008.
- [42] M. Rhudy and Y. Gu, “Understanding nonlinear Kalman filters,” in *Interactive Robotics Letters*, West Virginia University, 2013, <http://www2.statler.wvu.edu/~irl/page13.html>.
- [43] M. Rhudy, J. Gross, Y. Gu, and M. R. Napolitano, “Fusion of GPS and redundant IMU data for attitude estimation,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Minneapolis, Minn, USA, August 2012.

Research Article

Inverse Kinematic Analysis and Evaluation of a Robot for Nondestructive Testing Application

Zongxing Lu,¹ Chunguang Xu,¹ Qinxue Pan,¹ Xinyu Zhao,² and Xinliang Li¹

¹*School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China*

²*School of Materials Science and Engineering, Dalian Jiaotong University, Dalian 116028, China*

Correspondence should be addressed to Qinxue Pan; panqx@bit.edu.cn

Received 29 December 2014; Revised 13 April 2015; Accepted 16 April 2015

Academic Editor: Nan Xiao

Copyright © 2015 Zongxing Lu 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 robot system has been utilized in the nondestructive testing field in recent years. However, only a few studies have focused on the application of ultrasonic testing for complex work pieces with the robot system. The inverse kinematics problem of the 6-DOF robot should be resolved before the ultrasonic testing task. A new effective solution for curved-surface scanning with a 6-DOF robot system is proposed in this study. A new arm-wrist separateness method is adopted to solve the inverse problem of the robot system. Eight solutions of the joint angles can be acquired with the proposed inverse kinematics method. The shortest distance rule is adopted to optimize the inverse kinematics solutions. The best joint-angle solution is identified. Furthermore, a 3D-application software is developed to simulate ultrasonic trajectory planning for complex-shape work pieces with a 6-DOF robot. Finally, the validity of the scanning method is verified based on the C-scan results of a work piece with a curved surface. The developed robot ultrasonic testing system is validated. The proposed method provides an effective solution to this problem and would greatly benefit the development of industrial nondestructive testing.

1. Introduction

Ultrasonic testing as an important nondestructive inspection method is adopted in numerous applications to test the internal defects of composite materials with simple structures. However, the manufacturing technology of composite materials with a complex shape, such as a curved shape, variable thickness, and complex rotary structure, has developed rapidly. Thus, the automatic inspection of complex structures has become a challenge.

The joint robot with six degrees of freedom (DOF) can reach any position and orientation in its operating range; the inspection problem can be solved by combining this joint robot with ultrasonic testing technology [1]. In such an automatic ultrasonic testing system, optimization and simulation of the inspection path are important so that the robot that scans these complex structures can be operated according to reasonable tracks. Many researchers [2–5] have worked on the inverse kinematics of the general 6R serial robot. Robot simulation was based on robot kinematics. Yu

et al. [6] built a 6R serial robot using OpenGL and conducted motion simulation. Lee and Liang [7] provided a resultant elimination procedure by using complex number method and vector theory. However, the geometric interpretation of their elimination procedure was not completely revealed because of its complexity. Establishing a common inverse algorithm so that the analytical method [8, 9], geometric method [10, 11], and numerical method [12] are all developed to address this problem is difficult. Wang et al. [13] provided a preliminary inverse kinematics method for the Staubli robot. However, this method cannot explain the optimization of these solutions.

A new evolutionary arm-wrist separateness method is used in the present study to formulate the kinematics equations of the general 6-DOF robot. Inverse kinematics only has eight solutions; hence, the difficulty in solving is reduced. With the optimized method, the optimal solution for a trajectory can be identified. A 3D-application software is created to simulate ultrasonic trajectory planning. Based

on the feature that the subsequent three joint axes intersect at one point, the arm-wrist separateness method is adopted in Section 2 to solve the inverse kinematics of the robot. Additionally, the shortest distance rule is adopted in Section 3 to optimize the results [14]. A 3D-application software is also developed to simulate ultrasonic trajectory planning for complex-shape work pieces. Finally, the work piece with a curved surface is detected with this robot. The experiment results verify the effectiveness and feasibility of the proposed method. The simulation and experiment results are shown in Section 4, and the paper ends with concluding remarks in Section 5.

2. Inverse Kinematics Solution

Denavit and Hartenberg established the D-H method [15, 16], which is commonly utilized in robot kinematics models. The inverse kinematics solution uses the position and orientation $(p_x, p_y, p_z, \alpha, \beta, \gamma)$ of robot end-effector, which has been known to solve the joint angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$. An arm-wrist separateness method was used in this study.

In the TX90 XL robot, the axes of the last three joints intersect at one point, which is referred to as point A. The position of point A is independent of the last three joints $\theta_4, \theta_5, \theta_6$. Therefore, only the three previous joints should be considered when solving the position of point A. These joints are the main content of the arm-wrist separateness method. The position and orientation are $(p_x, p_y, p_z, \alpha, \beta, \gamma)$, and matrix T_0^6 can be achieved from the D-H model. The position of A is denoted as $p_a = [p_{ax}, p_{ay}, p_{az}]'$.

The position of point A can be described as

$$\begin{aligned} p_{ax} &= p_x - d_6 a_x, \\ p_{ay} &= p_y - d_6 a_y, \\ p_{az} &= p_z - d_6 a_z. \end{aligned} \quad (1)$$

2.1. Solutions of Arm Joint Angles $\theta_1, \theta_2, \theta_3$. The position of point A can be determined from the homogeneous transformation matrix, which is derived from $T_0^1, T_1^2, T_2^3, T_3^4$. Consider

$$T_0^4 = T_0^1 T_1^2 T_2^3 T_3^4 = \begin{bmatrix} c_1 c_{23} c_4 - s_1 s_4 & -c_1 s_{23} & -c_1 c_{23} s_4 - s_1 c_4 & d_4 c_1 s_{23} - d_3 s_1 + a_1 c_1 + a_2 c_1 c_2 \\ s_1 c_{23} c_4 + c_1 s_4 & -s_1 s_{23} & -s_1 c_{23} s_4 + c_1 c_4 & d_4 s_1 s_{23} + d_3 c_1 + a_1 s_1 + a_2 c_2 s_1 \\ -s_{23} c_4 & -c_{23} & s_{23} s_4 & d_4 c_{23} - a_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

In the expressions above,

$$\begin{aligned} c_i &= \cos \theta_i, \\ s_i &= \sin \theta_i, \\ c_{ij} &= \cos(\theta_i + \theta_j), \\ s_{ij} &= \sin(\theta_i + \theta_j). \end{aligned} \quad (3)$$

The elements of P_a can be drawn as follows:

$$p_{ax} = d_4 c_1 s_{23} - d_3 s_1 + a_1 c_1 + a_2 c_1 c_2, \quad (4)$$

$$p_{ay} = d_4 s_1 s_{23} + d_3 c_1 + a_1 s_1 + a_2 c_2 s_1, \quad (5)$$

$$p_{az} = d_4 c_{23} - a_2 s_2. \quad (6)$$

The following can be obtained by calculating (4) $\times (-s_1)$ + (5) $\times (-c_1)$. Consider

$$p_{ay} c_1 - p_{ax} s_1 = d_3. \quad (7)$$

The solution of θ_1 can be obtained through solving (7). Therefore,

$$\begin{aligned} \theta_1 &= A \tan 2(p_{ay}, p_{ax}) \\ &\quad - A \tan 2\left(d_3, \pm \sqrt{p_{ax}^2 + p_{ay}^2 - d_3^2}\right). \end{aligned} \quad (8)$$

The following can be obtained by calculating (4) $\times c_1$ + (5) $\times s_1$. Moreover,

$$p_{ax} c_1 + p_{ay} s_1 = d_4 s_{23} + a_1 + a_2 c_2. \quad (9)$$

In the TX90 XL robot, s_{23} can be obtained from (9) as follows:

$$s_{23} = \frac{(p_{ax} c_1 + p_{ay} s_1) - a_1 - a_2 c_2}{d_4}. \quad (10)$$

We can obtain c_{23} by solving (6) as follows:

$$c_{23} = \frac{p_{az} + a_2 s_2}{d_4}. \quad (11)$$

Substituting the equation $s_{23}^2 + c_{23}^2 = 1$ with (10) and (11) yields

$$\begin{aligned} &(p_{ax} c_1 + p_{ay} s_1 - a_1) c_2 - p_{az} s_2 \\ &= \frac{(p_{ax} c_1 + p_{ay} s_1 - a_1)^2 + a_2^2 + p_{az}^2 - d_4^2}{2a_2}. \end{aligned} \quad (12)$$

The solution of θ_2 is achieved by solving (12). Consider

$$\begin{aligned} \theta_2 = & A \tan 2 \left((p_{ax}c_1 + p_{ay}s_1 - a_1), p_{az} \right) \\ & - A \tan 2 \left(t_1, \right. \\ & \left. \pm \sqrt{(p_{ax}c_1 + p_{ay}s_1 - a_1)^2 + p_{az}^2 - t_1^2} \right), \end{aligned} \quad (13)$$

where

$$t_1 = \frac{(p_{ax}c_1 + p_{ay}s_1 - a_1)^2 + a_2^2 + p_{az}^2 - d_4^2}{2 \times a_2}. \quad (14)$$

The result of performing the calculation of (10)/(11) is

$$\tan(\theta_2 + \theta_3) = \frac{p_{ax}c_1 + p_{ay}s_1 - a_1 - a_2c_2}{p_{az} + a_2s_2}. \quad (15)$$

The solution of (15) is θ_3 , and we have

$$\begin{aligned} \theta_3 = & A \tan 2 (p_{ax}c_1 + p_{ay}s_1 - a_1 - a_2c_2, p_{az} + a_2s_2) \\ & - \theta_2. \end{aligned} \quad (16)$$

2.2. Solutions of Wrist Joint Angles $\theta_4, \theta_5, \theta_6$. The orientation of the robot is controlled by the rotation matrix, and the orientation of A is described by R_0^3 . The orientation of the tool end-effector is described by R_0^6 . The relationship between R_0^3 and R_0^6 is $R_0^6 = R_0^3R_3^6$. Matrix R_3^6 can be described as

$$R_3^6 = R_3^4R_4^5R_5^6 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 \\ -s_5c_6 & s_5s_6 & c_5 \end{bmatrix}. \quad (17)$$

θ_5 can then be calculated as follows:

$$\theta_5 = A \tan 2 \left(\pm \sqrt{a_{wx}^2 + a_{wy}^2}, a_{wz} \right). \quad (18)$$

When $\theta_5 \neq 0$,

$$\theta_4 = A \tan 2 = \left(\frac{a_{wy}}{s_5}, \frac{a_{wx}}{s_5} \right). \quad (19)$$

When $\theta_5 = 0$, the arms are at the singular position where link axes 4, 5, and 6 are collinear. This condition has only one motion form of the robot tool side, the orientation of which is calculated by the sum or difference of θ_4 and θ_6 . Oftentimes, the current value of θ_4 is used.

According to (17), when $\theta_5 \neq 0$,

$$\theta_6 = A \tan 2 \left(\frac{a_{wz}}{s_5}, -\frac{n_{wz}}{s_5} \right). \quad (20)$$

When $\theta_5 = 0$, the current value of θ_6 is also utilized because the case is similar to the value determination of θ_4 .

TABLE 1: D-H parameters of the TX90XL robot.

Number	α_i (°)	a_i (mm)	d_i (mm)	θ_i (°)	Limit joints (°)
1	-90	50	0	θ_1 (0)	-180~180
2	0	650	0	θ_2 (-90)	-130~148
3	90	0	50	θ_3 (90)	-145~145
4	-90	0	650	θ_4 (0)	-270~270
5	90	0	0	θ_5 (0)	-115~140
6	0	0	100	θ_6 (0)	-270~270

3. Optimization of the Inverse Kinematics Solution

The obtained equations, $\theta_1, \theta_3, \theta_5$, show two roots; thus, the robot has eight groups of inverse kinematics solutions that correspond to the same position and orientation. For the kinematics of the robot, which has numbers of solutions, a suitable algorithm is required to select a set of values as the inverse solution of a robot.

For the eight group solutions, several solutions are real solutions, but some may be imaginary solutions. In the traditional inverse kinematics calculation method, the result of double variables arctangent function is in the range of $(-\pi, \pi)$. If the path trajectory has a singular point, then the joint will have a jump of 180° . Achieving a smooth motion is difficult because a long time is spent to rotate the joint of the robot by 180° , and the entire movement process is slow. We presented a type of inverse optimization method according to the incomplete solution of the inverse function. Comparison of the results of the joint with the actual range of the corresponding joint variables showed that the joint results may be $\pm 2\pi$. All of the trigonometric functions were not changed. This condition prevents the singular point jump and maintains the continuity of the trajectory. The actual range of the joint is shown in Table 1. The optimized process is shown in Figure 1.

The shortest distance weighted calculation method was developed to select the most suitable solution for the next trajectory point. The effect of each joint moving the same angle to the entire robot is not the same. The effect of joint 1 move angle is more than the effect of joint 6. A six-weighted factor was set up to represent the joint rotation effect for the entire robot. The weighted factor is large, and the effect of the entire robot is significant, which is not the optimal result that can be used for the trajectory. The shortest distance weighted calculation method can be expressed as

$$U_n = \sum_{i=1}^6 x_i (\theta_{(i,n)} - \theta_{(i,n-1)}), \quad (21)$$

where U_n is the solution effect of the n th point in the trajectory. x_i is the weighted factor of the i th joint. $\theta_{(i,n)}$ is the i th joint angle of the n th trajectory point, and $\theta_{(i,n-1)}$ is the i th joint angle of the $(n-1)$ trajectory point. If U_n is small, then the effect of the solution on the robot is minimal, which is the optimal solution for the trajectory.

Based on the shortest distance principle, a set of joint angles closest to the current value of the joint angles were

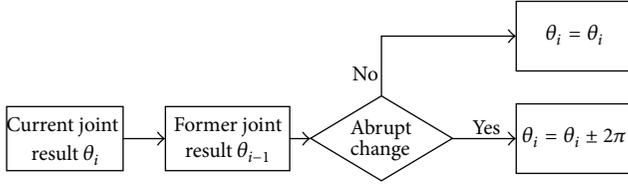


FIGURE 1: Optimized process of joints.

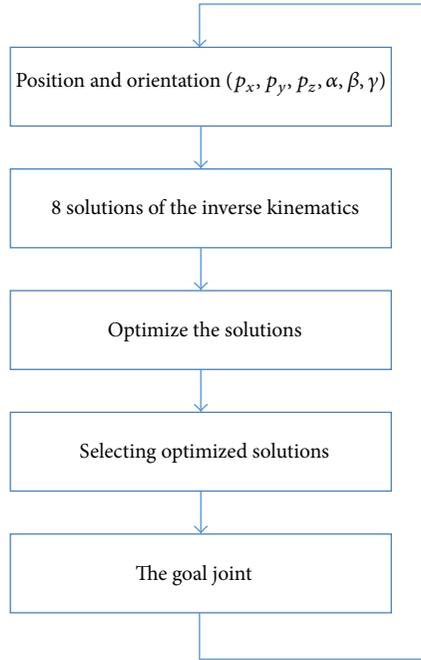


FIGURE 2: Optimization of the inverse kinematics solution.

selected. The process of the inverse kinematics solution optimization is shown in Figure 2.

4. Kinematics Simulation and Experimental Result

To validate the theoretical developments and conclusions, representative simulations were conducted in a laboratory. The test bed was composed of industrial robots produced by STAUBLI Faverges SCA, France. An external PC was used as a top layer controller to implement all the controller logics.

Driving the robot end-effector to an arbitrary point and recording the robot joints $\theta_1 = 56.31$, $\theta_2 = -3.24$, $\theta_3 = 105.09$, $\theta_4 = -1.08$, $\theta_5 = 62.93$, and $\theta_6 = -110.10$ form the controller panel. The coordinate position and orientation configurations for the robot are as follows:

$$\begin{aligned} p_x &= 334.59, \\ p_y &= 589.47, \\ p_z &= 419.04, \\ \alpha &= -167.76, \end{aligned}$$

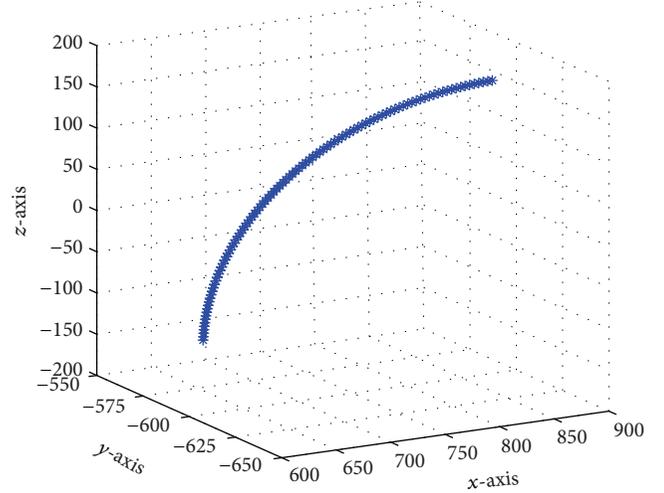


FIGURE 3: Trajectory of the robot in motion.

$$\begin{aligned} \beta &= 9.18, \\ \gamma &= 12.24. \end{aligned}$$

(22)

These positions and orientation configurations were substituted into the robot inverse kinematic equations. With the proposed inverse kinematics method, the joint angles were acquired with eight solutions, as shown in Table 2.

We consider the situation where the joint is in the trajectory. The only solution is the optimization joint. With the shortest distance rule, the eight solutions are compared with the joint angles of the current position and orientation. The proper joint angles can then be identified as $\theta_1 = 56.309$, $\theta_2 = -3.241$, $\theta_3 = 105.088$, $\theta_4 = -1.075$, $\theta_5 = 62.926$, and $\theta_6 = -110.098$. The results show that the solution obtained is fully correct and effective.

A set trajectory of the inspection task for the cylindrical work piece is shown in Figure 3. This trajectory is a section of a circular arc. When testing the cylindrical work piece, the transducer should be along the normal direction. Therefore, the orientations are toward the center of the circle. The x -axis is along the tangential direction of motion. The trajectory was generated from CATIA first and then optimized with MATLAB.

The trajectory was divided into 100 discrete points. Joint solutions can be solved with the inverse kinematics method proposed in Section 2. The motion trajectory of six joint angles can be obtained by using the 100 solutions. The joint positions for the robot are shown in Figure 4. The trajectory of joint 4 exhibits singularity. The trajectory has a 180° abrupt change.

The above trajectory was reoptimized with the optimized method proposed in this study. The new trajectory is shown in Figure 5. In the entire process, no large degree flop occurs. This finding proves that the inverse kinematics of the proposed method is feasible.

Based on the correctness of this algorithm, a 3D-application software was built with OpenGL technology to

TABLE 2: Solutions of inverse kinematics.

Number	θ_1 (°)	θ_2 (°)	θ_3 (°)	θ_4 (°)	θ_5 (°)	θ_6 (°)
1	56.309	101.847	-105.088	-4.597	167.973	-115.083
2	56.309	101.847	-105.088	-175.403	-167.973	64.9170
3	-114.885	-5.547	-96.119	176.365	63.462	-100.453
4	-114.885	-5.547	-96.119	-3.634	-63.462	79.546
5	56.309	-3.241	105.088	-1.075	62.926	-110.098
6	56.309	-3.241	105.088	178.925	-62.926	69.902
7	-114.885	-101.667	96.119	170.772	159.289	-110.72
8	-114.885	-101.667	96.119	-9.228	-159.289	69.278

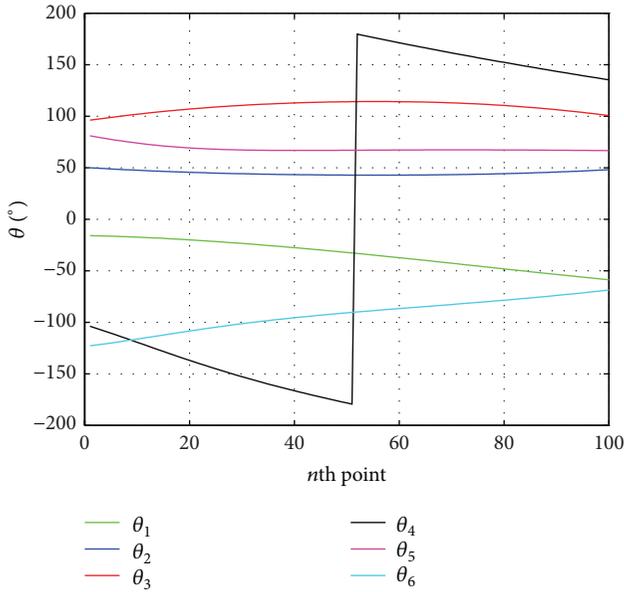


FIGURE 4: Joint position in the trajectory of the traditional method.

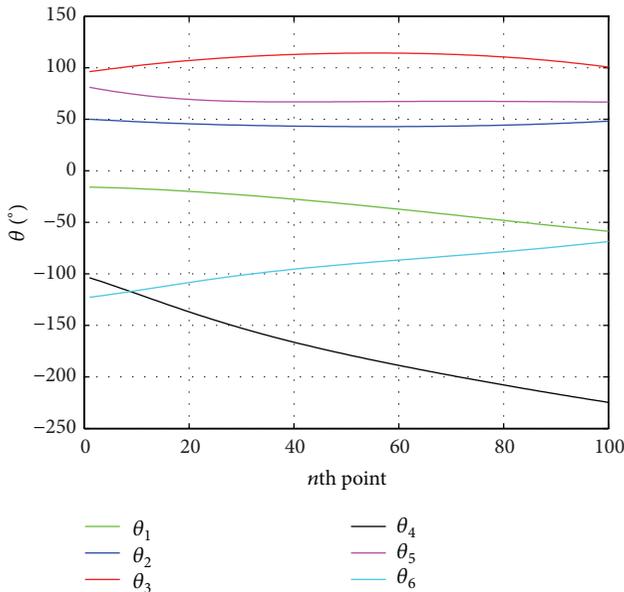


FIGURE 5: Joint position in the trajectory of the proposed method.

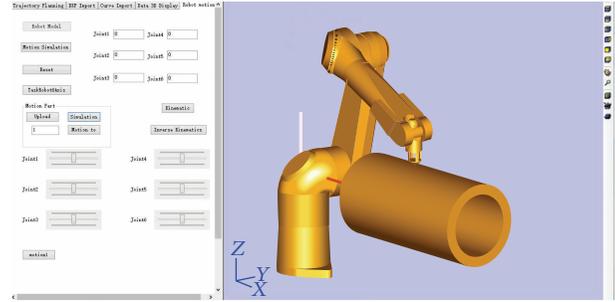


FIGURE 6: Kinematic simulation of the ultrasonic testing robot.



FIGURE 7: Robotic ultrasonic testing system.

simulate ultrasonic trajectory planning [17]. The 3D solid modeling in the OpenGL scene is shown in Figure 6. The scene displays the movement of the robot for the trajectory. It provides a good platform for further research and the verification problem of solving trajectory planning and error compensation.

The real ultrasonic testing experiment configuration is shown in Figure 7. The test work piece is a thick cylinder with standard artificial defects. A transducer is attached to the end-effector of the robot. A water ejection unit couples these transducers with the work piece surface. An ultrasonic testing system is responsible for data acquisition and processing. According to trajectory planning, the joint angles of the robot can be calculated.

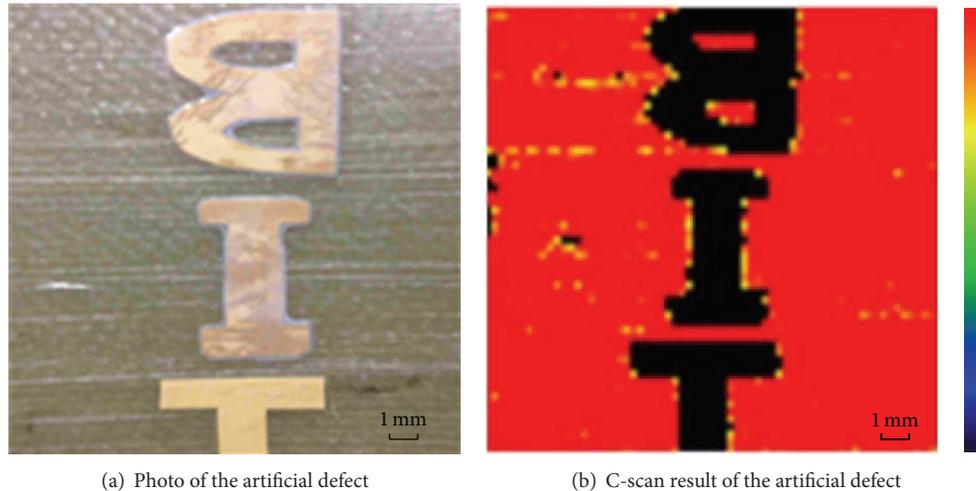


FIGURE 8: C-scan image of the ultrasonic testing robot.

Figure 8 shows an artificial defect with “BIT” shape manufactured in the curved composite work piece. The robot was controlled to scan this curved surface and simultaneously collect A-scan data at each given position. The data were utilized to create an image called C-scan image with data processing. The ultrasonic C-scan image was displayed on a PC. The effectiveness of the inspection path we provided is proven by the good agreement between the C-scan result and picture of the artificial defect.

5. Conclusions

A simple and effective method for the inverse kinematics problem of an industrial robot system was presented. The arm-wrist separateness method was adopted to solve the inverse kinematics for robots with 6 DOF. A large number of complicated matrix operations can be avoided, and the solution process is not time consuming.

With the aid of the shortest distance criteria, the orientation closest to the current position and orientation can be selected from eight solutions provided by the arm-wrist separateness method. These criteria can ensure the continuity of robot movement. This method is primarily considered from the continuity of angle motions. However, the principles of “disorder avoidance” and “best force” are also considered in several cases.

A 3D-application software was developed based on OpenGL technology. Simulation of ultrasonic trajectory planning for complex-shape work pieces was included in the software. A robotic ultrasonic testing system was established to test the complex-shape work pieces. The C-scan result is satisfactory and verifies the validity and effectiveness of the proposed method. The application of the inverse kinematics method will herald a new approach in the robot nondestructive testing industry. Meanwhile, complicated matrix operations greatly limit traditional inverse kinematics methods. The proposed method provides an effective solution to this problem and will greatly benefit the development of nondestructive industrial testing.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Program of International S&T Cooperation (Grant no. 2012DFA70260) and a major project of the National Science and Technology of China (Grant no. 2011ZX04014-081).

References

- [1] Z. Lu, C. Xu, X. Zhao, L. Zhang, H. Wang, and Q. Pan, “Ultrasonic transmission testing of twin-robots coordinated control,” in *Proceedings of the 10th IEEE International Conference on Mechatronics and Automation (ICMA '13)*, pp. 1256–1260, IEEE, Takamatsu, Japan, August 2013.
- [2] Y. Wei, S. Jian, S. He, and Z. Wang, “General approach for inverse kinematics of nR robots,” *Mechanism and Machine Theory*, vol. 75, pp. 97–106, 2014.
- [3] D. Manocha and J. F. Canny, “Efficient inverse kinematics for general 6R manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 648–657, 1994.
- [4] S. Z. Xin, L. Y. Feng, H. L. Bing, and Y. T. Li, “A simple method for inverse kinematic analysis of the general 6R serial robot,” *Journal of Mechanical Design, Transactions of the ASME*, vol. 129, no. 8, pp. 793–798, 2007.
- [5] M. L. Husty, M. Pfurner, and H.-P. Schröcker, “A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator,” *Mechanism and Machine Theory*, vol. 42, no. 1, pp. 66–81, 2007.
- [6] Y. Yu, P. Wang, and Q. Liao, “Inverse kinematics analysis or general 6r serial robot and motion simulation,” *China Mechanical Engineering*, vol. 14, no. 24, pp. 2130–2132, 2003.
- [7] H.-Y. Lee and C.-G. Liang, “Displacement analysis of the general spatial 7-link 7R mechanism,” *Mechanism and Machine Theory*, vol. 23, no. 3, pp. 219–226, 1988.

- [8] Z.-S. Ni, Q.-Z. Liao, S.-M. Wei, S.-G. Qiao, and R.-H. Li, "New algorithm for inverse kinematics analysis of general 6R serial robot," *Journal of Beijing University of Posts and Telecommunications*, vol. 32, no. 2, pp. 29–33, 2009.
- [9] P. Kalra, P. B. Mahapatra, and D. K. Aggarwal, "An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots," *Mechanism and Machine Theory*, vol. 41, no. 10, pp. 1213–1229, 2006.
- [10] J. Angeles, *Fundamentals of Robotic Mechanical Systems*, Springer, New York, NY, USA, 2002.
- [11] F. Yin, Y.-N. Wang, and S.-N. Wei, "Inverse kinematic solution for robot manipulator based on electromagnetism-like and modified DFP algorithms," *Acta Automatica Sinica*, vol. 37, no. 1, pp. 74–82, 2011.
- [12] X. P. Yang, K. Zhou, and Z. Ni, "A general inverse kinematic analysis of 6-Dof articulated manipulators," *Applied Mechanics and Materials*, vol. 532, pp. 275–279, 2014.
- [13] P. Wang, C. Wu, and X. Hu, "Inverse kinematics solution for Staubli robot manipulator," *Journal of Zhejiang Institute of Science and Technology*, vol. 22, no. 3, pp. 245–249, 2005.
- [14] Q.-J. Wang and J.-J. Du, "A new solution for inverse kinematics problems of MOTOMAN robot," *Journal of Harbin Institute of Technology*, vol. 42, no. 3, pp. 451–454, 2010.
- [15] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Journal of Applied Mechanics*, vol. 22, pp. 215–221, 1955.
- [16] J. Craig J, *Introduction to Robotics: Mechanics and Control*, Pearson Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 2005.
- [17] R. S. Wright and B. Lipchak, *OpenGL Superbible*, Sams, Indianapolis, Ind, USA, 2004.

Research Article

Simulation of Octopus Arm Based on Coupled CPGs

Juan Tian and Qiang Lu

College of Information and Engineering, Taishan Medical University, Taian 271016, China

Correspondence should be addressed to Juan Tian; jtian@tsmc.edu.cn

Received 9 January 2015; Revised 25 March 2015; Accepted 5 May 2015

Academic Editor: Nan Xiao

Copyright © 2015 J. Tian and Q. Lu. 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 octopus arm has attracted many researchers' interests and became a research hot spot because of its amazing features. Several dynamic models inspired by an octopus arm are presented to realize the structure with a large number of degrees of freedom. The octopus arm is made of a soft material introducing high-dimensionality, nonlinearity, and elasticity, which makes the octopus arm difficult to control. In this paper, three coupled central pattern generators (CPGs) are built and a 2-dimensional dynamic model of the octopus arm is presented to explore possible strategies of the octopus movement control. And the CPGs' signals treated as activation are added on the ventral, dorsal, and transversal sides, respectively. The effects of the octopus arm are discussed when the parameters of the CPGs are changed. Simulations show that the octopus arm movements are mainly determined by the shapes of three CPGs' phase diagrams. Therefore, some locomotion modes are supposed to be embedded in the neuromuscular system of the octopus arm. And the octopus arm movements can be achieved by modulating the parameters of the CPGs. The results are beneficial for researchers to understand the octopus movement further.

1. Introduction

Animals exploit soft structures to move effectively in complex natural environments and the typical one is octopus [1]. Octopus whose body and arms totally lack hard elements is unique marine invertebrate. And its advanced motor skills and intelligent behavior have attracted interest from both biologists and roboticists [2, 3]. Octopus arms have peculiar features such as the ability to bend in all directions, to achieve significant elongation, and to vary and control their stiffness [4].

Inspired by an octopus arm, the concept of continuum arms for use in robotic systems has been proposed and studied. Continuum arms have a large number of actuated degrees of freedom (DOF) and are therefore well suited for operations in highly constrained environments [5]. There have been several attempts to dynamically model continuum arms. Some dynamic models [5–8] inspired by the octopus arms are presented to explore possible strategies of movement control in the muscular hydrostat. However, it is difficult to control octopus arms with conventional techniques because of their high-dimensional body structures and their diverse body dynamics [9]. It is well known that the nervous system

of the octopus is highly distributed throughout the entire body. The octopus has a relatively small central brain which controls the large peripheral nervous system of the arms. A typical example showing the effectiveness of this distribution of the nervous system is the reaching behavior [10–13]. Sumbre et al. [13] showed that the arm extensions can be evoked in arms whose connection with the brain has been severed. Because the evoked motions in denervated octopus arms were identical to natural bend propagations, an underlying motor program appears to be embedded in the neuromuscular system of the arm, which does not require continuous central control. And the researchers suggested that the major part of the voluntary movement is controlled by a pattern generator that is confined to the arm's neuromuscular system.

The central pattern generator (CPG) is neural circuit found in both invertebrate and vertebrate animals that can produce rhythmic patterns of neural activity without receiving rhythmic inputs [14–16]. The famous one is the Matsuoka model [17, 18]. The CPG presents several interesting properties including distributed control, the ability to deal with redundancies, fast control loops, and allowing modulation of locomotion by simple control signals.

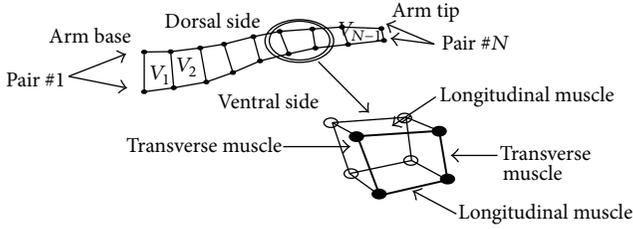


FIGURE 1: The octopus arm model.

In this paper, a model which includes three coupled central pattern generators (CPGs) is established and a 2-dimensional dynamic model [10, 19] of the octopus arm is used to explore possible strategies of movement control. And the CPG signals treated as activation are added on the ventral, dorsal, and transversal sides, respectively. The effects of the octopus arm are discussed using simulation method [20] when the parameters of the CPG are changed.

This paper is organized as follows. In Section 2, the 2-dimensional dynamic model and the CPG model are presented. Simulation is shown in Section 3. The conclusions and future works are made in Section 4.

2. The Octopus Arm Model and the CPG Model

2.1. Structure of the Octopus Model. In this paper, the octopus arm model is a 2-dimensional dynamic model of a soft robotic arm [10, 19], utilizing only masses and springs for its dynamic characteristics. The arm is divided into $(N - 1)$ rectangular segments and each one is defined by four vertices. For simplicity, the muscles are deprived of their mass and the entire arm's mass content is concentrated in point masses. The point masses are located in the four vertices of each segment, giving a total of $2N$ masses. The idealized massless springs function as muscles and connect all the adjacent point mass pairs of the model. The $2N$ masses are arranged in N pairs, each consisting of one ventral and one dorsal mass. $(N - 1)$ ventral and $(N - 1)$ dorsal longitudinal muscles connect the N ventral and N dorsal masses, respectively. In addition, a transverse muscle connects each ventral-dorsal pair. In this paper, there are 10 segments in this model. Figure 1 shows the general structure of the modeled arm.

As a model of a muscular hydrostat, this model relies on a basic assumption that the muscle tissue of the octopus arm is incompressible. From this assumption, it is evident that the arm's volume must be constant at all times. Due to the constant volume constraint, a contraction of a muscle reduces its length in one axis but must increase its length in at least one of the axes perpendicular to the first. Therefore, in the 2-dimensional model, shortening a segment in one direction will force it to elongate in the other. Using this simple physical mechanism, the octopus arm gains almost unconstrained motion and transfers force from one direction to another without needing a rigid skeleton.

2.2. Dynamics of the Octopus Model. The basic model is 2-dimensional, meaning that all the forces in its scope are vectors in an x - y plane. Thus, the motion of the arm is constrained to a plain as well. The model takes into account four types of forces acting on the arm. The first is internal forces generated by the arm's muscles (F^m). The second one is vertical forces caused by the combined influence of gravity and buoyancy (F^g). The third one is drag force produced by the arm's motion through the surrounding medium (F^w). The fourth is internal forces that maintain the constant volume constraint (F^c) [10].

The motion equations can be written as

$$M\ddot{q} = F^m + F^g + F^w + F^c, \quad (1)$$

where M is a diagonal mass matrix and \ddot{q} is the position vector.

A muscle is simulated by an ideal damped spring which exerts force caused by changes in its spring constant. The adjustments of the spring constant enable the user of the model to control the arm's movement. The arm's weight and the drag forces are calculated using the relevant physical theories. These forces are calculated from algebraic and differential manipulations on the equations of motion and the volume constraint.

There are two types of muscle models: one is the nonlinear muscle model and the other is linear damped spring model [10]. In this paper, every linear muscle in the linear model exerts the following force:

$$f(t) = [k_0 + k_{\max}a(t)] [l(t) - l_{\text{rest}}] + \alpha \frac{dl(t)}{dt}, \quad (2)$$

where l_{rest} is the rest length of the muscle. This was chosen as the largest length at which both active and passive forces are zero in real muscles. The linear damping coefficient α has dimensions of Ns/m. The passive spring constant of the muscle is expressed by k_0 and the maximal active spring constant of the muscle by k_{\max} , both having dimensions of N/m. $a(t)$ is a dimensionless activation function.

2.3. Implementation and Parameters of the Octopus Model. The model enables the user to activate the arm by changing the constants of the muscle simulating springs. Any number of spring constants can be changed simultaneously and a given set of different spring constants changes is dubbed activation. All the activations last a constant time. The simulation recalculates the coordinates and velocities for each simulation time interval and changes the activation for each activation time interval. All the parameters set either the physical environment of the arm or various technical features of the simulation. Table 1 summarizes some of the more influential parameters of the simulation.

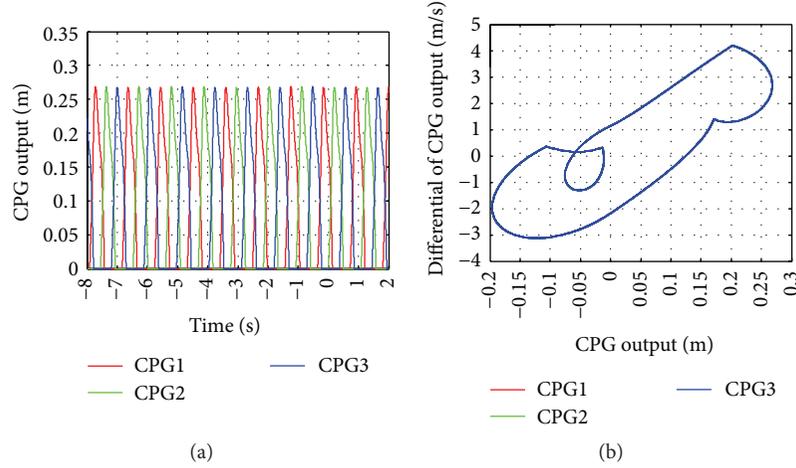


FIGURE 2: Outputs and phase diagrams of three CPGs: (a) the CPGs outputs and (b) the phase diagrams of these CPGs.

TABLE 1: Octopus arm simulation parameters.

Parameter	Implication
N	$(N - 1)$ is the number of segments constructing the arm
g	Gravitational acceleration [m/s^2]
water_sw	Specific weight of the medium (seawater, in this case) [kg/m^3]
arm_sw	Specific weight of the octopus arm [kg/m^3]
passive_elasticity	Ratio between maximal and minimal spring constants
sim_time	Simulation time [s]
delta_t	Time resolution for simulation [s]
act_interval	Interval between possible activation changes [s]
n_act_types	Number of available activations
slow_base	Force proportional to base speed, resisting its rotation
Activation	Declaration of an activation followed by a serial number, base rotation moment, and activation values for each spring

2.4. Model of Interaction between the CPG and the Octopus Arm. The coupled CPG model [21, 22] can be described by

$$\begin{aligned}
 T_r \dot{x}_1 + x_1 &= -dx_4 - wg(x_2) - wg(x_3) + e, \\
 T_a \dot{x}_4 + x_4 &= g(x_1), \\
 T_r \dot{x}_2 + x_2 &= -dx_5 - wg(x_1) - wg(x_3) + e, \\
 T_a \dot{x}_5 + x_5 &= g(x_2), \\
 T_r \dot{x}_3 + x_3 &= -dx_6 - wg(x_1) - wg(x_2) + e, \\
 T_a \dot{x}_6 + x_6 &= g(x_3),
 \end{aligned}$$

$$y_1 = g(x_1),$$

$$y_2 = g(x_2),$$

$$y_3 = g(x_3).$$

(3)

The function $g(\cdot)$ is a piecewise linear function defined by $g(x) = \max(0, x)$, which represents a threshold property of the neurons. These variables x_1 , x_2 , and x_3 represent the membrane potential. Self-inhibitory inputs x_4 , x_5 , and x_6 represent adaptation or fatigue property that ubiquitously exists in real neurons. The parameter e denotes the tonic input and determines the amplitude of CPG output. Parameters w and d represent the strength of mutual and self-inhibition, respectively; parameters T_r and T_a are the time constants that determine the reaction times of variables x_1 , x_2 , and x_3 and x_4 , x_5 , and x_6 . In other words, parameters T_r and T_a determine the frequency of CPG output. Three variables y_1 , y_2 , and y_3 represent the output of three CPGs, respectively.

The fundamental values of these CPG parameters are set as $T_r = 0.1$ s, $T_a = 0.2$ s, $d = 4.5$, $w = 4.5$, and $e = 1$. And the initial values are set as $[0.1 \ 0 \ 0 \ 0 \ 0 \ 0.1]$. Then the CPGs' output and phase diagrams are obtained, as shown in Figure 2.

In Figure 2(a), there is a uniform phase difference among three CPGs and the three phase diagrams have the same shape and they overlap each other. And the output of CPG approximates the overall shape of the electromyograms (EMG) activation measured in an octopus arm during reaching movements [23, 24]. EMG recorded from arm muscles showed that the octopus arm movement is associated with a propagating wave of muscle activation [24]. Therefore, each CPG output can be treated as the activation wave.

Yekutieli et al. [11] showed that the mechanism for bend propagation is a stiffening wave caused by muscle activation pattern. Therefore, it is reasonable that one-cycle outputs of three coupled CPGs treated as activation wave are added to the ventral, dorsal, and transversal sides of the octopus arm,

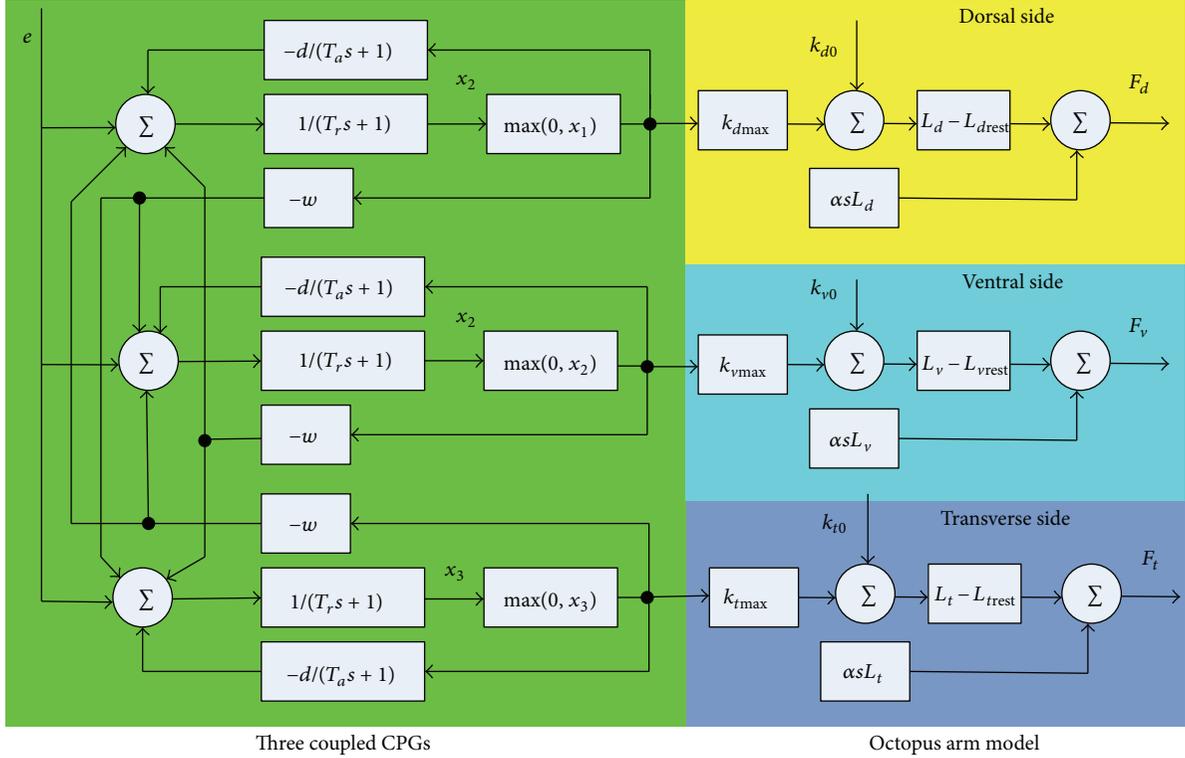


FIGURE 3: Model of interaction between the CPGs and the octopus arm.

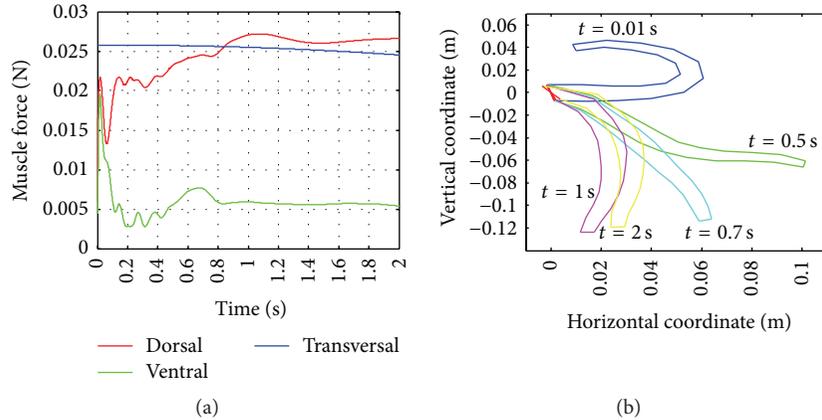


FIGURE 4: Muscles forces of the first octopus arm segment and a sequence of octopus arm movements: (a) muscles forces of the first segment and (b) a sequence of octopus arm movements.

respectively. And the model of interaction between the CPGs and the octopus arm is shown in Figure 3.

In Figure 3, the left part is the control block diagram of the three coupled CPGs [22], and the right one is the control block diagram of the octopus arm model by using Laplace transform. Parameters F_d , F_v , and F_t denote the force generated by the arm muscles of the ventral, dorsal, and transversal sides, respectively.

The octopus arm parameters are set as $g = 9.8 \text{ m/s}^2$, $\text{arm_sw} = 1200 \text{ kg/m}^3$, $\text{water_sw} = 1025 \text{ kg/m}^3$, $\text{muscle_strength} = 5000 \text{ N/m}^2$, $\text{passive_elasticity} = 0.03$, $\text{sim_time} = 2 \text{ s}$, delta_t

$= 0.001 \text{ s}$, $\text{slow_base} = 0.37$, $\text{act_interval} = 0.5 \text{ s}$, and $n_act_types = 5$. Then the arm muscles' forces are obtained, as shown in Figure 4(a). And a sequence of octopus arm movements is shown in Figure 4(b). In the sequence of octopus arm motion, the times are selected as 0.01 s, 0.5 s, 0.7 s, 1 s, and 2 s.

Because the muscles forces of other segments are similar to the ones of the first segment, the diagram of the first segment is shown only. In Figure 4(a), the forces generated by the dorsal and ventral sides are symmetrical and the force of the transversal side change little all the simulation time. Moreover, the movements in Figure 4(b) can mimic

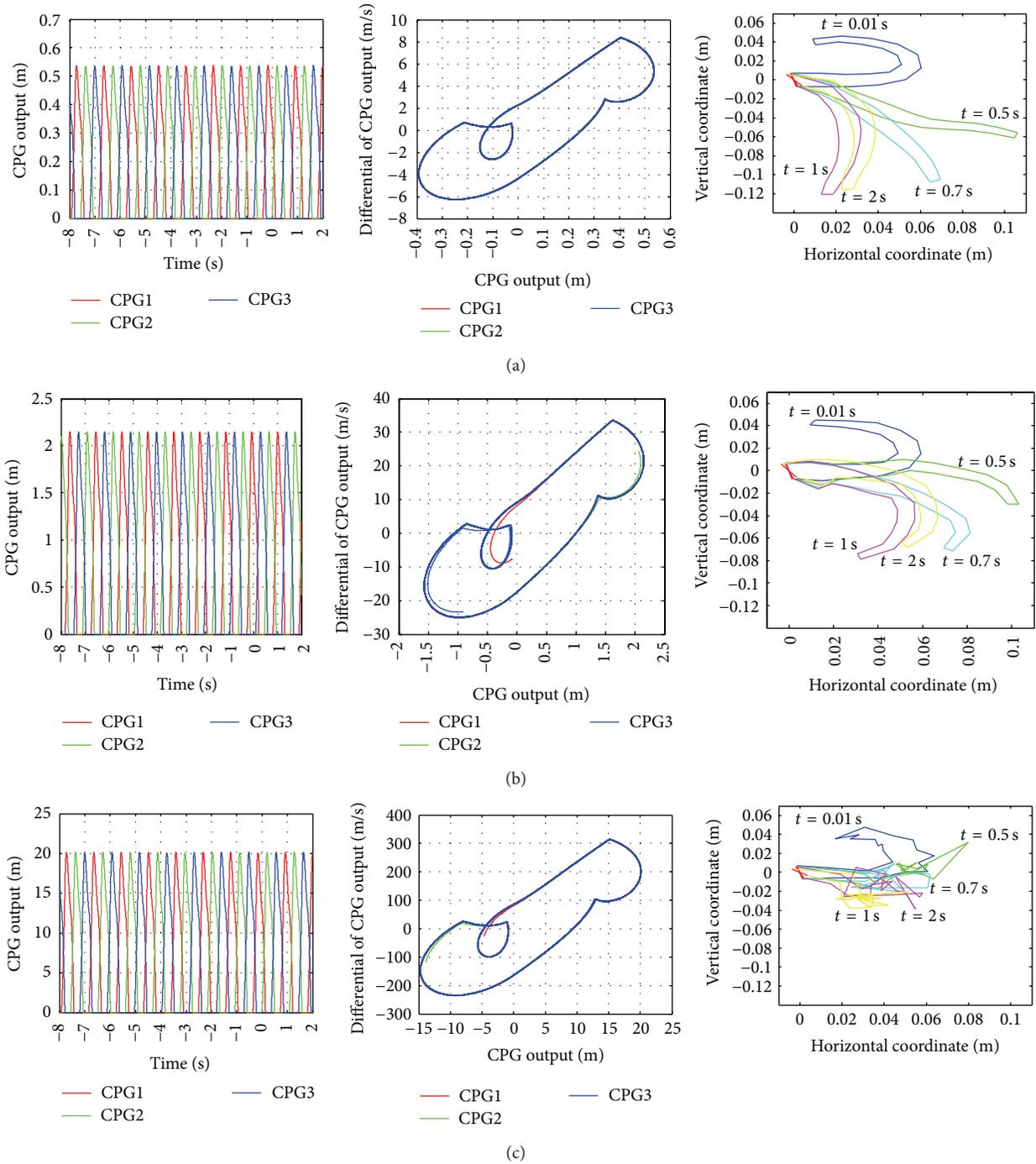


FIGURE 5: Outputs and phase diagrams of three coupled CPGs and a sequence of octopus arm movements with the parameter e (from left to right): (a) $e = 2$, (b) $e = 8$, and (c) $e = 75$.

the octopus movements. This simulation shows the mechanism for octopus movements: a stiffening wave caused by a symmetrical muscle activation pattern propagates along the arm and propels the octopus movements [11].

3. Simulation

Now the effects of octopus arm movements are discussed by simulation when the parameters of CPG are changed.

3.1. Effects of Octopus Arm Motion with Parameters e and T_r . In this simulation, the value of parameter e is varied in the interval $(0, 100]$ in step of 1. The outputs and phase diagrams of three coupled CPGs are obtained and a sequence of octopus arm movements is shown in Figure 5. When $e \in (0, 4)$, the shapes of three CPGs' phase diagrams are all the limit cycles and the simulation can mimic the movements of the octopus. The typical diagram is shown in Figure 5(a). While $e \geq 4$, there exists difference in the phase diagrams

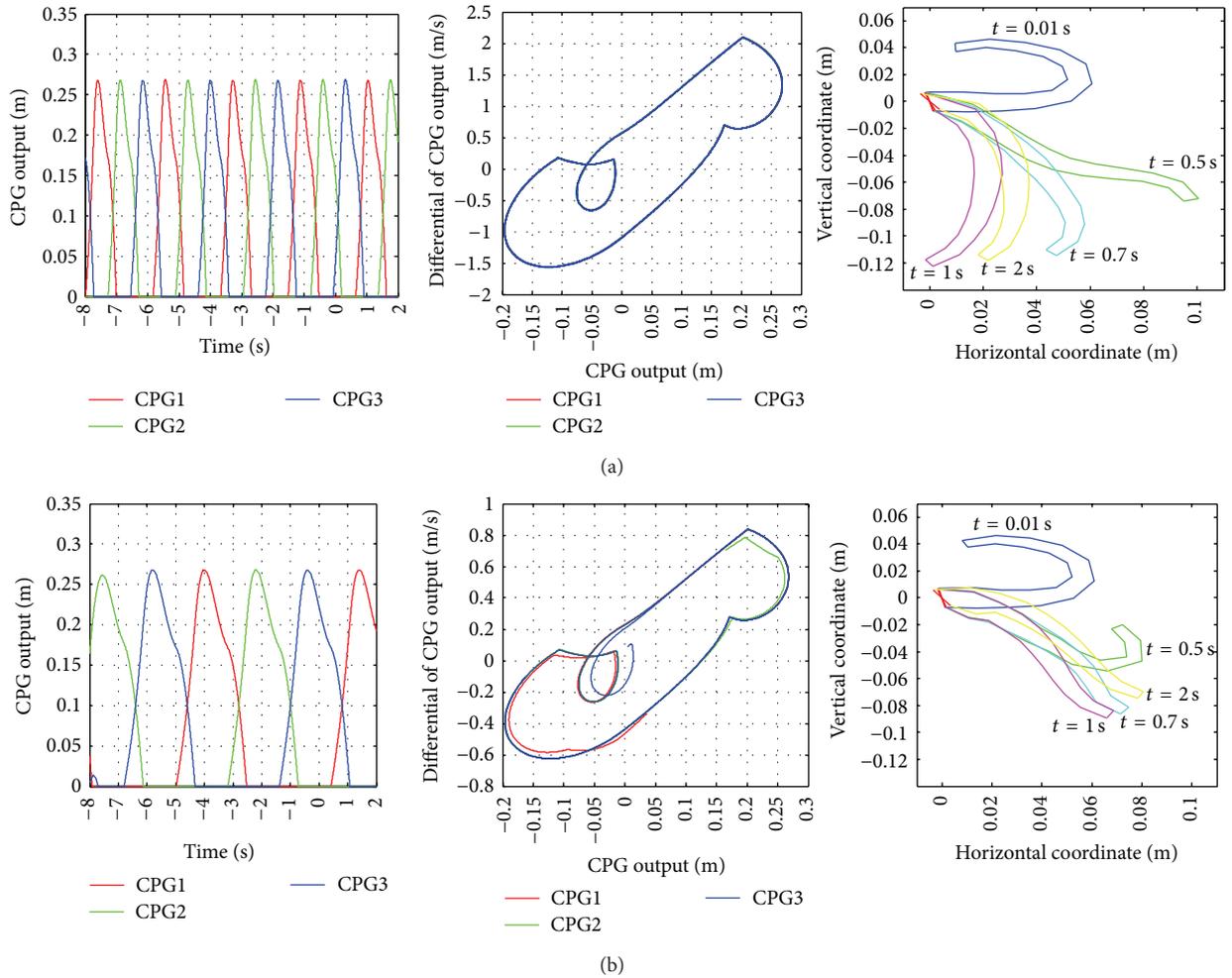


FIGURE 6: Outputs and phase diagrams of three coupled CPGs and a sequence of octopus arm movements with the parameter T_r (from left to right): (a) $T_r = 0.2$ and (b) $T_r = 0.5$.

of the three coupled CPGs and they do not overlap. The tip of octopus arm is not as straight as one in Figure 5(a). The typical diagram is shown in Figure 5(b). With the increase of the parameter e , the phase diagrams of the three CPGs become the limit cycles again. However, the degree of overlap decreases and the difference of the phase diagrams among three coupled CPGs becomes larger when $e \geq 53$. And the sequence of octopus arm movements is in disorder. The typical diagram is shown in Figure 5(c).

Here the effects of octopus arm are studied when the parameter T_r is changed and $e = 1$. The value of parameter T_r is varied in the interval $(0, 1]$ in step of 0.1. When $T_r \in (0, 0.2]$, the phase diagrams of the three CPGs are all the limit cycles and they overlap completely. The typical diagram is shown in Figure 6(a). While $T_r \geq 0.3$, there exists difference of the phase diagrams among the three coupled CPGs. And the movements of the octopus are changed. The typical diagram is shown in Figure 6(b). However, the difference of the phase diagrams among the three coupled CPGs becomes larger and the limit cycles are broken when $T_r = 1$. Then it cannot generate sustaining rhythmic motion.

3.2. Effects of Octopus Arm Motion with Parameters d and w . The effects of octopus arm are discussed when parameters d and w are changed, and other parameters are set as $e = 1$ and $T_r = 0.1$. The values of parameters d and w are varied in the interval $(0, 200]$ in step of 1. At the beginning, the phase diagrams of the CPGs are not the limit cycles and the movements cannot be continuous. When $d = w = 3$, the phase diagrams of the three CPGs are the limit cycles and the shapes are circles. But the three circles do not overlap each other. The movements of the octopus are different from the ones in Figure 6(a). The typical diagram is shown in Figure 7(a). When $d \geq 4$ and $w \geq 4$, the diagrams are similar to ones in Figure 6(a). The typical diagram is shown in Figure 7(b). With the increase of parameters d and w , the phase diagram of each CPG gradually changes to be a type of the limit cycle whose shape is different from the one in Figure 6(b) when $d \geq 12$ and $w \geq 12$. The typical diagram is shown in Figure 7(c).

The amplitude of CPG is proportional to the tonic input e , and the frequency of the limit cycle oscillation is proportional to $1/T_r$ [22]. From the above simulation, the amplitude

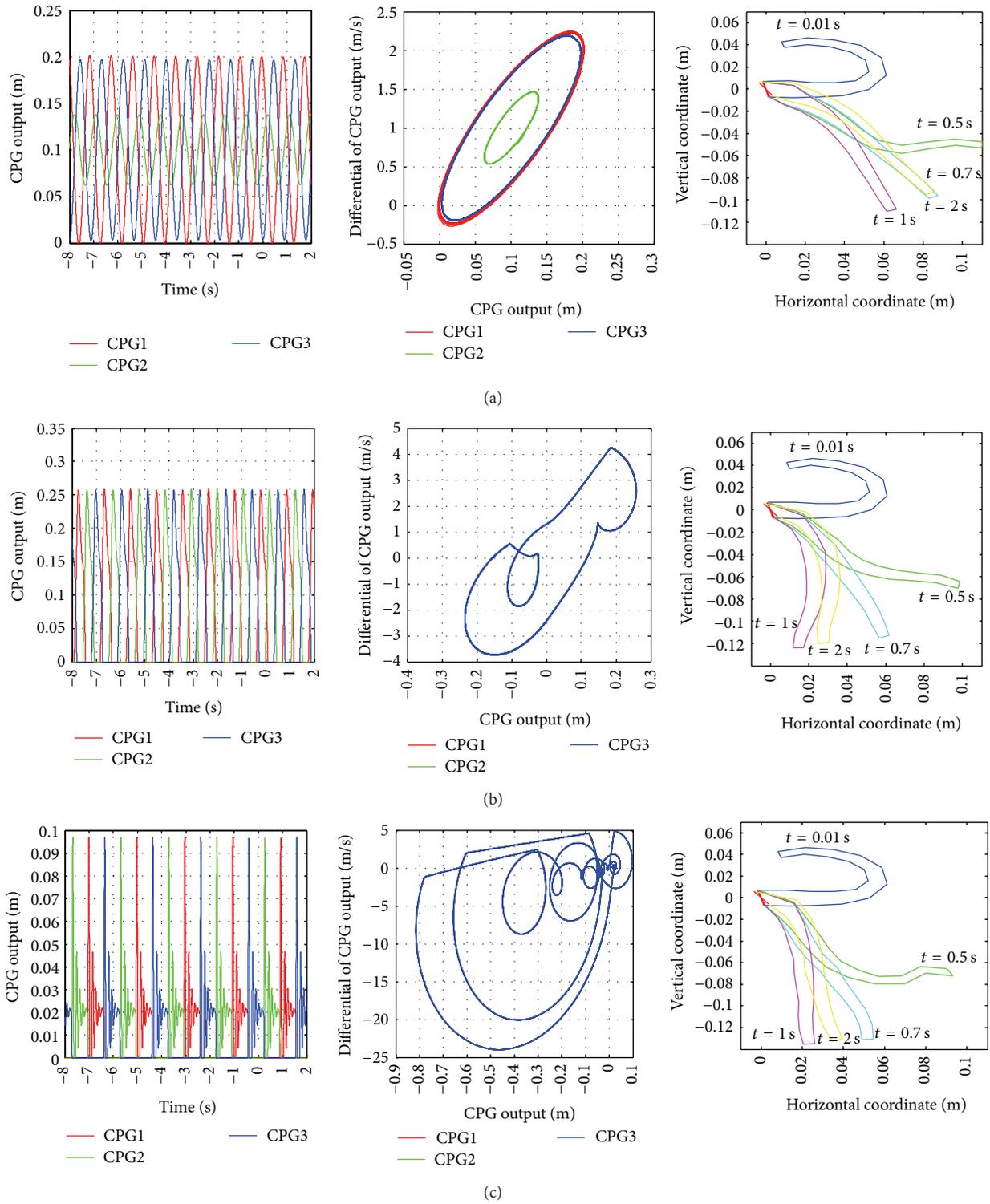


FIGURE 7: Outputs and phase diagrams of three coupled CPGs and a sequence of octopus arm movements with parameters d and w (from left to right): (a) $d = w = 3$, (b) $d = w = 5$, and (c) $d = w = 48$.

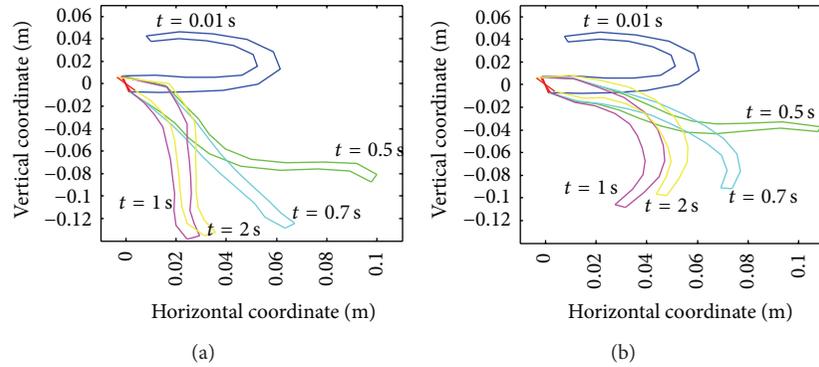


FIGURE 8: A sequence of octopus arm movements: (a) the third CPG for ventral side and the first CPG for transversal side and (b) the third CPG for dorsal side and the second CPG for transversal side.

and frequency should be in an allowable range. The larger amplitude leads the octopus arm to be disorder. The smaller frequency causes the limit cycle to be broken and the octopus movement cannot be sustaining. However, parameters d and ω also affect the shapes of the CPGs' phase diagrams which correspond to different octopus movements.

4. Discussion and Conclusion

In Figures 5–7, the octopus arm movements are determined by the shape and overlap degree of the three CPGs' phase diagrams. Different shapes of the phase diagrams correspond to different motion types, as shown in Figures 7(a)–7(c). Moreover, analogical shape of the phase diagram leads to similar movement of the octopus arm, as shown in Figures 5(a), 6(a), and 7(b)–7(c). Although the shapes of the three coupled CPGs are similar, the difference among them leads to different motion, as shown in Figures 5(a)–5(b) and 6(a)–6(b). The activation time also affects the octopus arm movement. The octopus arm locates in different position with different time.

The sequence of the three coupled CPGs also affects the octopus arm movements. Taking Figure 5(a) as an example, when changing the sequence of the three coupled CPGs, the octopus arm movements are shown in Figure 8. In Figure 8(a), the third CPG is for ventral side and the first CPG is for transversal one. In Figure 8(b), the third CPG is for dorsal side and the second CPG is for transversal one. Then the octopus movements are different from the ones in Figure 5(a).

Sumbre et al. [13] showed that the octopus reduces the complexity of controlling the flexible appendage by using highly stereotypical movements and there appears to be an underlying motor program embedded in the neuromuscular system of the arm. From simulations above, some locomotion modes are suggested to be embedded in the neuromuscular system of the arm. And the octopus arm movements can be achieved by modulating the parameters of the CPGs. And the simulation results enhance and improve the conclusion in [13]. The results are beneficial for researchers to understand the octopus movement further.

The octopus arm can apply force with the sole use of muscles without any rigid skeletal support. The biomechanical attributes of such an arm enable it to perform tasks no skeletal arm can perform. Hence, a robotic implementation of an octopus arm with a real-time learning control mechanism will yield a highly versatile application. And it is the direction of the future works.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work is supported by Project of Shandong Province Higher Educational Science and Technology Program, China (Grant no. J13LN04), and Taian Science and Technology Development Program, China (Grant no. 201430774).

References

- [1] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," *Trends in Biotechnology*, vol. 31, no. 5, pp. 287–294, 2013.
- [2] C. Laschi, M. Cianchetti, B. Mazzolai, L. Margheri, M. Follador, and P. Dario, "Soft robot arm inspired by the octopus," *Advanced Robotics*, vol. 26, no. 7, pp. 709–727, 2012.
- [3] L. Margheri, C. Laschi, and B. Mazzolai, "Soft robotic arm inspired by the octopus: I. From biological functions to artificial requirements," *Bioinspiration & Biomimetics*, vol. 7, no. 2, Article ID 025004, 2012.
- [4] W. M. Kier and M. P. Stella, "The arrangement and function of octopus arm musculature and connective tissue," *Journal of Morphology*, vol. 268, no. 10, pp. 831–843, 2007.
- [5] R. Kang, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Dynamic modeling and control of an octopus inspired multiple continuum arm robot," *Computers and Mathematics with Applications*, vol. 64, no. 5, pp. 1004–1016, 2012.
- [6] T. Zheng, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "A 3D dynamic model for continuum robots inspired by an

- octopus arm,” in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 3652–3657, IEEE, Shanghai, China, May 2011.
- [7] T. Zheng, D. T. Branson, R. Kang et al., “Dynamic continuum arm model for use with underwater robotic manipulators inspired by octopus vulgaris,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5289–5294, Saint Paul, Minn, USA, 2012.
- [8] F. Renda, M. Cianchetti, M. Giorelli, A. Arienti, and C. Laschi, “A 3D steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm,” *Bioinspiration and Biomimetics*, vol. 7, no. 2, Article ID 025006, 2012.
- [9] K. Nakajima, H. Hauser, R. Kang, E. Guglielmino, D. G. Caldwell, and R. Pfeifer, “A soft body as a reservoir: case studies in a dynamic model of octopus-inspired soft robotic arm (11916 words),” *Frontiers in Computational Neuroscience*, vol. 7, article 91, 2013.
- [10] Y. Yekutieli, R. Sagiv-Zohar, R. Aharonov, Y. Engel, B. Hochner, and T. Flash, “Dynamic model of the octopus arm. I. Biomechanics of the octopus reaching movement,” *Journal of Neurophysiology*, vol. 94, no. 2, pp. 1443–1458, 2005.
- [11] Y. Yekutieli, R. Sagiv-Zohar, B. Hochner, and T. Flash, “Dynamic model of the octopus arm. II. Control of reaching movements,” *Journal of Neurophysiology*, vol. 94, no. 2, pp. 1459–1468, 2005.
- [12] Y. Gutfreund, “Patterns of arm muscle activation involved in octopus reaching movements,” *Journal of Neuroscience*, vol. 18, no. 15, pp. 5976–5987, 1998.
- [13] G. Sumbre, Y. Gutfreund, G. Fiorito, T. Flash, and B. Hochner, “Control of octopus arm extension by a peripheral motor program,” *Science*, vol. 293, no. 5536, pp. 1845–1848, 2001.
- [14] A. J. Ijspeert, “Central pattern generators for locomotion control in animals and robots: a review,” *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.
- [15] R. M. Harris-Warrick, “Neuromodulation and flexibility in central pattern generator networks,” *Current Opinion in Neurobiology*, vol. 21, no. 5, pp. 685–692, 2011.
- [16] C. J. Liu, D. W. Wang, and Q. J. Chen, “Locomotion control of quadruped robots based on workspace trajectory modulations,” *International Journal of Robotics and Automation*, vol. 27, no. 4, pp. 345–354, 2012.
- [17] K. Matsuoka, “Sustained oscillations generated by mutually inhibiting neurons with adaptation,” *Biological Cybernetics*, vol. 52, no. 6, pp. 367–376, 1985.
- [18] K. Matsuoka, “Mechanisms of frequency and pattern control in the neural rhythm generators,” *Biological Cybernetics*, vol. 56, no. 5-6, pp. 345–353, 1987.
- [19] P. Wawrzyński and A. K. Tanwani, “Autonomous reinforcement learning with experience replay,” *Neural Networks*, vol. 41, pp. 156–167, 2013.
- [20] T. D. Frank, T. D. Gifford, and S. Chiangga, “Minimalistic model for navigation of mobile robots around obstacles based on complex-number calculus and inspired by human navigation behavior,” *Mathematics and Computers in Simulation*, vol. 97, pp. 108–122, 2014.
- [21] Q. Lu and J. Tian, “Synchronization and stochastic resonance of the small-world neural network based on the CPG,” *Cognitive Neurodynamics*, vol. 8, pp. 217–226, 2014.
- [22] K. Matsuoka, “Analysis of a neural oscillator,” *Biological Cybernetics*, vol. 104, no. 4-5, pp. 297–304, 2011.
- [23] Y. Gutfreund, T. Flash, Y. Yarom, G. Fiorito, I. Segev, and B. Hochner, “Organization of octopus arm movements: a model system for studying the control of flexible arms,” *Journal of Neuroscience*, vol. 16, no. 22, pp. 7297–7307, 1996.
- [24] Y. Gutfreund, T. Flash, G. Fiorito, and et al, “Patterns of arm muscle activation involved in octopus reaching movements,” *Journal of Neuroscience*, vol. 18, no. 15, pp. 5976–5987, 1998.