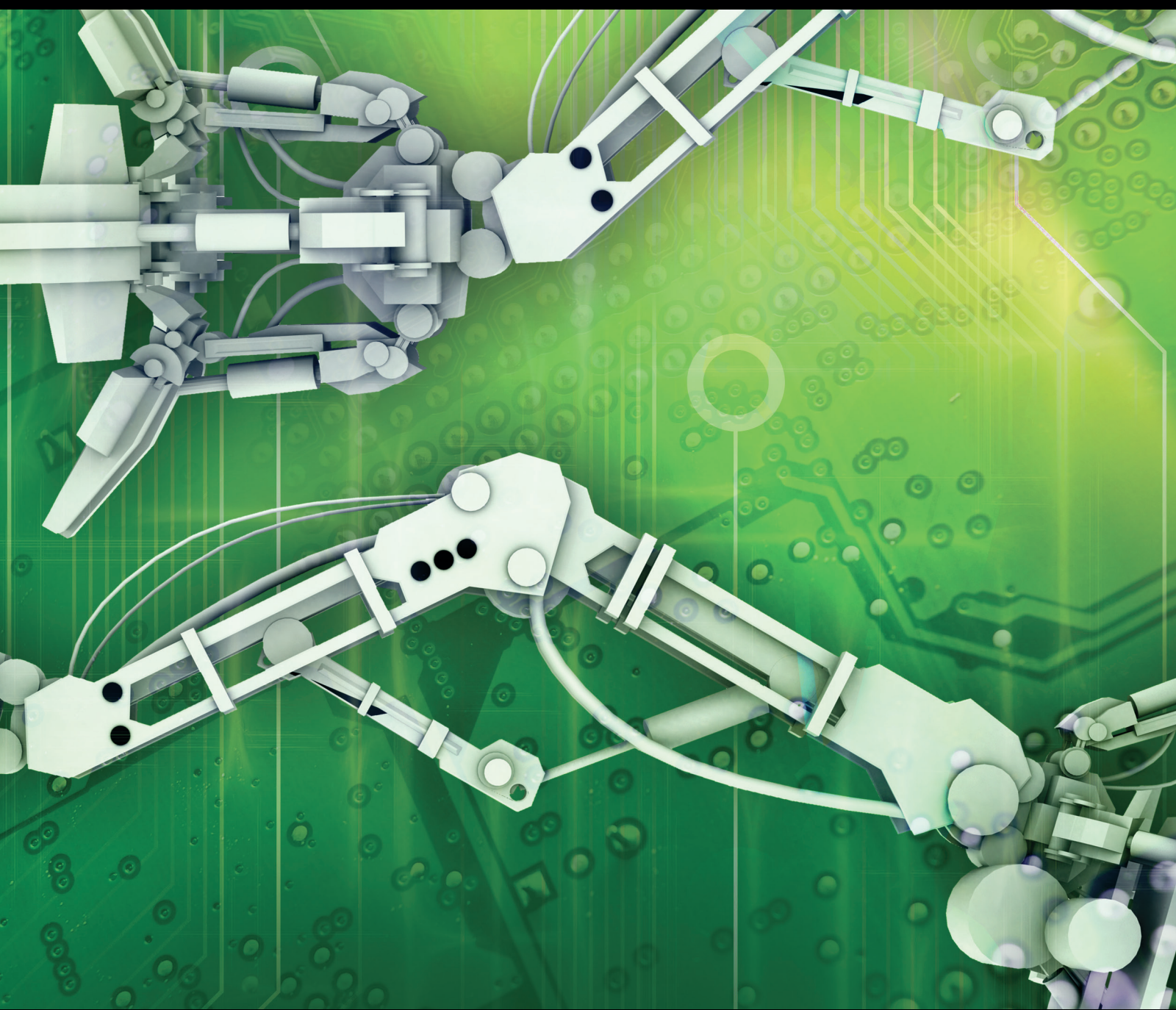
Journal of Robotics

Multiple Autonomous Robots Coordination and Navigation

Lead Guest Editor: Hsiung-Cheng Lin
Guest Editors: Cheng-Siong Lee and Ling-Ling Li



Multiple Autonomous Robots Coordination and Navigation

Journal of Robotics

Multiple Autonomous Robots Coordination and Navigation

Lead Guest Editor: Hsiung-Cheng Lin
Guest Editors: Cheng-Siong Lee and Ling-Ling Li

Copyright © 2019 Hindawi. 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

Raffaele Di Gregorio, Italy
Jorge Dias, Portugal
Meng Joo Er, Singapore
L. Fortuna, Italy
Andrew A. Goldenberg, Canada
Jose J. Guerrero, Spain
Farrokh Janabi-Sharifi, Canada
Yangmin Li, Hong Kong
Rene V. Mayorga, Canada
Giovanni Muscato, Italy
Shahram Payandeh, Canada
Gordon R. Pennock, USA
Oscar Reinoso, Spain
Tarek M. Sobh, USA
Keigo Watanabe, Japan
Simon X. Yang, Canada

Contents

Hindawi
Journal of Robotics
Volume 2019, Article ID 1274372, 2 pages
https://doi.org/10.1155/2019/1274372

Editorial

Multiple Autonomous Robots Coordination and Navigation

Hsiung-Cheng Lin,[1] **Ling-Ling Li**,[2] **and Vincent C. S. Lee**[3]

[1]*National Chin-Yi University of Technology, Taiwan*
[2]*Hebei University of Technology, China*
[3]*Monash University, Australia*

Correspondence should be addressed to Hsiung-Cheng Lin; hclin@ncut.edu.tw

Received 13 May 2019; Accepted 14 May 2019; Published 20 May 2019

Copyright © 2019 Hsiung-Cheng Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A team of autonomous robots working in parallel has a high potential to accomplish an assigned task faster than a single robot. To achieve this target, two fundamental challenges need to be addressed: The first challenge is to assign actions to robots under the coordination of the team so that the task can be finished more effectively. The second challenge is the environment mapping for robots navigation and path planning under unknown environments. As above, the dynamic task assignment of multirobot may be achieved using a self-organizing map based feature, reaching real-time collision-free robot path through sensor measurement in the environment. Therefore, the development of an algorithm to incorporate task assignment, path planning, and tracking control of a multirobot system is an indispensable mission for autonomous robots.

This special issue contains original research articles to address problems in both conventional and emerging human-robot interaction fields.

The paper "A Swarm Robotic Exploration Strategy Based on an Improved Random Walk Method" by B. Pang et al. presents an improved random-walk method in which each robot adjusts its step size adaptively to reduce the number of repeated searches. An environment can be searched far more efficiently if the appropriate search strategy is used. Because of the limited individual abilities of swarm robots, namely, local sensing and low processing power, random searching is the main search strategy used in swarm robotics. The random-walk methods that are used most commonly are Brownian motion and Lévy flight, both of which mimic the self-organized behavior of social insects. However, both methods are somewhat limited when applied to swarm robotics, where having the robots search repeatedly can result in highly inefficient searching. Therefore, by analyzing the characteristics of swarm robotic exploration, this paper proposes an improved RW method based on the density of robots in the environment. Each swarm robot adjusts its step size adaptively to arrive at other areas, and the proposed method distributes the robots uniformly in the environment to reduce the number of repeated searches.

The paper "Toward Dynamic Monitoring and Suppressing Uncertainty in Wildfire by Multiple Unmanned Air Vehicle System" by S. Rabinovich et al. presents an efficient response and persistent monitoring method for a wildfire. A crucial aspect is the ability to search for the boundaries of the wildfire by exploring a wide area. However, even as wildfires are increasing today, the number of available monitoring systems that can provide support is decreasing, creating an operational gap and slow response in such urgent situations. The objective of this work is to estimate a propagating boundary and create an autonomous system that works in real time. It proposes a coordination strategy with a new methodology for estimating the periphery of a propagating phenomenon using limited observations. The complete system design, tested on the high-fidelity simulation, demonstrates that steering the vehicles towards the highest perpendicular uncertainty generates the effective predictions. The results indicate that the new coordination scheme has a large beneficial impact on uncertainty suppression. This study thus suggests that an efficient solution for suppressing uncertainty in monitoring a wildfire is to use a fleet of

low-cost unmanned aerial vehicles that can be deployed quickly.

The paper "Optimal Skipping Rates: Training Agents with Fine-Grained Control Using Deep Reinforcement Learning" by A. Khan et al. presents the method for how the number of skip counts influences the learning process by employing convolutional deep neural networks (CDNN) with Q-learning and experience replay in a new game learning environment known as VizDoom. Game AI is one of the emerging, focused, and active research areas in artificial intelligence because computer games are the best test-beds for testing theoretical ideas in AI before practically applying them in real life world. Similarly, VizDoom is an artificial intelligence research platform based on Doom used for visual deep reinforcement learning in 3D game environments such as first-person shooters (FPS). While learning, the speed of the learning agent greatly depends on the number of frames the agent is permitted to skip. The agent is trained and tested on Doom's basic scenario(s) where the results are compared and found to be 10% better than the existing state-of-the-art research work on Doom based agents.

The paper "Human-Machine Interface for a Smart Wheelchair" by A. Hartman and V. K. Nandikolla presents the integration of hardware and software with sensor technology and computer processing to develop the next generation intelligent wheelchair. The focus is a computer cluster design to test high performance computing for smart wheelchair operation and human interaction. The LabVIEW cluster is developed for real-time autonomous path planning and sensor data processing. Four small form factor computers are connected over a Gigabit Ethernet local area network to form the computer cluster. Autonomous programs are distributed across the cluster for increased task parallelism to improve processing time performance.

The paper "Particle Filter and Finite Impulse Response Filter Fusion and Hector SLAM to Improve the Performance of Robot Positioning" by A. Bassiri et al. presents a hybrid (PF/FIR) algorithm for robot positioning in harsh environments, where there are more noise and sudden changes. This paper uses a hybrid filter algorithm for the indoor positioning system for robot navigation integrating Particle Filter (PF) algorithm and Finite Impulse Response (FIR) filter algorithm to ensure the continuity of the positioning solution. Additionally, the Hector Simultaneous Localisation and Mapping (Hector SLAM) algorithm is used to map the environment and improve the accuracy of the navigation.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this special issue.

Hsiung-Cheng Lin
Ling-Ling Li
Vincent C. S. Lee

Hindawi
Journal of Robotics
Volume 2019, Article ID 6914212, 9 pages
https://doi.org/10.1155/2019/6914212

Research Article

A Swarm Robotic Exploration Strategy Based on an Improved Random Walk Method

Bao Pang,[1] Yong Song,[2] Chengjin Zhang,[1,2] Hongling Wang,[1] and Runtao Yang[2]

[1]*School of Control Science and Engineering, Shandong University, Jinan 250061, China*
[2]*School of Mechanical, Electrical and Information Engineering, Shandong University at Weihai, Weihai 264209, China*

Correspondence should be addressed to Yong Song; songyong@sdu.edu.cn

Received 24 October 2018; Revised 30 January 2019; Accepted 26 February 2019; Published 13 March 2019

Academic Editor: Shahram Payandeh

Copyright © 2019 Bao Pang 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.

An environment can be searched far more efficiently if the appropriate search strategy is used. Because of the limited individual abilities of swarm robots, namely, local sensing and low processing power, random searching is the main search strategy used in swarm robotics. The random walk methods that are used most commonly are Brownian motion and Lévy flight, both of which mimic the self-organized behavior of social insects. However, both methods are somewhat limited when applied to swarm robotics, where having the robots search repeatedly can result in highly inefficient searching. Therefore, by analyzing the characteristics of swarm robotic exploration, this paper proposes an improved random walk method in which each robot adjusts its step size adaptively to reduce the number of repeated searches by estimating the density of robots in the environment. Simulation experiments and experiments with actual robots are conducted to study the effectiveness of the proposed method and evaluate its performance in an exploration mission. The experimental results presented in this paper show that an area is covered more efficiently using the proposed method than it is using either Brownian motion or Lévy flight.

1. Introduction

A complex problem in robotics and one that has received widespread attention is area exploration, which is used for various tasks including planetary exploration [1], search and rescue [2], foraging for food [3], and nanoscale drug delivery [4]. In area exploration, the core research issue is how to traverse an unknown area effectively. In a very large environment, it is relatively inefficient to have just one robot traverse the entire area. Instead, the exploration should be done using a multirobot approach, and swarm robots are used widely for this type of area exploration because of their robustness, flexibility, and scalability [5]. Most existing search methods depend on delicate systems of sensors (e.g., odometers and ultrasound radar) and sophisticated mapping algorithms [6, 7]. However, swarm robots, with their limited individual abilities (i.e., local sensing and low processing power), do not support complex localization and mapping, and instead they generally use a random walk (RW) as the area-exploration strategy. RWs can be divided into two categories, namely, (i) uncorrelated RWs, where the direction moved at each step is completely random, and (ii) correlated RWs, where there is a correlation between successive step orientations [8]. The main difference between the two types of RW is that in a correlated RW each step orientation is influenced by either the previous direction or the direction toward a given target. The RW methods studied herein refer mainly to uncorrelated RWs, the most commonly used being Brownian motion (BM) and Lévy flight (LF).

BM is the random motion of particles that are suspended in a fluid, and it results from the particles colliding with the fast-moving molecules of the fluid [9]. This pattern of motion involves a particle alternating its position randomly from one domain to another. BM is a continuous-time stochastic process and is usually described by the Wiener process. Because it is relatively easy to realize, BM has been used widely in robotics for random searching [10–12]. Each robot is regarded as a particle whose step size is normally distributed and each of whose movements is in an isotropically random direction. Wagner et al. [13] used robots with no sensory inputs to cover the gray area by means of BM; even though this method is not optimal, it has the advantages of (i)

requiring no sensors and (ii) being relatively inexpensive and tolerant. Furthermore, a novel RW method based on BM has been proposed to improve the area coverage ratio by allowing the motion of each robot to be influenced by landmarks installed in the environment [14].

An LF is an RW by which the walker can travel a large distance by taking many short steps and the occasional long step [15]. The step size has a power-law distribution, and a robot using LF is more likely to reach a remote area than is one that uses BM. Generally speaking, searching for a target and foraging can both be viewed as exploration missions. The foraging behaviors of many creatures in nature resemble LF, such as (i) the flight trajectories of albatrosses when foraging [16, 17], (ii) the intermittent foraging flight trajectories of fruit flies [18], and (iii) the flight trajectories of pelagic seabirds [19]. As Viswanathan et al. [20] noted, when the target sites are sparse and can be visited any number of times, an inverse-square power-law distribution of flight lengths, corresponding to LF, is an optimal strategy. When the target sites are abundant, simple Brownian motion is sufficiently efficient [21]. Fricke et al. evaluated the effectiveness of a LF search strategy and used a genetic algorithm to map the relationship between the search parameters and target configurations [22]. For the exploration missions of swarm robotics, many studies have used LF as the search strategy to improve the searching efficiency (SE). Fujisawa and Dobata showed that the LF search strategy maximized the SE of swarm robots by using pheromone to communicate with each other [23]. Schroeder et al. proposed a control law that combines a virtual pheromone and LF for efficient area coverage [24].

Although an exploration mission can be completed using either BM or LF, several deficiencies remain. BM is better for local searching and LF is better for global searching. For a balance between local searching and global searching, Deshpande et al. proposed a control law for efficient area coverage in a robot swarm by using a pheromone and by switching adaptively between BM and LF [25]. Sutantyo combined LF and an artificial potential field to improve the SE, with the potential field generating a repulsive force between pairs of robots, thereby dispersing neighboring robots [26]. Palmieri proposed using a weighted RW to find multiple paths among dynamical obstacles to improve the performance of robot navigation [27]. To explore the correlation between RW methods and the environment, Dimidov et al. used a swarm of Kilobots to search for a static target in different environments; the experimental results revealed which type of RW was best suited to each experimental scenario [28].

All the aforementioned methods achieved a certain degree of progress, but there still exist some problems. For one, BM and LF are used mainly for exploration missions conducted by a single robot. With swarm robots, having too many robots perform the exploration mission concurrently not only produces more instances of physical interference but also causes repeated searches, thereby reducing the SE markedly. The existing random search methods are therefore less efficient for swarm robotic exploration missions. Furthermore, the existing methods are based mostly on pheromones to simulate the foraging behavior of an ant colony, but such methods have considerable limitations in practical applications. Instead, what is proposed herein is an improved RW method in which each robot adjusts its step size adaptively by estimating the density of robots in the environment. The average time interval between two instances of physical interference is used to estimate the robot density; the shorter the time interval, the higher the density. When the robot density is relatively high, each robot searches a relatively small region with a relatively short step size; when the robot density is relatively low, each robot searches a relatively large region with a relatively long step size. To maintain a certain minimum distance between any two robots, one robot will turn directly away from another robot if obstacle avoidance occurs between them. Eventually, by adjusting their step sizes adaptively and by controlling their searching directions, the swarm robots become distributed evenly in the environment and each robot searches its own local area, thereby improving the SE.

The rest of this paper is organized as follows. Section 2 reviews the classical RW methods and Section 3 introduces the improved RW method. Section 4 reports on experiments that were conducted to assess the effectiveness of the improved RW method. Finally, Section 5 presents the conclusions and suggestions for future work.

2. Random Walk Methods

When no environmental information can be obtained, a random search is a basic search strategy for both animals and robots, especially for swarm robots that have limited individual abilities (i.e., local sensing and low processing power) and do not support more-complex search strategies. The commonly used RW methods are BM and LF.

2.1. Brownian Motion. BM describes the random motion of particles suspended in a fluid that is caused by the interactions between the particles. BM can be used to guide the random motion of robots and it has been widely used in robotics. In an exploration mission, a robot moves ahead by a given step size that is produced by the BM and then turns to a direction chosen randomly from the search space. In practice, BM can be viewed as a continuous-time stochastic process and can be described by the Wiener process. Mathematically, the Wiener process W_t is characterized by the following four properties:

(1) $W_0 = 0$;

(2) W_t is almost surely continuous;

(3) W_t has independent increments: if $0 \leq s_1 < t_1 \leq s_2 < t_2$, then $W_{t_1} - W_{s_1}$ and $W_{t_2} - W_{s_2}$ are independent random variables;

(4) W_t has Gaussian increments: $W_{t+u} - W_t \sim \mathcal{N}(0, u)$, where $\mathcal{N}(0, u)$ denotes the normal distribution with zero expectation and variance u.

In an RW by a single robot, the step size of the robot has a normal distribution with an expected value of zero and a variance of $u = 1$. To show how a single robot moves on an RW, a simulation experiment was performed in a fixed area.

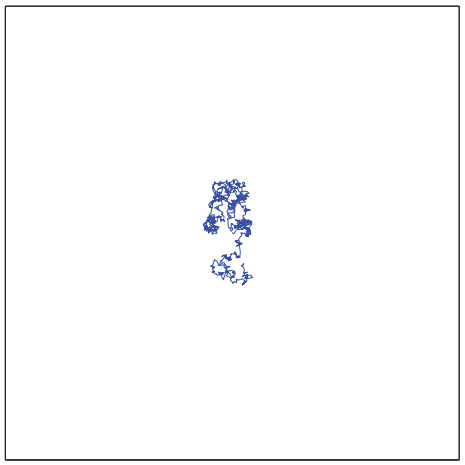


Figure 1: Random walk (RW) of one robot with Brownian motion (BM).

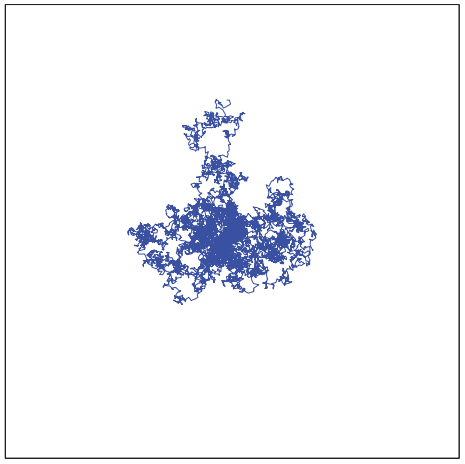


Figure 2: RWs of 10 robots with BM.

Figure 1 shows the trajectory of the robot over a given time; the robot tends to search around its original area, which is good for a local search but poor for a global search. Figure 2 shows the trajectories of 10 robots over a given time; with RWs, the robots produce too many repeated searches, which reduces the SE greatly.

2.2. Lévy Flight. An LF is an RW in which the step size has a heavy-tailed probability distribution that can be expressed as follows [20]:

$$P(s) = s^{-\lambda}, \tag{1}$$

where s is the step size with $1 < \lambda \leq 3$. LF generates a smaller step size with high frequency and occasionally a larger step size. In an exploration mission, this occasional larger step size allows the robot to reach the full range of the search space to complete a global search, whereas a robot with the smaller step size tends to complete a local search.

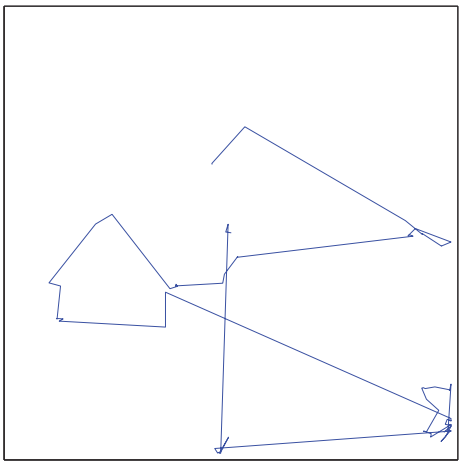


Figure 3: RW of one robot with Lévy flight (LF). The step size was generated according to (2) with $\beta = 1.5$.

This paper uses the method proposed by Mantegna [29] to calculate the LF step size, namely,

$$s = \frac{u}{|v|^{1/\beta}}, \tag{2}$$

where $\beta \in [0.3, 1.99]$; u and v are two normal stochastic variables with standard deviations σ_u and σ_v, respectively:

$$\begin{aligned} u &\sim N\left(0, \sigma_u^2\right), \\ v &\sim N\left(0, \sigma_v^2\right), \end{aligned} \tag{3}$$

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma\left[(1+\beta)/2\right] 2^{(\beta-1)/2}\beta} \right\}^{1/\beta}, \tag{4}$$

$$\sigma_v = 1,$$

where $\Gamma(z)$ is the gamma function.

Figure 3 shows that a robot using LF can better complete the exploration mission than one using BM. However, Figure 4 shows that having multiple robots performing the exploration mission concurrently results in many repeated searches. Therefore, although LF leads to better searching, some problems remain. In the experiments, the robots used infrared sensors to measure the proximity of objects up to 6 cm away. If the proximity between a robot and the boundary was less than 6 cm, then the robot turned to another direction to avoid the boundary; consequently, the robots never collided with the boundary.

2.3. Other Random Walk Methods. To improve the SE, some researchers have proposed other RW methods. For example, Sutantyo et al. proposed the combination of LF and an artificial potential field for multirobot explorations [26]. The LF generates the step size of the movement, while the artificial potential field improves the dispersion efficiency during deployment by generating a repulsive force between pairs of robots to disperse neighboring robots.

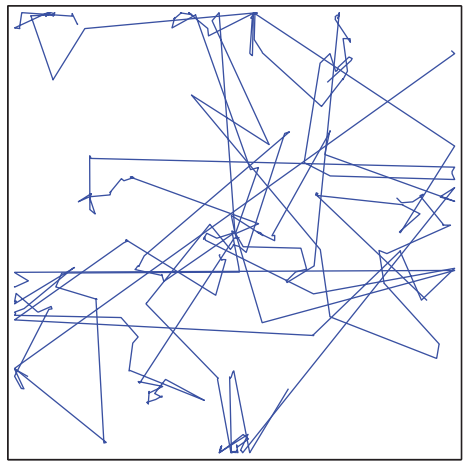


Figure 4: RWs of 10 robots with LF. The step size was generated according to (2) with $\beta = 1.5$.

Most other RWs are based on a virtual pheromone that serves as an indirect means of communication for swarm robots. When searching an environment, a robot deposits a pheromone that is detected by the other robots, along with its relative concentration. In an RW, a robot moves in an isotropically random direction. However, by using pheromone communication, a robot decides where to move based on the information conveyed by the pheromone, thereby not only improving the SE but also reducing the amount of physical interference between robots. Fujisawa et al. showed that the LF search strategy enhances the SE of swarm robots by using pheromone to communicate with other robots [23, 30].

However, the RW methods that involve an artificial potential field or a virtual pheromone are difficult to implement in practical applications. Consequently, only the basic RWs (i.e., BM and LF) are studied herein and are compared with the proposed improved RW.

3. Improved Random Walk Method

Because the existing RW methods have relatively low SE and are not particularly suited to swarm robotic exploration, the present paper proposes an improved RW method for completing such an exploration mission.

The best way to improve the SE is to reduce the number of repeated searches. These can be divided into two categories, namely, (i) those produced by the same robot (as in Figure 1) and (ii) those produced by other robots (as in Figures 2 and 4). Regarding category (i), because a robot with a small step size produces many repeated searches, the number of these can be reduced by increasing the step size. Indeed, Figure 3 shows that a robot with a large step size produces fewer repeated searches. Regarding category (ii), when many robots search an environment concurrently, BM and LF both produce many more repeated searches. Repeated searches are inevitable when two robots are close together, as can be seen in Figure 2; when the two robots are farther apart, the one with the larger step size can still produce many category (ii) repeated searches, as can be seen in Figure 4. To reduce the number of such repeated searches, the robots should therefore maintain a certain separation and the step size should be set to an appropriate value. In this way, if the swarm robots can be distributed evenly in the environment so that each robot searches only its local area, there will be fewer repeated searches.

Following this line of reasoning, an RW method is proposed herein based on the density of robots in the environment. When the robot density is high, each robot should search a small area with a small step size; when the robot density is low, each robot should search a large area with a large step size. To ensure that each robot occupies its own separate area, a robot that encounters another robot turns in the opposite direction when obstacle avoidance occurs between them. By adjusting the step size of each robot and controlling the direction of obstacle avoidance, the proposed method not only distributes the robots evenly in the environment but also causes them to search locally with an adaptive step size.

In an exploration mission, the area of the environment is usually unknown and the number of robots can change. Moreover, because of the limited sensory capabilities of swarm robots, the robot density cannot be calculated directly. However, the higher the robot density is, the more instances of physical interference (e.g., obstacle avoidance between robots) arise. Unfortunately, because the robots do not communicate with each other in the present case, an individual robot cannot assess the amount of physical interference, which is a global variable. Instead, the robot density is estimated from the average time interval ($\overline{t}$) between two instances of physical interference for a single robot; the smaller this average time interval, the higher the robot density.

In the proposed RW method, when a robot either moves forward by a given step size or encounters another robot, it calculates its step size as

$$S_t = \begin{cases} v * \overline{t} + k * S_{t-1}, & \Delta t \geq \overline{t} \\ v * \overline{t} - k * S_{t-1}, & \Delta t < \overline{t}, \end{cases} \tag{5}$$

where S_t is the step size with which the robot should move, S_{t-1} is the step size that the robot calculated last time, v is the speed at which the robot moves, and k ($0 < k < 1$) is an adjustment factor used to regulate the contribution of the previous step size S_{t-1}. The variable $\overline{t}$ is the average time interval between two instances of physical inference and is used to estimate the robot density in the entire search area; $\overline{t}$ changes with time and is updated when obstacle avoidance occurs. The variable Δt is the time between the current instance of physical inference and the previous one and is used to estimate the robot density in the local search area. When $\Delta t \geq \overline{t}$, the local robot density is lower than the global one; in this case, the local area contains fewer robots and therefore the robot in question should use a larger step size to search its area. When $\Delta t < \overline{t}$, the local robot density is higher than the global one; in this case, the local area contains

Table 1: Comparison of proposed RW method with other methods for different numbers of robots. Mean: average coverage ratio; Std.: standard deviation.

Methods	Brownian motion		Lévy flight		Our method	
	Mean	Std.	Mean	Std.	Mean	Std.
1	1.10%	9.31×10^{-4}	4.28%	4.52×10^{-3}	5.25%	2.63×10^{-3}
10	3.45%	3.47×10^{-3}	33.02%	1.11×10^{-2}	40.81%	1.10×10^{-2}
20	4.59%	2.91×10^{-3}	54.05%	1.35×10^{-2}	63.57%	1.13×10^{-2}
30	5.51%	4.17×10^{-3}	68.45%	1.07×10^{-2}	77.12%	1.59×10^{-2}

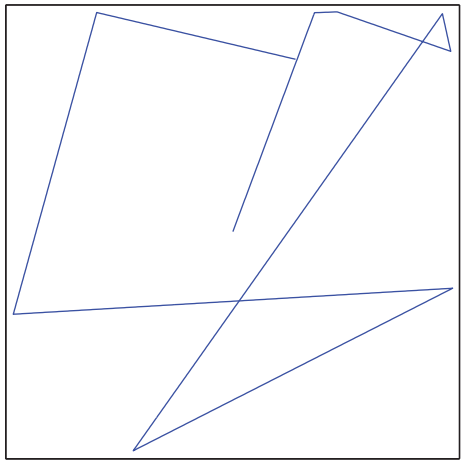


Figure 5: RW of one robot with the proposed method.

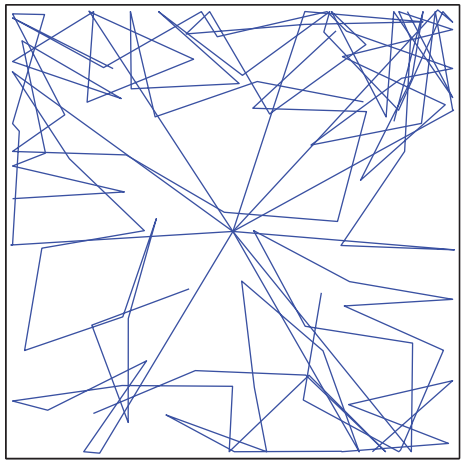


Figure 6: RWs of 10 robots with proposed method.

more robots and therefore the robot in question should use a smaller step size to search its area.

When there is only one robot, the proposed RW method becomes simply the linear search method. Figure 5 shows that with the linear search method, a robot makes fewer repeated searches. Figure 6 shows the trajectories of 10 robots performing the exploration mission.

Figure 7 shows that, by adjusting the step size adaptively, the proposed method distributes the robots evenly in the environment. As the exploration mission proceeds, the step sizes of the robots gradually converge. Moreover, because each robot searches only its local area, there are fewer repeated searches.

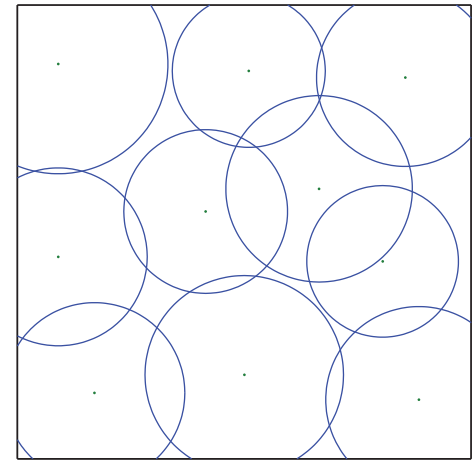


Figure 7: Distribution of robots and step size of each robot after exploration mission.

4. Experiments and Results

4.1. Simulation Experiments. To test the performance of the proposed RW method, simulation experiments were performed on the Webots platform. Figure 8 shows a screenshot of the simulation interface at the beginning of a simulation experiment. The effectiveness of the proposed RW method was evaluated by comparing its results with those of BM and LF. After each simulation experiment, MATLAB was used to process the image and calculate the coverage ratio (i.e., the ratio of the explored area to the total area).

For adequate comparison, we performed simulation experiments with one, 10, 20, and 30 robots. Each experiment lasted for 10 min (600 s), and the experimental area was a square with length of side $L = 20.0$ m. At the beginning of each experiment, all the robots were placed in the central area. Each robot moved with a speed of $v = 0.1$ m/s, and the parameter k was set as 0.1. The purpose of the area exploration was efficient area coverage, with the coverage ratio being used to evaluate the effectiveness of the proposed method. Each robot had a limited detection range, and the domain within the detection range was considered to have been covered only if the footprint of the robot passed over it, the footprint being the area covered by the robot. The coverage ratio is the ratio of

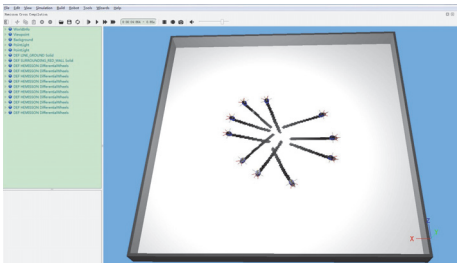


Figure 8: Exploration mission performed by 10 robots in simulation environment.

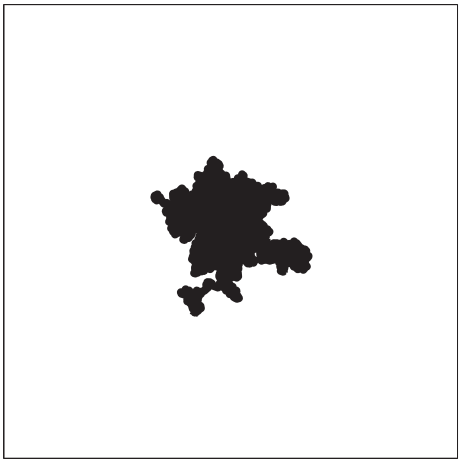


Figure 9: Result of exploration mission using 10 robots with BM. Black denotes explored area.

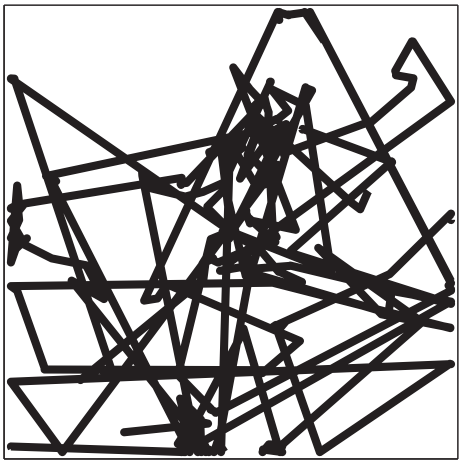


Figure 10: Result of exploration mission using 10 robots with LF. Black denotes explored area.

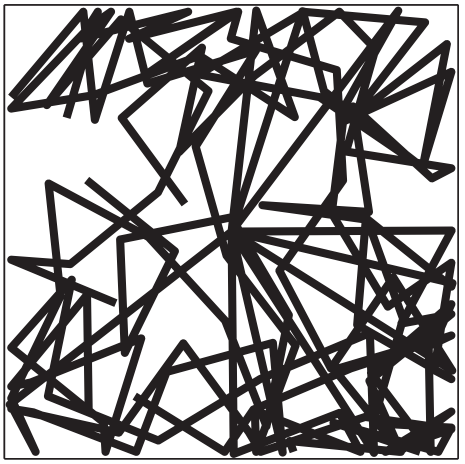


Figure 11: Result of exploration mission using 10 robots with proposed method. Black denotes explored area.

the explored area to the total area. For each RW method, the results of 20 simulation experiments were averaged to obtain the average coverage ratio.

Figures 9–11 show the results of exploration missions with different RW methods. Because BM generates a relatively small step size, the robots with BM did well in searching locally (Figure 9), but they did not reach other areas and thus many repeated searches were produced, thereby reducing the SE. Figure 10 shows that the robots with larger step sizes reached other areas to reduce the number of repeated searches, but the swarm robots did not become uniformly distributed and the SE remains to be improved. Figure 11 shows that the robots with adaptive step sizes not only reached other areas but also became uniformly distributed in the environment. Consequently, the proposed RW method resulted in far fewer repeated searches.

Table 1 gives the mean and standard deviation of the coverage ratio for each method for different numbers of robots. In each case, the proposed method achieved a much higher coverage ratio, performing significantly better than the other RW methods. Moreover, the standard deviation of each coverage ratio is relatively small, meaning that the SE of each of the three RW methods is stable.

To study how the initial positions of the robots influence the subsequent exploration mission, we also ran simulation experiments in which all the robots were initially distributed randomly in the environment. Figures 12–14 show the results of exploration missions with different RW methods. Because BM does well in a local search, the randomly distributed robots made fewer repeated searches (Figure 12). Because of the larger step size, there was little influence on the robots with LF (Figure 13). By adjusting the step size adaptively, the proposed method again distributed the robots uniformly in the environment (Figure 14), meaning that distributing the robots at random initially does not affect the SE of the proposed method.

Table 2 gives the mean and standard deviation of the coverage ratio for each method with different numbers of robots. The proposed method still achieves the highest coverage ratio. Compared with Table 1, the exploration mission using BM obtained a higher coverage ratio. From the standard deviation of the coverage ratio, we can conclude that the three RW methods possess stable SE.

Table 2: Comparison of proposed RW method with other methods when robots are distributed at random initially. Mean: average coverage ratio; Std.: standard deviation.

Methods	Brownian motion		Lévy flight		Our method	
	Mean	Std.	Mean	Std.	Mean	Std.
1	1.08%	6.98×10^{-4}	4.40%	4.30×10^{-3}	5.00%	5.71×10^{-3}
10	5.38%	4.04×10^{-3}	33.59%	1.02×10^{-2}	41.34%	1.17×10^{-2}
20	9.73%	4.11×10^{-3}	55.54%	1.07×10^{-2}	62.77%	1.30×10^{-2}
30	13.72%	6.86×10^{-3}	69.34%	9.59×10^{-3}	77.59%	1.08×10^{-2}

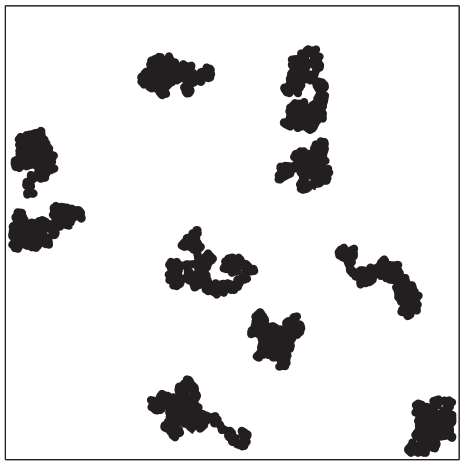


Figure 12: Result of exploration mission using 10 randomly distributed robots with BM. Black denotes explored area.

Figure 13: Result of exploration mission using 10 randomly distributed robots with LF. Black denotes explored area.

4.2. Experiments with Actual Robots. To assess further the effectiveness of the proposed method, we conducted exploration missions using e-puck robots [31], which are open tools that are used extensively for swarm experiments. With a diameter of 6.8 cm and a height 5.3 cm, each e-puck robot is equipped with (i) eight infrared proximity sensors for detecting obstacles, (ii) one CMOS camera to look for objects, (iii) one three-dimensional accelerometer, (iv) three microphones, and (v) one loudspeaker. The environment was designed as a rectangular area of 1.2 m × 1.5 m, and each robot moved at a speed of $v = 0.05$ m/s. In these experiments, it would have been difficult to measure the explored area to assess the effectiveness of the proposed method. Instead, an alternative method was used in which, as shown in Figure 15, the e-puck robots searched for objects scattered in the arena. Normally, the most effective random search method would be the one that finds all the objects in the shortest time. However, because the purpose here was to test the SE of each RW method, whenever a robot found an object the latter was removed from the environment manually.

Figure 16 shows a photograph of an experiment with actual robots. After a period of time, the robots were distributed almost uniformly in the environment. During each of these experiments, the e-puck robots found the objects relatively quickly in the initial stage of the experiment because there were relatively many objects. However, with fewer objects, the robots took much longer to find them. All the objects were found in 5 min with the proposed RW method and 7 min with the LF method. However, the robots searched predominantly locally in the middle of the arena with the BM method because of its small step size, and it took 15 min for them to find seven objects. These experiments show that the proposed method resulted in all the objects being found in the least amount of time. It can therefore be concluded that, of BM, LF, and the proposed RW method, the latter is the most effective.

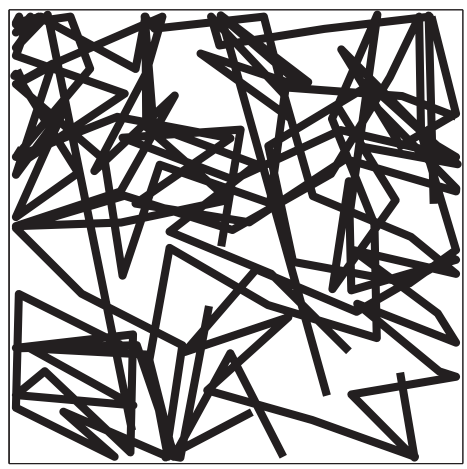


Figure 14: Result of exploration mission using 10 randomly distributed robots with proposed method. Black denotes explored area.

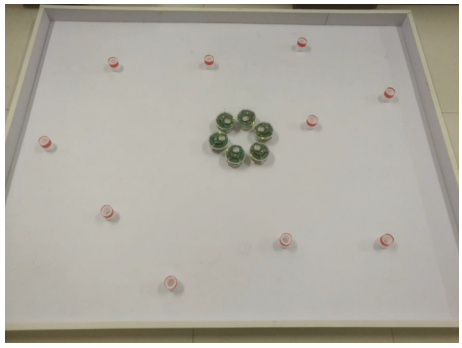


Figure 15: Photograph of arena used to conduct experiments with actual robots. At the beginning of each experiment, six e-puck robots in the waiting state were placed in the middle of the arena in which 10 objects were distributed randomly.

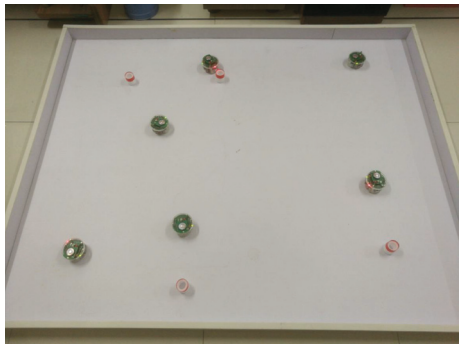


Figure 16: Photograph of experiment with actual robots using proposed method. After a period of time, the robots were distributed almost uniformly in the environment.

5. Conclusions and Future Work

This paper proposed an improved RW method based on the density of robots in the environment. Each swarm robot adjusts its step size adaptively to arrive at other areas, and the proposed method distributes the robots uniformly in the environment to reduce the number of repeated searches. The experimental results showed that the proposed method leads to the area being covered more efficiently than with either BM or LF, thereby improving the SE.

However, despite this, the efficiency of the proposed method still lacks theoretical analysis. Future work will therefore introduce the mathematical theory behind RWs in a straightforward manner and analyze theoretically the efficiency of the proposed RW method.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that no conflicts of interest exit in the submission of this manuscript.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 61573213 and 61673245, Natural Science Foundation of Shandong Province under Grant ZR2017PF008, National Key Research and Development Plan of China under Grant 2017YFB1300205, and Shandong Province Key Research and Development Plan under Grant 2016ZDJS02A07.

References

[1] S. Curtis, M. Rilee, P. Clark et al., "Use of swarm intelligence in spacecraft constellations for the resource exploration of the asteroid belt," in *Proceedings of the Third International Workshop on Satellite Constellations and Formation Flying*, pp. 24–26, 2003.

[2] S. Waharte, N. Trigoni, S. J. Julier et al., "Coordinated search with a swarm of UAVs," in *Proceedings of the Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, pp. 1–3, Italy, 2009.

[3] L. Pitonakova, R. Crowder, and S. Bullock, "Information flow principles for plasticity in foraging robot swarms," *Swarm Intelligence*, vol. 10, no. 1, pp. 33–63, 2016.

[4] U. Kei Cheang, K. Lee, A. A. Julius et al., "Multiple-robot drug delivery strategy through coordinated teams of microswimmers," *Applied Physics Letters*, vol. 105, no. 8, Article ID 083705, 2014.

[5] Y. Wei, T. Yasuda, and K. Ohkura, "Autonomous task allocation for swarm robotic systems using behavioral decomposition," in *Intelligent and Evolutionary Systems*, pp. 469–481, Springer International Publishing, Cham, Switzerland, 2017.

[6] D. W. Sims, A. M. Reynolds, N. E. Humphries et al., "Hierarchical random walks in trace fossils and the origin of optimal search behaviour," *Proceedings of the National Academy of Sciences*, vol. 111, no. 30, pp. 11073–11078, 2014.

[7] F. Ducatelle, G. A. Di Caro, A. Förster et al., "Cooperative navigation in robotic swarms," *Swarm Intelligence*, vol. 8, no. 1, pp. 1–33, 2014.

[8] E. A. Codling, M. J. Plank, and S. Benhamou, "Random walk models in biology," *Journal of the Royal Society Interface*, vol. 5, no. 25, pp. 813–834, 2008.

[9] A. Einstein, "Ueber die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen," *Annalen der Physik*, vol. 322, no. 8, pp. 549–560, 1905.

[10] D. W. Sims, N. E. Humphries, R. W. Bradford, and B. D. Bruce, "Lévy flight and Brownian search patterns of a free-ranging predator reflect different prey field characteristics," *Journal of Animal Ecology*, vol. 81, no. 2, pp. 432–442, 2012.

[11] T. Itami, "Controlling Brownian motion applied to macroscopic group robots without mutual communication," in *Proceedings of the 2014 19th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 903–908, 2014.

[12] A. A. Munishkin, D. Milutinović, and D. W. Casbeer, "Safe navigation with the collision avoidance of a brownian motion

obstacle," in *Proceedings of the ASME 2017 Dynamic Systems and Control Conference, American Society of Mechanical Engineers*, 2017.

[13] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "Robotic exploration, Brownian motion and electrical resistance," *Randomization and Approximation Techniques in Computer Science*, pp. 116–130, 1998.

[14] F. Martinez, E. Jacinto, and D. Acero, "Brownian motion as exploration strategy for autonomous swarm robots," in *Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2375–2380, 2012.

[15] V. Zaburdaev, S. Denisov, and J. Klafter, "Levy walks," *Reviews of Modern Physics*, vol. 87, no. 2, pp. 483–530, 2015.

[16] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev et al., "Lévy flight search patterns of wandering albatrosses," *Nature*, vol. 381, no. 6581, pp. 413–415, 1996.

[17] M. G. DaLuz, E. P. Raposo, H. E. Stanley et al., "Revisiting Lévy flight search patterns of wandering albatrosses, bumblebees and deer," *Nature*, vol. 497, no. 7165, pp. 1044–1048, 2007.

[18] A. M. Reynolds and M. A. Frye, "Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search," *PLoS ONE*, vol. 2, no. 4, p. e354, 2007.

[19] S. Focardi and J. G. Cecere, "The Lévy flight foraging hypothesis in a pelagic seabird," *Journal of Animal Ecology*, vol. 83, no. 2, pp. 353–364, 2014.

[20] G. M. Viswanathan, S. V. Buldyrev, S. Havlin et al., "Optimizing the success of random searches," *Nature*, vol. 401, no. 6756, pp. 911–914, 1999.

[21] V. V. Palyulin, A. V. Chechkin, and R. Metzler, "Lévy flights do not always optimize random blind search for sparse targets," *Proceedings of the National Acadamy of Sciences*, vol. 111, no. 8, pp. 2931–2936, 2014.

[22] G. M. Fricke, J. P. Hecker, J. L. Cannon et al., "Immune-inspired search strategies for robot swarms," *Robotica*, vol. 34, no. 8, pp. 1791–1810, 2016.

[23] R. Fujisawa and S. Dobata, "Lévy walk enhances efficiency of group foraging in pheromone-communicating swarm robots," in *Proceedings of the In System Integration (SII), 2013 IEEE/SICE International Symposium on*, pp. 808–813, IEEE, 2013.

[24] A. Schroeder, S. Ramakrishnan, M. Kumar et al., "Efficient spatial coverage by a robot swarm based on an ant foraging model and the Lévy distribution," *Swarm Intelligence*, vol. 11, no. 1, pp. 39–69, 2017.

[25] A. Deshpande, M. Kumar, and S. Ramakrishnan, "Robot swarm for efficient area coverage inspired by ant foraging: the case of adaptive switching between brownian motion and lévy flight," in *Proceedings of the ASME 2017 Dynamic Systems and Control Conference*, 2017.

[26] D. K. Sutantyo, S. Kernbach, P. Levi et al., "Multi-robot searching algorithm using Lévy flight and artificial potential field," in *Proceedings of the IEEE Safety Security and Rescue Robotics*, pp. 1–6, 2010.

[27] L. Palmieri, A. Rudenko, and K. O. Arras, "A fast random walk approach to find diverse paths for robot navigation," *IEEE Robotics Automation Letters*, vol. 2, no. 1, pp. 269–276, 2017.

[28] C. Dimidov, G. Oriolo, and V. Trianni, "Random walks in swarm robotics: an experiment with kilobots," in *Proceedings of the International Conference on Swarm Intelligence*, pp. 185–196, 2016.

[29] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes," *Physical*, vol. 49, no. 5, p. 4677, 1994.

[30] R. Fujisawa, S. Dobata, K. Sugawara, and F. Matsuno, "Designing pheromone communication in swarm robotics: Group foraging behavior mediated by chemical substance," *Swarm Intelligence*, vol. 8, no. 3, pp. 227–246, 2014.

[31] F. Mondada, M. Bonani, X. Raemy et al., "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, vol. 1, pp. 59–65, 2009.

Hindawi
Journal of Robotics
Volume 2019, Article ID 2970408, 10 pages
https://doi.org/10.1155/2019/2970408

Research Article

Optimal Skipping Rates: Training Agents with Fine-Grained Control Using Deep Reinforcement Learning

Adil Khan,[1] **Jiang Feng**,[1] **Shaohui Liu**,[1] **and Muhammad Zubair Asghar**[2]

[1]*School of Computer Science and Technology, Harbin Institute of Technology, NO. 92, Xidazhi Street, Harbin, Heilongjiang 150001, China*
[2]*Institute of Computing and Information Technology, Gomal University, D. I. Khan, KP, Pakistan*

Correspondence should be addressed to Adil Khan; adil.adil25@yahoo.com

Received 29 August 2018; Revised 11 November 2018; Accepted 23 January 2019; Published 3 March 2019

Guest Editor: Ling-Ling Li

Copyright © 2019 Adil Khan 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.

These days game AI is one of the focused and active research areas in artificial intelligence because computer games are the best test-beds for testing theoretical ideas in AI before practically applying them in real life world. Similarly, ViZDoom is a game artificial intelligence research platform based on Doom used for visual deep reinforcement learning in 3D game environments such as first-person shooters (FPS). While training, the speed of the learning agent greatly depends on the number of frames the agent is permitted to skip. In this paper, how the frame skipping rate influences the agent's learning and final performance is proposed, particularly using deep Q-learning, experience replay memory, and the ViZDoom Game AI research platform. The agent is trained and tested on Doom's basic scenario(s) where the results are compared and found to be 10% better compared to the existing state-of-the-art research work on Doom-based agents. The experiments show that the profitable and optimal frame skipping rate falls in the range of 3 to 11 that provides the best balance between the learning speed and the final performance of the agent which exhibits human-like behavior and outperforms an average human player and inbuilt game agents.

1. Introduction and Research Motivation

Due to the success and achievements of Google DeepMind [1] technologies, deep learning methods are widely employed in video games particularly in first-person-shooter (FPS) games such as Doom to obtain human-level control through raw pixels information [2]. So for this purpose ViZDoom is introduced which is a unique test-bed platform based on FPS Doom for deep learning research from raw visual information that engages the first-person perspective in a partially realistic 3D world that permits programming agents to play the game consuming the screen memory buffer.

Achieving such objectives has become possible now as computer systems have passed humans in terms of complex calculations and massive raw data processing; however, they still struggle in matching their ability to respond in complex 3D realistic environments [3].

In addition, visual signals are the major roots of information about the surroundings for living and artificial beings. Because of the advancements in dealing with the visual information, a progress has been observed in this area of research in the form of employing deep architectures in a set of Atari 2600 games [4] from raw pixel information where Atari 2600 games have been extensively accepted as a standard for visual learning systems. Now more significant progress is expected in 3D realistic environments due to the increase in computing power (GPU's [5] and TPU's [6]) and advancements in machine learning particularly visual learning and evolution of neural networks [7].

Visual deep reinforcement learning is a delightful research area of artificial intelligence as it places the player and the artificial intelligence agents on similar playing field particularly when it reaches partially observable environments [8]. Before the innovation of ViZDoom, a first-person-shooter based environment that could have permitted research on agents depending entirely on raw visual information was not present, which can be considered a serious cause obstructing the development of vision-based deep reinforcement learning because involving in vision-based reinforcement learning needs a huge volume of encoding

Figure 1: A sample screen from Doom showing the first-person perspective.

efforts [9]. The presence of a ready-to-use tool assists in managing experiments, and concentrating on the objective of research as the job of playing the first-person-shooter game in a 3D realistic environment is far more difficult than playing many Atari games because it includes a vast variety of skills such as routing through a map, gathering items, and identifying and battling opponents [10]. To facilitate and support computationally dense machine learning research ViZDoom is provided with off-screen execution or rendering characteristics. Off-screen execution minimizes the performance liability of truly displaying the game on the screen and makes it feasible to run the simulations on the servers by eliminating the need for using GUI [11].

So far, after studying the existing research on first-person-shooter games, specifically Doom, we thought to propose a devoted research to show how the number of skipping frames influences the learning process particularly using the ViZDoom AI research platform which could be of an extreme benefit to the research community to have a state-of-the-art frame skipping scale while training the agents or bots using any 3D realistic environment such as ViZDoom. Therefore, in this paper, the proposed work is based on finding an optimal frame skipping scale that provides the best balance or adjustment between the learning speed and the agent's final performance which might help in making the base for further improvement and research on FPS games. A sample screen of the Doom game is shown in Figure 1.

In order to explain the proposed research work in further detail, the paper is prepared in different sections. Section 2 explains the research on Doom using the ViZDoom AI research platform. Section 3 presents the proposed methodology. Section 4 shows results with experiments. Finally, Section 5 concludes the paper with future work.

2. Research on Doom Using the ViZDoom AI Research Platform

ViZDoom is a Doom-based AI research platform used for reinforcement learning from raw visual information. It allows developing AI bots that play Doom using only the visual information (the screen buffer). It is primarily intended for research in machine visual learning, and in particular for deep reinforcement learning. One of the recent research works based on visual reinforcement learning and the ViZDoom AI research platform is proposed in [12] by training an AI agent for the game Doom. The agent outperformed both human players and inbuilt game agents. However, in comparison, the concept proposed in this paper is different in the form of proposing an optimal scale for frame skipping while training game AI agents or bots.

The early research works based on visual reinforcement learning were performed long ago in [13, 14] by simply developing the robots soccer ball skills which were followed by state-of-the-art works using the ViZDoom AI research platform for training intelligent agents such as [15] in which a deep reinforcement learning based agent Clyde was developed to play the game Doom. Clyde participated in the Visual Doom AI Competition held at the IEEE Conference on Computational Intelligence and Games in 2016 [16]. In this competition, Clyde competed with 8 other bots and survived to achieve the 3rd place. Moreover, it also performed well in partially observable multiagent 3D virtual environments using deep visual reinforcement learning methods which were applied conventionally before in the fully observable 2D environments.

In the same way, another deep visual reinforcement learning based autonomous and comprehensive agent known as Arnold showed useful performance on the first-person-shooter game Doom. It performed well by simply considering the information on the screen in the form of raw pixels. Besides, deep reinforcement learning action navigation architectures based on convolutional neural networks were used to train Arnold for exploring and fighting the opponents on the game maps. Moreover, for effective training several techniques were employed such as augmenting high level game information, reward shaping, and using sequential updates to support Arnold in outperforming average human players and inbuilt game agents on different variations of the death-match by obtaining the highest kill-to-death ratio on both tracks of the Visual Doom AI Competition where Arnold was placed the 2nd in terms of the number of frags [17].

AI agents have been trained using the ViZDoom AI research platform in [18], which is a correlated research work on training agents that performed on two different scenarios, i.e., a simple basic move-and-shoot scenario and a complex maze exploring problem scenario using the convolutional deep neural networks, Q-learning [19], and experience replay memory for storing the game transitions [20]. The agents were tested on similar game scenarios or maps that demonstrated human-like behaviors and were able to outperform inbuilt game agents.

An AI agent is trained on two different maps, i.e., FlatMap and CIGTrack-1 in [21] using deep visual reinforcement learning and curriculum learning for first-person-shooter game Doom. Later on, this game AI agent won the Track-1 of the ViZDoom AI Competition held at the IEEE Conference on Computational Intelligence and Games in 2016 on known

maps by 35% greater score than the agent which secured a second position. The proposed framework for this agent is simple and links the state-of-the-art reinforcement learning concept of A3C model [22] with curriculum learning that does not rely on the opponent (adversaries) information; rather it uses the game states information from AI in real-time.

Reinforcement learning and deep learning are the real generic and useful methods for training AI bots or agents that result in rational and well-organized behavior for making intelligent decisions. In this regard, a correlated example can be found in [23] which employed deep visual reinforcement learning methods for training AI bots or agents to make basic and interactive intelligent decisions. Such RL and DL based methods are mathematically modeled using Markov decision processes (MDPs) [24]. An MDP is a data structure consisting of multiple parts or to be more specific it is an ordered set of data constituting a record or tuple such as (S, A, P, R, γ) where "S" is the set of different states, "A" is the set of changed actions the agent usually takes at each time step "t", "P" is the transitional probability of moving from one state (s) to another state ($\acute{s}$) taking an action (a), "R" is the reward function which represents the signal that the agent gets after taking several actions and changing states, and "γ" is the discount factor. Normally, using deep visual reinforcement learning methods the goal is to secure a policy π: s⟶a to improve the average expected discounted rewards and well renowned state-of-the-art general action value function Q^{π}(s, a) to learn a policy for estimating the regular expected rewards.

In addition, to support first-person-shooter games in an era in which technology has advanced and been upgraded to a high extent, it is of extreme significance to analyze the impact of skip counts (frame skip rates) while training AI agents particularly using the ViZDoom AI research platform. In the same way, besides computer games, in research on image and video processing the effect of frame skipping is also measured from an enormous group of user-studies by observing the performance over different sets of experiments where in some cases the data is in-sighted over a range of frame repeats normally to assess the video streaming which shows the significance of frame skip rates in the research domain of artificial intelligence.

Moreover, frame repeats are found significant for agent's movements as the degradation in performance for tasks related to agent or player's movements from lower frame skipping rates does not drop as quickly as for tasks related to the shooting with high frame skipping rates. Furthermore, the existing literature on training bots and agents reveals that only optimal frame skipping rates (frame repeats) are acceptable for getting well-balanced agents with better performance [25].

As in first-person-shooter games, frame repeats have a major impact on agents' performance, so sometimes the chosen frame repeat can be preserved even if scenario resolutions are sometimes to be sacrificed in order to reduce the training overburden and complexity level. On the other hand, it is worth noting that the scenario details such as the maps and their dark and light backgrounds along with several kinds of weapons add an interest to research on first-person-shooter games, so a tradeoff needs to be decided before choosing any option that spoils the real desired results or requirements.

To support and present more related work on first-person-shooter games, a generic model can be trained similarly to the one in [26] to simultaneously learn game features information such as the existence of opponents (adversaries) or objects along by lessening the Q-learning objective to reveal a progress in the training speed and performance of the model. In this regard, the proposed architecture in the referred article is modularized to be trained in the form of two different autonomous models for numerous phases of the game. The architecture suggestively outperformed inbuilt game AI agents and human players in death-match scenarios.

Now to mathematically model the overall concept which is presented in this section so far, a state-of-the-art DQN model is chosen that uses deep visual reinforcement learning to learn a policy for training agents to increase the sum of the expected discounted rewards, i.e., R_t; it can be mathematically represented as follows:

$$R_t = \sum_{i=t}^{T} \gamma \acute{t} - t_{\gamma \acute{t}} \tag{1}$$

where "T" denotes the game termination time and "γ" represents the discount factor, i.e., $\gamma \in [0, 1]$, that computes the importance of the future rewards. The Q-function for predicting the return after executing an action "a" in a state "s" for a given policy π can be mathematically defined as follows.

$$Q^{\pi}(s, a) = \mathbb{E}\left[R_t \mid s_t = \mathrm{s},\ a_t = a\right] \tag{2}$$

To achieve a maximum return using a function approximator for estimating the activation-value function Q, the DQN can use a neural network which is parametrized by θ, and to achieve an estimate of the Q-function of the current policy adjacent to the optimal Q-function, Q^* can be mathematically represented as follows.

$$Q^*(s, a) = \max_{\pi} E\left[R_t \mid s_t = s,\ a_t = a\right] = \max_{\pi} Q^{\pi}(\mathrm{s}, \mathrm{a}) \tag{3}$$

In other words, the goal is to find θ such that $Q_{\theta}(\mathrm{s}, \mathrm{a}) \approx Q^*(s, a)$. The optimum Q-function validates the Bellman optimality equation.

$$Q^*(\mathrm{s}, \mathrm{a}) = E\left[\mathrm{r} + \gamma \max_{\acute{a}} Q^*(\acute{s}, \acute{a}) \mid \mathrm{s}, \mathrm{a}\right] \tag{4}$$

If $Q_{\theta} \approx Q^*$, it is obvious to specify that Q_{θ} needs to be adjacent in verifying the Bellman equation that leads to the below loss function:

$$L_t(\theta_t) = \mathbb{E}_{s,a,r,\acute{s}}\left[\left(y_t - Q_{\theta_t}(s, a)\right)^2\right] \tag{5}$$

where t is the current time step, and $\mathrm{y_t} = \mathrm{r} + \gamma \max_{\acute{a}} Q_{\theta_t}(\acute{s}, \acute{a})$. The value of $\mathrm{y_t}$ is fixed and corresponds to the following gradient.

$$\nabla_{\theta_t} L_t(\theta_t) = \mathbb{E}_{s,a,r,\acute{s}}\left[y_t - Q_{\theta}(s, a) \nabla_{\theta_t} Q_{\theta_t}(\mathrm{s}, \mathrm{a})\right] \tag{6}$$

The approximation in (7) can also be used to compute the gradient instead of using an accurate estimate results from (6) for gaining the gradient.

$$\nabla_{\theta_t} L_t(\theta_t) \approx (y_t - Q_\theta(s,a)) \nabla_{\theta_t} Q_{\theta_t}(s,a) \quad (7)$$

One of the well-known approaches for breaking the correlation between sequential samples is to use the experience replay memory; i.e., at each time step the agent experiences (s_t, a_t, r_t, s_{t+1}) are saved in the replay memory; in addition, the Q-learning updates are executed on batches of experiences subjectively sampled from the replay memory. An ϵ – *greedy strategy* [27] can be used to create the next action at every training step with a probability ϵ for selecting the next action randomly and with a probability $1 - \epsilon$ allowing the best action of the network. In practice, it is common to start with $\epsilon = 1$ and to progressively decay ϵ to its end limit.

An approach of using the supervised learning techniques is proposed in [28] for a sensorimotor mechanism in immersive environments, which is moreover a correlated concept of training agents or bots using the ViZDoom AI research platform. The approach practices a high dimensional sensory stream and a lower-dimensional measurement stream. The concurrent structure of the streams delivers rich supervisory signals that empower training a sensorimotor control model by interacting with the environment. The model learns to act based on raw sensory input from a complex 3D environment. Such formulation empowers learning without a fixed goal at training time and pursuing dynamically changing goals at test time [29]. In this way, broad experiments were managed in 3D simulations based on the classical first-person-shooter Doom video game; the results validated that such an approach can outperform the current revolutionary inventions mainly on challenging tasks, and models trained with such an approach can efficiently generalize across environments and goals; for example, one of the models trained with such concept won the full death-match track of the Visual Doom AI Competition held earlier in unseen environments.

After studying the research works on such artificial intelligence competitions in computer games and the related research on developing and training agents using ViZDoom AI research platform, so far none of the articles could specify a devoted research on the optimal frame skipping rates for training AI agents based on Doom, which is a serious factor obstructing the improvement of vision-based reinforcement learning. In short, it is important and of interest to the community researching on agents and bots to have a basic frame skipping scale using game AI research platforms such as ViZDoom for visual deep reinforcement learning. To sum up, there is an essential research question significant enough (at least in its current state) to find an optimum scale of frame skipping which could be briefly defined as follows.

2.1. Research Question. What is the needed optimal skip count scale (range) in order to develop a balanced, well-trained, and robust agent particularly using any 3D AI research platform such as ViZDoom?

Figure 2: A rectangular chamber as a basic scenario.

As learning is the slowest when the agent does not skip any frame and learning is faster and smoother when the agent skips more frames, the primary purpose of the research is to examine how the number of skip counts influences the learning process and to find a standard and optimized skip counts scale (range) that can provide a balance or tradeoff between the final performance and the learning speed, specifically using any 3D AI research platform such as ViZDoom. But conversely, too large skip counts could make the agent graceless due to the lack of balance control that results in suboptimal concluding results.

3. Proposed Methodology

A rectangular chamber is considered as a basic scenario shown in Figure 2 where an agent spawns in the middle of the room's long wall, and a static monster spawns at arbitrary positions along the opposite wall. The agent moves toward the left and right and shoots. A solo shot is sufficient to massacre the monster. The scenario finishes either by killing the monster or with the completion of 300 frames, whichever comes first. The agent gets a score of 101 if it kills the monster, otherwise scores -5 for a miss hit and scores -1 for each action (living reward).

A convolutional neural network (CNN) architecture of three convolutional layers with 32 square filters, 7, 4, and 2 pixels wide, is used, respectively, which is shown in Figure 3. Each convolutional layer is trailed by a max-pooling layer with max pooling of size 2 and ReLU function for activation. Moreover, there is a fully connected layer with 800 leaky rectified linear units and an output layer with 8 linear units corresponding to the 8 combinations of the 3 available actions, i.e., left, right, and shooting [12].

Deep Q-learning, a method of deep reinforcement learning (see Section 2), is used to learn the policy. In order to experiment, the problem is modeled as a Markov Decision Process (MDP). A ϵ–greedy policy is used to select an action with linear decay ϵ. The convolutional neural network is used to approximate the Q-function trained with "Stochastic Gradient Decent" [30]. Besides, a reply memory is used to store the game transitions.

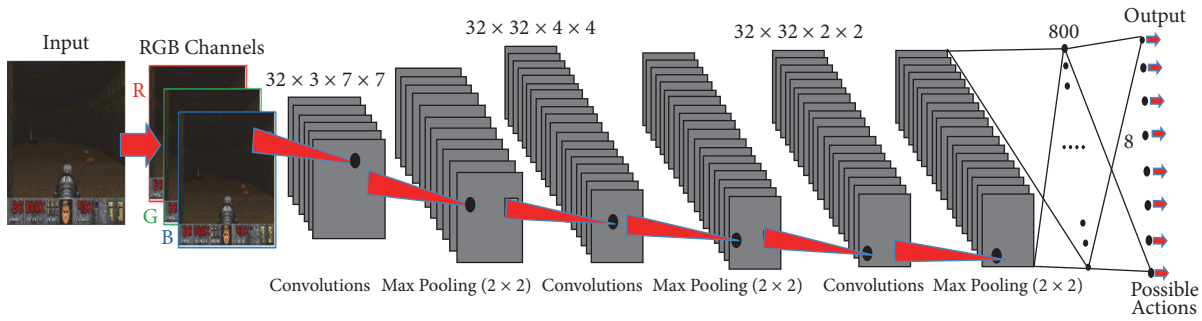


Figure 3: The proposed convolutional neural network (CNN) architecture employed in experiments.

4. Experiments and Results

The main objective of the experiments is to determine an optimal skip count scale (range) for producing (developing) well-balanced and robust agents or bots and to show how the number of skipped frames affects the learning process, particularly using any 3D game AI research platform such as ViZDoom.

4.1. Experiment-1 (Step-Size=2,000). The effect of skip counts is determined by training the agents for each skip count up to 20 epochs. The discount factor is set to γ=0.99, learning rate α=0.00025, replay memory capacity of 10,000 elements, resolution (45, 60), and minibatch size 32. Each time the agent learned for 40,000 steps involving executing an action, perceiving a transition, and updating the network. To determine and monitor how the number of skip counts affects the learning process while the agent learns, 100 test episodes are played after each 2000 learning steps and as well after the agent got fully trained.

All of the experiments are performed in PyCharm 2017.2 professional version using ViZDoom 1.1.5, OpenCV 3.3 [31], CMake 2.8+, GCC 4.6+, and Python 3.6 (64-bit) with NumPy on an Ubuntu 16.04.3 Server with Intel® Core™ i7-7700 CPU @3. 60 GHz x 8 and NVIDIA GeForce GTX 1080/PCIe/SSE2 GPU for processing CNNs. The whole learning and testing process in Figure 4 lasted for approximately 2 hours and 30 minutes, and in Figure 5 for approximately 1 hour and 30 minutes of playing approximately more than 35,000 game episodes collectively.

In Figure 4(L), the x-axis denotes the learning steps and y-axis denotes average learning results of the agent. The skip count legend shows the labels for 19 different skip counts considered during the experiments.

The performance of the agent for each considered skip count can be observed in the graph where the agent learns to get the perfect score gradually but the average learning score (result) is not better and high for all skip counts because there exist poor performances even below the score of 50. However, the results for the skip counts that are considered optimal (via experiments) are high and reach above the 70 as can be clearly observed in graph (L).

In Figure 4(R), the x-axis represents the testing steps and the y-axis represents the average test score (results) of the agent.

To test and verify the agent learning ability, the agent is tested on the same scenario(s) on which it was trained where it is observed that the shooting performance of the agent is not highly accurate and best for all skip counts except the optimal skip counts that range from 3 to 11 that can be observed for further study and understanding in graph (R).

To study and understand the performance of the agent for only optimal skip counts (3-11), a clear and simple view of the graph in Figure 4 is provided in Figure 5 by not considering the skip counts that result in irrational behavior and worst performance.

4.2. Experiment-2 (Step-Size=6,000). Similarly, another secondary experiment is conducted in order to confirm the validation of the proposed research question where the learning rate and the experimental setup are the same as described in experiment-1 except for the difference of learning step which is set to 6,000 in order to see any improvement or change in the agent learning and testing performance, or in other words to observe the effect of skip counts. After setting the learning step-size to 6,000 each time the agent learned for 1,20,000 steps including by executing an action, perceiving a transition, and updating the network. To analyze and observe the behavior and performance of the agent, similarly, 100 test episodes are played after each 6,000 learning steps and as well after the agent got fully trained.

This time the whole learning and testing process lasted for almost 8 hours and 30 minutes by playing approximately more than 5,79,292 game episodes.

Table 1 shows the agent average final score for each skip count with the total number of episodes played and the total amount of time taken. It is worth noting that the "episodes" column, which indicates that the swiftness of the learning system greatly depends on the number of frames the agent is permitted to skip during learning, means the more the number of skip counts, the more the number of episodes played and vice versa. In the table, the "average final score" column presents the final performance of the agent for each skip count where the highest scores for the optimal skip counts scale (range 3-11) are in italic font.

In Figure 6, the x-axis signifies the skip counts and the y-axis signifies the average final scores of the agent. The graph shows the performance (average scores) for all the considered skip counts where only the data points on or above the dotted

Table 1: Agent final performance for each skip count that affects the learning performance.

Skip count	Average Final Score	Episodes	Learning Time [Min]
1	67.1	1913	45.2
2	68.5	5729	31.1
3	*77.7*	8855	27.6
4	*77.6*	11733	25.4
5	*75*	14423	28.9
6	*74.8*	19332	28.7
7	*84.2*	23182	28.4
8	*74.1*	22121	28.2
9	*83.1*	26520	27.3
10	*74.1*	28411	28.5
11	*80.3*	28884	27.1
15	61.9	32597	27.2
20	70.7	42156	27.4
25	66	46985	26.2
30	73.6	45704	27.1
35	40.8	53034	27.4
40	61.4	52483	27.2
45	45.8	57653	27.5
50	43.4	57577	26.3

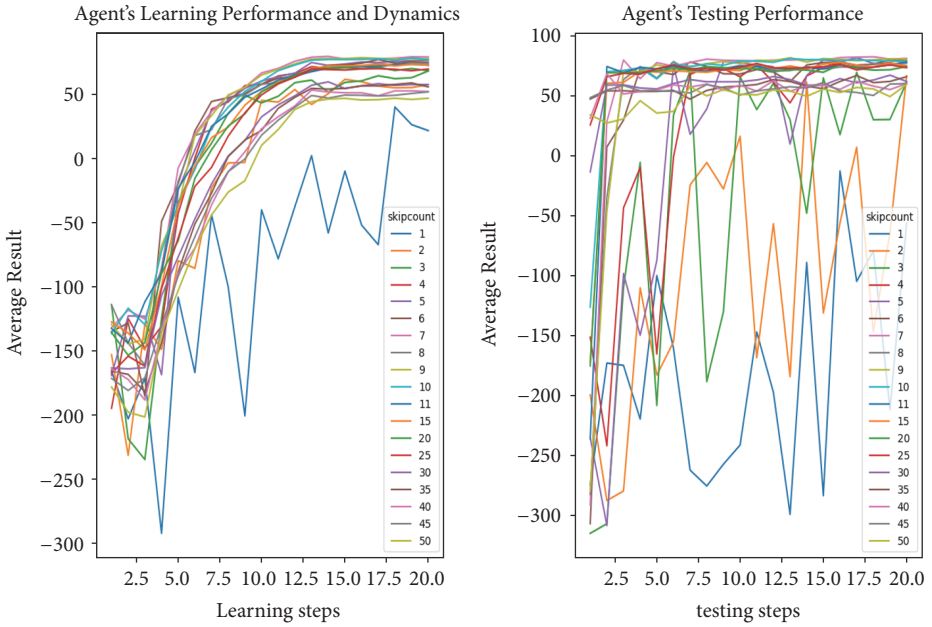


Figure 4: Showing the agent learning and testing performance for 19 different skip counts.

line are the concerning high scores for the skip counts from 3 to 11 which is the optimal scale (range) proposed for training game agents or bots such as Doom.

4.3. Comparison, Novelty, and Discussion. The comparison is made with the "A. Basic Experiment" publication from Michal et al. (2016), in which a neural network architecture employed in experiments is proposed, which is comprised of two convolutional layers that provided a base for suggesting an optimal skip count scale of 4 to 10.

However, in this paper, the proposed neural network architecture consists of three convolutional layers with the differences of learning and game settings, which means that the optimal skip counts scale is determined on a neural network architecture of three convolutional layers with modifications in hyperparameters, where according to the experiments and results it is proposed that the best optimal skip count scale lies in the range of 3 to 11.

Further in a simple move-and-shoot basic scenario, no concept of rewards shaping exists or applied that does not

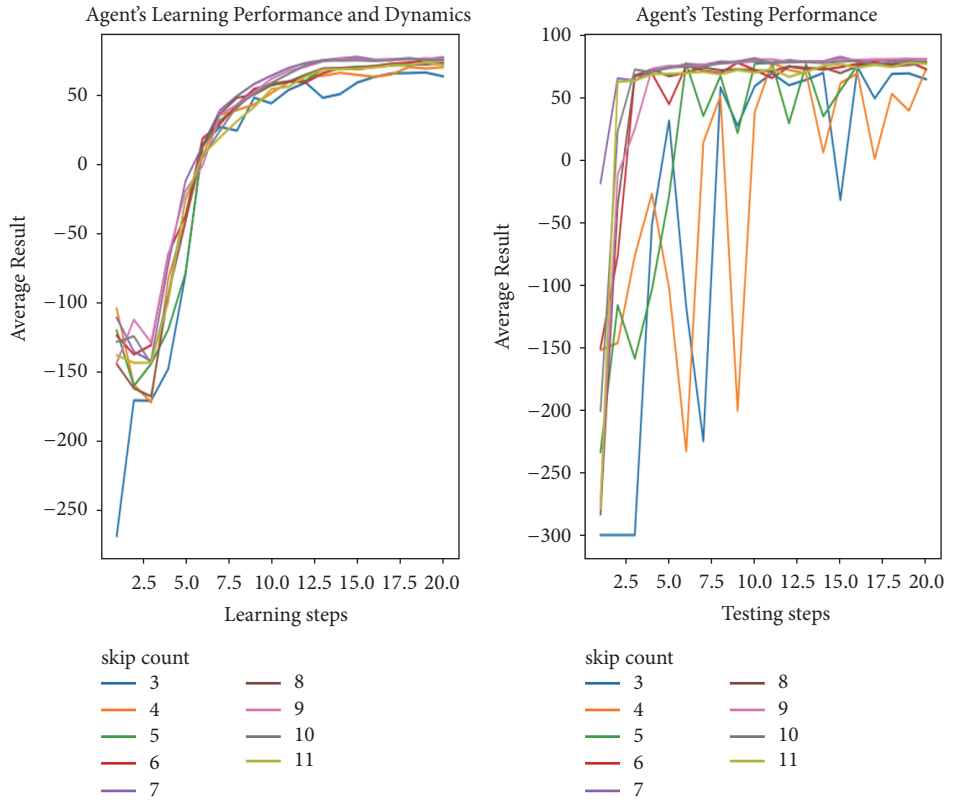


FIGURE 5: Showing the agent learning and testing performance for 3-11 optimal skip counts only.

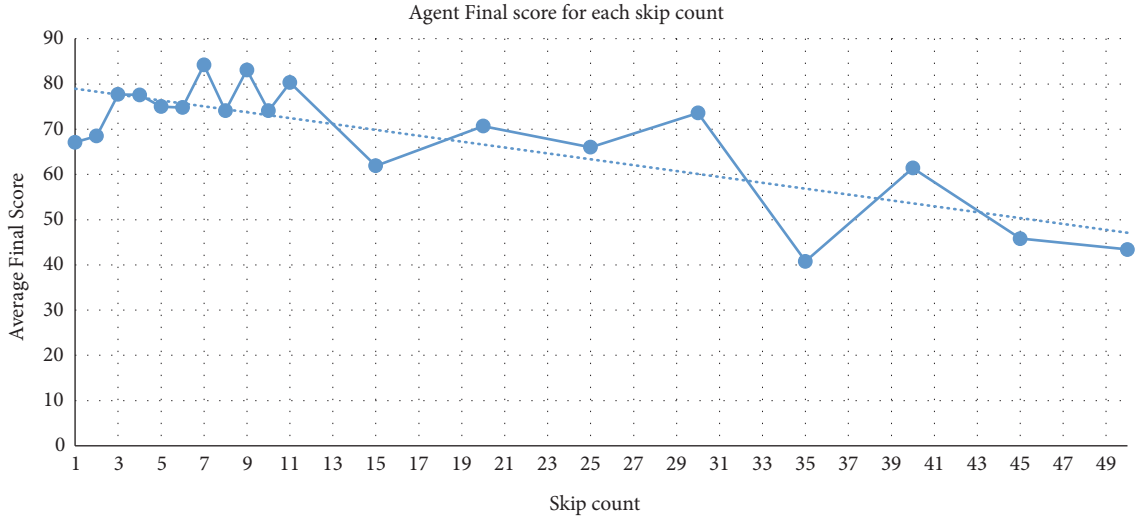


FIGURE 6: Proposing an optimum (standard) skip count scale and showing that the number of skip counts influences the learning process.

compute the final score, but in fact it is used while training the agent to help understand its goal. In such type of scenario(s), the agent movement matters where it is not allowed to move forward or backward except left and right.

Besides, the Michal et al. (2016) experiments were based on 15 skip counts, 7 of them are graphed (Figure 7). However, our proposed experiments, in comparison, are based on 19 skip counts (Figure 4) performed on a latest powerful GPU machine technology. In addition, unlike Michal et al., the experimental environment and the learning settings are also partially different as the learning rate is set to α=0.00025 and square filter width to 2 (third layer) with a minibatch size of 32.

In this paper, the final average results for the agent trained on different skip counts are at least 10% better than the results proposed by Michal et al. (2016) in [18] as they faced a few sudden, but transient, drops in the best and average score of the learning dynamics, which can be observed and

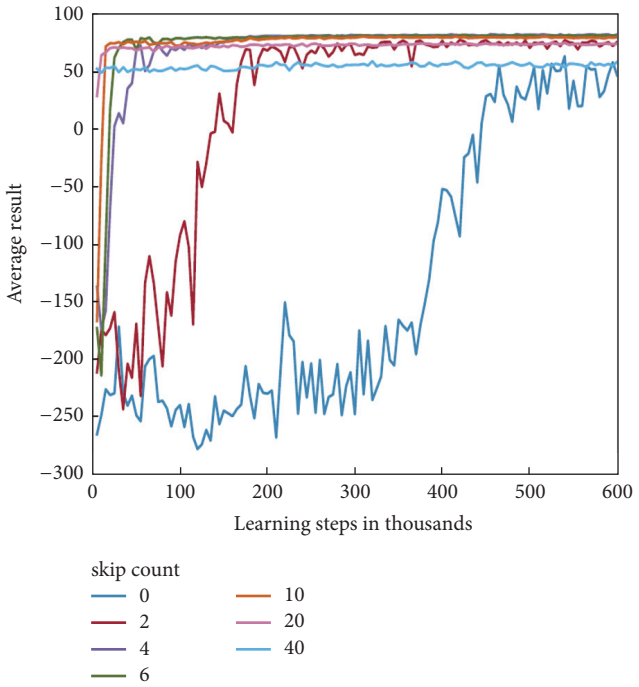


Figure 7: Target graph proposed in [18].

brainstormed by comparing Figure 4 with Figure 7 because the agents that learned with skip counts less than 3 are less robust, which cannot give an accurate and best results, and the agents trained with higher skip counts are more susceptible to irrational behaviors such as waiting idle or going the way contrary to the monster, which results in a higher change on the plots. Also, excessively huge skip counts make the agent clumsy due to the absence of fine-grained control that results in suboptimal concluding scores. On the other hand, the agents trained with a certain lower skip counts are found somehow robust, but the learning consumes a lot of time and results in a lesser number of scenarios. In short, to conclude, the skip count in the range of 3-11 delivers the greatest stability between the learning speed and the final performance. The outcomes also specify that it would be profitable to begin learning with extraordinary skip counts to maneuver the precipitous learning curve and progressively decline it to fine-tune the performance.

5. Conclusion and Future Work

In this paper, we proposed how the number of skip counts influences the learning process by employing convolutional deep neural networks (CDNN) with Q-learning and experience replay in a new game learning environment known as ViZDoom. According to the experiments, the results achieved are at least 10% better compared to the publication from Michal et al. (2016). Thus, it is concluded that skipping 3 to 11 frames is profitable in order to achieve human-like behavior of the agent in outperforming an average human player or inbuilt game agents. The learning steps are set to 2,000 and 6,000 and testing episodes 100 after each 2,000 and 6,000 learning steps for each epoch that would be kept dynamic and larger in future work for the different scenarios (collection of maps) such as a deadly corridor, defending the center, defending the line, and health gathering scenario(s).

Glossary

AI:	Artificial intelligence.
Atari 2600 Games:	The Atari 2600, originally called the Atari VCS, is the godfather of modern video game systems, and helped spawn a multibillion-dollar industry. Atari sold over thirty million of the consoles, and together with other companies sold hundreds of millions of games.
ALE:	Arcade Learning Environment is a framework that allows researchers to develop AI agents for the Atari 2600 games and currently supports over 50 games.
CNN:	Convolutional neural networks.
DL:	Deep learning.
Doom:	A 1993 first-person-shooter (FPS) video game by id software. It is considered one of the most significant and influential titles in video game history.
Death-match:	A gameplay mode that pits two or more players in a fight to the death.
FPS:	First-person shooter (FPS) is a video game genre centered around the gun and other weapon-based combat from a first-person perspective.
Game AI:	Game artificial intelligence or artificial intelligence in computer games.
Google DeepMind:	DeepMind Technologies is a British artificial intelligence company founded in September 2010, currently owned by Alphabet Inc. The company is based in London and additionally has research centers in Canada, France, and the United States. Their objective is to solve intelligence and to use it to make the world a better place.
GPU:	is a processing unit that performs rapid mathematical calculations, primarily for the purpose of rendering images. It is able to render images more quickly than a central processing unit (CPU) because of its parallel processing architecture, which allows it to perform multiple calculations at the same time.

GUI: Graphic User Interface.
Map: A map is a subpart of the virtual and immersive world in video games called scenario that contains multiple maps one can choose from.
RL: Reinforcement learning.
Scenario: In video games, it defines how a world works and looks like (maps).
TPU: A tensor processing unit (TPU) is a proprietary type of processor designed by Google in 2016 for use with neural networks and in machine learning projects. TPUs can offer an advantage for all ML applications implemented in Tensorflow.
ViZDoom: Is a Doom-based AI research platform for reinforcement learning from raw visual information. It allows developing AI bots that play Doom using only the screen buffer. ViZDoom is primarily intended for research in machine visual learning and in particular deep reinforcement learning.

Data Availability

The research data (raw and processed) used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially funded by the MOE–Microsoft Key Laboratory of Natural Language Processing and Speech, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China; the Major State Basic Research Development Program of China (973 Program 2015CB351804); the National Natural Science Foundation of China under Grant No. 61572155, 61672188, and 61272386; and the Higher Education Department KPK, Pakistan. The authors would also like to acknowledge and thank NVIDIA Corporation for the donation of two powerful sets of GPU machines.

References

[1] C. Beattie, J. Z. Leibo, D. Teplyashin et al., "DeepMind Lab," https://arxiv.org/abs/1612.03801.

[2] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[3] B. Kot, B. Wuensche, J. Grundy, and J. Hosking, "Information visualisation utilising 3D computer game engines case study," in *Proceedings of the the 6th ACM SIGCHI New Zealand chapter's international conference*, pp. 53–60, Auckland, New Zealand, July 2005.

[4] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing atari with deep reinforcement learning," https://arxiv.org/abs/1312.5602.

[5] J. D. Owens, M. Houston, D. Luebke et al., "GPU computing," in *Proceedings of the IEEE, 2008*, vol. 96, 5, pp. 879–899.

[6] N. P. Jouppi, C. Young, N. Patil et al., "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the Computer Architecture (ISCA, ACM/IEEE 44th Annual International Symposium on*, ACM SIGARCH Computer Architecture News, 2017.

[7] D. Hafner, "Deep reinforcement learning from raw pixels in doom," https://arxiv.org/abs/1610.02164.

[8] S. Koenig and R. Simmons, "Xavier: A robot navigation architecture based on partially observable Markov decision process models," *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, no. partially, pp. 91–122, 1998.

[9] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[10] D. Perez-Liebana, S. Samothrakis, J. Togelius, T. Schaul, and S. M. Lucas, "Analyzing the robustness of general video game playing agents," in *Proceedings of the 2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, Santorini, Greece, September 2016.

[11] M. Wydmuch, M. Kempka, and W. Jaskowśki, "ViZDoom competitions: playing doom from pixels," https://arxiv.org/abs/1809.03470.

[12] K. Adil, F. Jiang, S. Liu, A. Grigoriev, B. B. Gupta, and S. Rho, "Training an agent for fps doom game using visual reinforcement learning and vizdoom," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 12, 2017.

[13] M. Asada, E. Uchibe, S. Noda, S. Tawaratsumida, and K. Hosoda, "A vision-based reinforcement learning for coordination of soccer playing behaviors," in *Proceedings of the in Proceedings of AAAI-94 Workshop on AI and A-life and Entertainment*, Seattle, USA, 1994.

[14] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda, "Purposive behavior acquisition for a real robot by vision-based reinforcement learning," *Machine Learning*, vol. 23, no. 2-3, pp. 279–303, 1996.

[15] D. Ratcliffe, S. Devlin, U. Kruschwitz et al., "Clyde: a deep reinforcement learning DOOM playing agent. What's Next For AI," in *In Games: AAAI 2017 Workshop*, USA, San Francisco, 2017.

[16] G. N. Yannakakis and J. Togelius, "A panorama of artificial and computational intelligence in games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 4, pp. 317–335, 2015.

[17] D. S. Chaplot and G. Lample, "Arnold: an autonomous agent to play FPS games," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 5085-5086, USA, February 2017.

[18] M. Kempka, M. Wydmuch, G. Runc et al., "ViZDoom: a doom-based AI research platform for visual reinforcement learning," https://arxiv.org/abs/1605.02097.

[19] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[20] T. Schaul, J. Quan, I. Antonoglou et al., "Prioritized experience replay," https://arxiv.org/abs/1511.05952.

[21] Y. Wu and Y. Tian, "Training agent for first-person shooter game with actor-critic curriculum learning," in *Proceedings of the conference paper at ICLR 2017*, 2017.

[22] M. Babaeizadeh, I. Frosio, S. Tyree et al., "Reinforcement learning through asynchronous advantage actor-critic on a GPU".

[23] S. Bhatti, A. Desmaison, O. Miksik et al., "Playing doom with slam-augmented deep reinforcement learning," https://arxiv.org/abs/1612.00380.

[24] M. Hausknecht and P. Stone, *Deep recurrent q-learning for partially observable mdps*, abs/1507.06527, CoRR, 2015.

[25] M. Adamsson, "Curriculum learning for increasing the performance of a reinforcement learning agent in a static first-person shooter game 2018".

[26] G. Lample and D. S. Chaplot, "Playing FPS games with deep reinforcement learning," https://arxiv.org/abs/1609.05521.

[27] M. Tokic, "Adaptive ε-Greedy Exploration in Reinforcement Learning Based on Value Differences," in *KI 2010: Advances in Artificial Intelligence*, vol. 6359 of *Lecture Notes in Computer Science*, pp. 203–210, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[28] A. Dosovitskiy and V. Koltun, "Learning to act by predicting the future," https://arxiv.org/abs/1611.01779.

[29] P. Agrawal, "Computational sensorimotor learning," Tech. Rep., 2018.

[30] S. Ruder, "An overview of gradient descent optimization algorithms," https://arxiv.org/abs/1609.04747.

[31] G. Bradski and A. Kaehler, "Dr. Dobbs journal of software tools," *OpenCV*, vol. 3, 2000.

Hindawi
Journal of Robotics
Volume 2019, Article ID 4837058, 11 pages
https://doi.org/10.1155/2019/4837058

Research Article

Human-Machine Interface for a Smart Wheelchair

Amiel Hartman and Vidya K. Nandikolla

College of Engineering and Computer Science, California State University at Northridge, CA, USA

Correspondence should be addressed to Vidya K. Nandikolla; vidya.nandikolla@csun.edu

Received 10 July 2018; Revised 18 November 2018; Accepted 2 December 2018; Published 2 January 2019

Guest Editor: Ling-Ling Li

Copyright © 2019 Amiel Hartman and Vidya K. Nandikolla. 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 paper describes the integration of hardware and software with sensor technology and computer processing to develop the next generation intelligent wheelchair. The focus is a computer cluster design to test high performance computing for smart wheelchair operation and human interaction. The LabVIEW cluster is developed for real-time autonomous path planning and sensor data processing. Four small form factor computers are connected over a Gigabit Ethernet local area network to form the computer cluster. Autonomous programs are distributed across the cluster for increased task parallelism to improve processing time performance. The distributed programs operating frequency for path planning and motion control is 50Hz and 12.3Hz for 0.3 megapixel robot vision system. To monitor the operation and control of the distributed LabVIEW code, network automation is integrated into the cluster software along with a performance monitor. A link between the computer motion control program and the wheelchair joystick control of the drive train is developed for the computer control interface. A perception sensor array and control circuitry is integrated with the computer system to detect and respond to the wheelchair environment. Multiple cameras are used for image processing and scanning laser rangefinder sensors for obstacle avoidance in the cluster program. A centralized power system is integrated to power the smart wheelchair along with the cluster and sensor feedback system. The on board computer system is evaluated for cluster processing performance for the smart wheelchair, incorporating camera machine vision and LiDAR perception for terrain obstacle detection, operating in urban scenarios.

1. Introduction

Assistive technology is essential for elderly and disabled communities to help in daily living activities, socialization, and traveling. It is also known that the robotic application in the medical mobility can provide a better life for the people with both lower and upper extremity impairments. While assistive robotic technology is progressing rapidly to improve the mobility of people, several challenges remain to make this technology truly usable by the humans. One important aspect that requires research development is defining the control protocols between the human and the robot technology. There are different types of wheelchairs including basic, lightweight, folding, multi-function, powered, fully/partially autonomous and so on. And there are many types of control design to manipulate the functionality of the wheelchair, from basic drive to fully controlled wheelchair using brain-controlled interface. However, the power wheelchair users frequently report accidents, therefore our focus is to advocate the use of robotic technology, in particular sensor-based detection and navigation using smart wheelchairs [1–4].

Smart wheelchair is generally equipped with sensors, cameras and computer-based system as main processing unit to be able to perform specific task. Autonomous smart wheelchairs are controlled by human user interface where the human makes decisions at the highest level of operation and the smart control technology makes the rest of the motion automatic. The advances in autonomous smart wheelchair are embedded with computers and focuses heavily on the computer cluster architecture. The intelligence is added to a wheelchair platform around user control despite their disabilities, which makes the study of the human-machine interface (HMI) between the user and the wheelchair an important assistive robotic field of study. Standard electric powered wheelchair has little computer control with some level of motor control using a joystick. Therefore, researchers are focusing on computer-controlled wheelchair integrating sensors and intelligence to decrease the need of human

intervention. But attempts to build smart wheelchairs mostly lack robustness in motion control, perception and control techniques. This research work describes the development of a smart wheelchair integrated with a HMI and its interaction between sensory feedback and the computer control system [5–8].

The main focus of the paper demonstrates the design and performance of the interface between sensory feedback and the computer-controlled system. The real time data processing is addressed here for a smart wheelchair that functions as a low speed autonomous vehicle. The focus is on the implementation of mobile high-performance computing (HPC) cluster comprised of a multi-computer system connected over a local area network (LAN). A scalable network communication pipeline is developed that is expandable to accommodate additional computers as needed for light or intensive data processing loads. A user interface is developed that runs on a single client computer terminal in the network. It is used to activate and monitor the real time performance status of the software on other server computers in the network. Hardware and software parallelism is implemented to improve data processing. The following sections in this paper focus on: related work, smart wheelchair system, computer cluster network (system architecture), visual interface, computer cluster performance followed with results and conclusion.

2. Related Work

Ground vehicles developed for intelligent navigation come in many form factors. When developing a smart wheelchair it is significant to consider the operating conditions for the platform, in regards to the environment and also the user interaction with the system controls. A smart wheelchair platform can be constructed from different approaches including modifying an EPW, converting a manual wheelchair, or adding a seat to a mobile robot. Building a smart wheelchair with electronics added to an EPW results in the smart features being connected to the incorporated EPW controller to access the embedded drive train [10–12]. Connecting smart feature electronics to an EPW motion controller has the added benefit of being compatible with commercially available controllers, both joysticks, and alternative controllers for persons with multiple disabilities.

The type and placement of perception sensors on an EPW for smart wheelchair navigation can have a significant impact on the performance of the system as well as how the platform interacts with the wheelchair user. Some smart wheelchair designs favor sensor placement where the vehicle perception can be optimized but it impacts the EPW height and user access to the platform [10–13]. Newer developments in perception sensor technology can help to negate sensor placement limitations and reduce the impact of sensor placement on the platform accessibility [14, 15]. Furthermore, processing of high-resolution data sets from perception sensors for real time navigation can require high performance computing methods.

Many self-driving wheelchair platforms use a laptop computer for onboard processing [12, 13, 16]. However, using only a single laptop computer can impose significant constraints on the real time processing capabilities of a smart wheelchair. An alternative method is to use cloud computing to offload real time data processing from the smart wheelchair hardware [12]. In the past, high cost perception sensors have hindered smart wheelchair applications. The advent of low cost LiDAR in recent years, and the abundance of inexpensive machine vision cameras, allows incorporating autonomous navigation capabilities universally into EPW platforms. Currently, high bandwidth, low cost perception sensors can collect large amounts of data in real time, which requires significant computing power for the real time autonomous navigation for the smart wheelchair. While laptops and small form factor computers can be used for real time data processing, but as sensor hardware improves, the large perception sensor data sets will require alternative and more specialized embedded data processing hardware. Our approach is to use four small form factor computers for high performance real time processing on the smart wheelchair. Perceptions sensor data processing is distributed in the computer cluster configuration to improve real time performance. The design of the smart wheelchair uses EPW with computer motion control integrated into EPW joystick controller for motion control of the embedded drive train. Movable sensor mounts and armrests provides desirable sensor placement while allowing access from the front or side without increasing the wheelchair height [8, 9].

3. Smart Wheelchair System

The smart wheelchair system is comprised of a Jazzy 600ES electric-powered wheelchair (EPW) with added hardware and software for autonomous navigation and user interface. Mechanical modifications to the EPW incorporate a retrofitted footplate with rotating sensor tower at the front of the platform for mounting oscillating LRF sensors for range sensing and color cameras for terrain detection (Figure 1) [8, 9]. Optical rotary encoders are coupled to the drive train as part of the motion control interface for autonomous navigation. A National Instruments (NI) reconfigurable input/output (RIO) board is connected to the encoders for motion control feedback, the EPW joystick for control commands, and a computer for the real time control technology. A secondary mobile power system for the smart wheelchair is operated in conjunction with the EPW embedded power system to boost platform run time. The secondary power system is built using automotive and marine commercial off-the-shelf (COTS) electronic components incorporating a lithium-ion polymer (Li-Po) battery pack and power distribution panel with component level circuit protection (Figure 1) [8, 9].

The autonomous navigation capabilities of the smart wheelchair including perception, localization, path planning, and motion control, operate in the LabVIEW software environment. The path planning and motion control programs are optimized for the differential drive wheelchair platform. Small form factor (SFF) computers are connected over LAN with an Ethernet switch to explore the benefits of HPC cluster for real time sensor data processing and autonomous

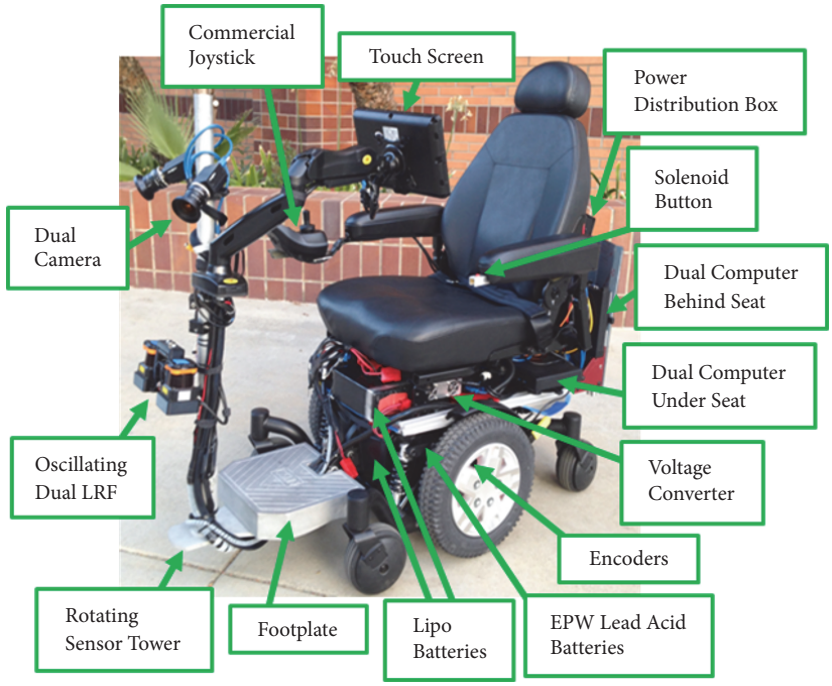


Figure 1: Retrofitted power wheelchair showing all the added features [8, 9].

navigation. Network automation is used to enhance control of the LabVIEW programs to simplify the user interface. A touch screen monitor user interface connects to the client computer for software system access. A real time performance monitor is incorporated into the user interface to provide sensor data visual feedback and track the status of the LabVIEW programs on the computer cluster during smart wheelchair operation [8].

4. Computer Cluster Network (System Architecture)

The primary objective of the HPC cluster is to improve real time data processing capabilities of the smart wheelchair's perception and intelligent navigation. The computer cluster system architecture can be seen in Figure 2 that shows the perception and cognition connectivity. The programs are expanded to function on a cluster as distributed computing network. The homogeneous cluster is constructed of four SFF desktop computers with low power components commonly found in laptop computers [9–11, 13, 17]. This allows achieving a small physical computer volume significant for the smart wheelchair mobile platform. Each computer contains a multi-core CPU with eight processing threads [18]. The LAN connecting the cluster computers is configured as an Internet Protocol version 4 (IPv4) class C private network. The scalable cluster network can incorporate additional computers, limited by available mounting space on the smart wheelchair platform.

The smart wheelchair system architecture (Figure 2) is separated into the two primary sections: perception and cognition. The computer cluster layout is designed so that one computer is designated for cognition and three computers are designated for the perception task. The perception sensor data processing is further separated into 3D point cloud depth data from the LRF sensors and RGB camera vision image processing. Cognition incorporates the decision process for path planning and interfacing with the EPW electronics for execution of motion control. The process of acquiring depth data starts with the LRF sensors that are used to collect a 3D point cloud of depth data. The data is filtered for noise and reduced to a horizontal plane 2D representation for obstacles in front of the smart wheelchair. A binary representation of the data as nearest obstacle edges is used to reduce the amount of data being transferred between the networked computers. This also reduces the amount of data being processed for path planning. The nearest obstacle edge simplified format is a means of combining LRF and camera vision data into useful map of terrain obstacles for cognition.

Two RGB cameras are used in front of the smart wheelchair to collect color image data of the environment. The cameras are connected to a computer through a USB hub for power and data transfer. The RGB data is processed through a series of filters before being converted into a binary representation of extracted terrain obstacles. The RGB data is also used to represent the terrain obstacle data for the user visual interface.

The perception data is transferred from each of the computers designated for the perception task over the Ethernet LAN to the cognition computer. The perception data is combined on the cognition computer and used for path planning determination or user visual interface. A distance histogram of obstacle leading edges is used for high speed real time path planning. The cognition computer is connected to

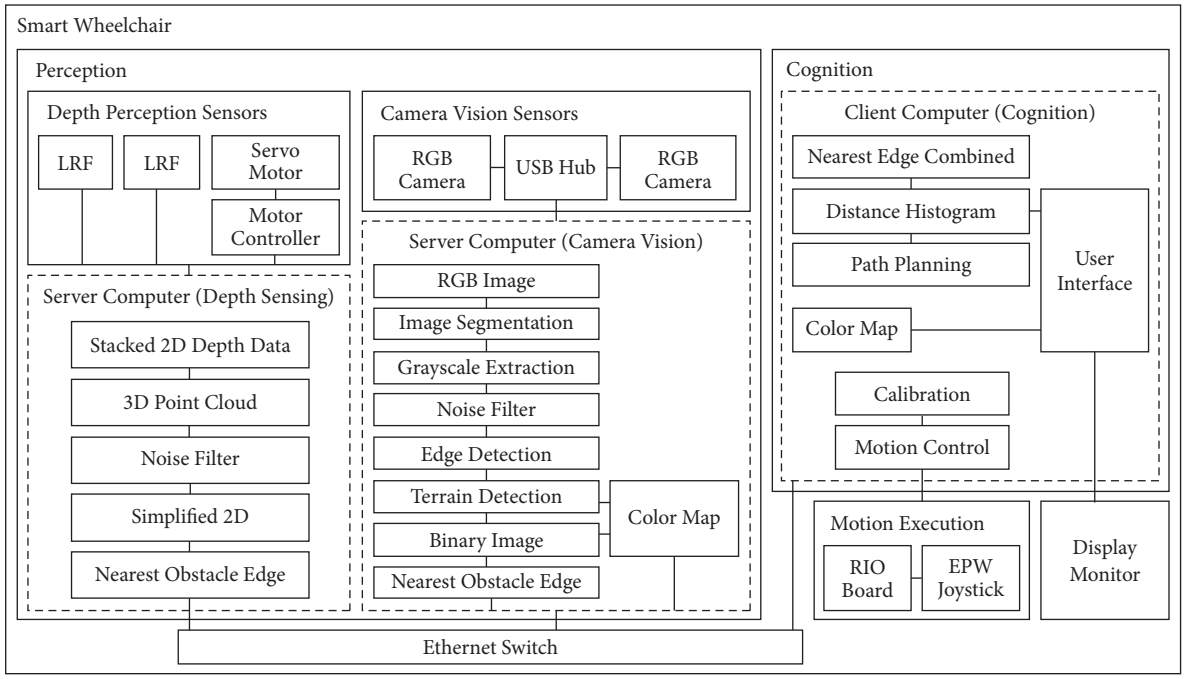


Figure 2: Computer cluster system architecture [9].

the EPW joystick using a National Instrument's RIO board for motion control of the drive train.

The smart wheelchair software is segmented into separate programs to operate concurrently on multiple computing nodes to reduce the time required for data processing. Data is transferred between the memory spaces of the computers and the RIO board using network published shared variables in LabVIEW. Beneath the client-server model for the computer cluster is the abstraction layer of the LabVIEW shared variable engine. This is hosted on the client computer as a virtual server and used to control the data transfer between computing nodes writing updates to and reading data from the network [19, 20]. The shared variables are configured for message passing without a buffer to transfer the most current data sets for real time autonomous navigation.

4.1. Network Automation. The goal of network automation is to simplify control of the autonomous navigation software distributed on the computer cluster network for real time operation of the wheelchair. The network automation subroutines operate in the background on the client computer transparent to the user interface. The subroutines are programmed with three primary functions: activating, monitoring and deactivating in coordination with the cognition program on the client computer. The deactivation of the server computer programs are used to make sure other programs are not running in the background when the data is not being processed by the cognition for navigation. The subroutines monitor the real time status of the server programs as part of the client computer to inform the wheelchair user if the system is operating normally. The details of the network automation architecture are described further.

The computer cluster network as seen in Figure 3 is separated into one client computer and three server computers based on the separation of the perception and cognition tasks. The client computer incorporates the user interface referred as the cognition computer. The client computer communicates with the three server computers used for perception over the LAN. The method of communication between computers on the network is using network variables that function as part of the LabVIEW shared variable engine. The cognition client computer is the host for the shared variable engine and the perception server computers are the subscribers. The cognition program is used to activate background network automation programs on the client computer. The background network automation programs contain the functionality to control and monitor the distributed perception programs on the server computers. They provide the automated functionality of launching the distributed algorithms across the network from the client computer. The network automation programs are also used to monitor the state of the perception and cognition programs to allow for more automated program interaction rather than limited to data transfer.

LabVIEW VI Server functionality is integrated in the client computer cognition program to dynamically call the subroutines to run independently so that the subroutines can respond to the activation and deactivation of the cognition program. Dynamic calling is used within the subroutines to target the server computer programs across the LAN in combination with static IP addressing. The cognition program for each server computer in the cluster activates a separate subroutine. Error handling is used within the network automation subroutines to prevent LabVIEW software failure. The cognition program and network automation

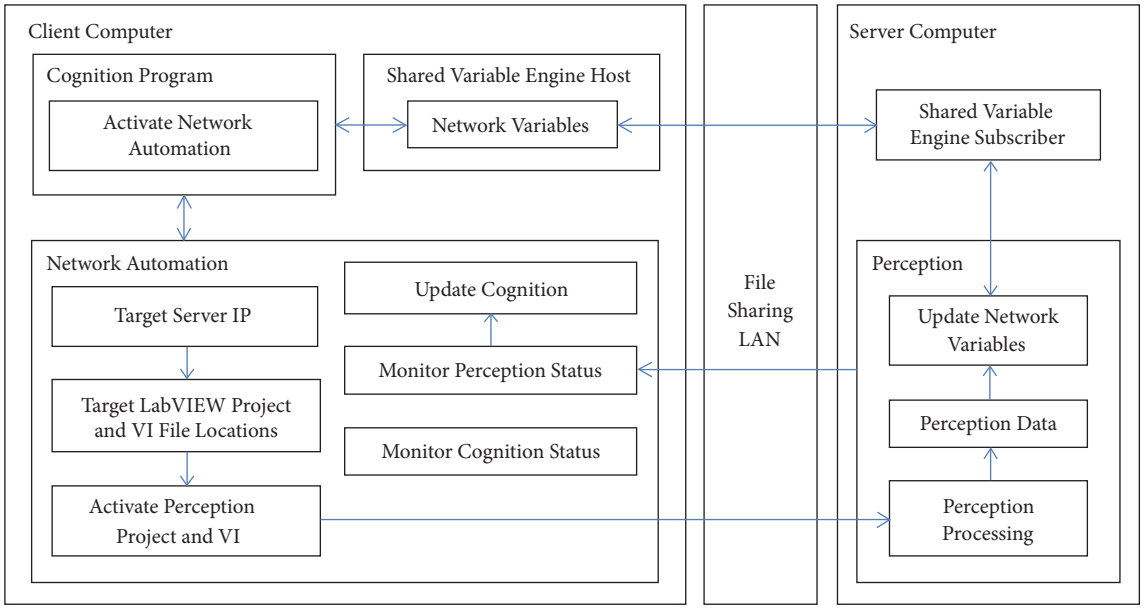


Figure 3: Network automation architecture [9].

subroutines work together to reset the status of the server computer programs if the autonomous navigation is reinitialized on the client computer. The reinitialization process is effective at resetting the cluster software from the client computer after correcting for accidental camera sensor hardware disconnect, without the need to access the server computers.

5. Vision Enhancement

The purpose of the vision system for the smart wheelchair is twofold: detection of terrain obstacles and integration of detected data into the user interface for visual feedback. The terrain obstacle detection features are expanded to improve processing with computer cluster hardware for higher resolution RGB images and multiple LRFs. Multiple sensors data is collected instead of a single sensor to improve data resolution and scope for feature detection [8, 13, 17, 21–23].

The color machine vision cameras used on the smart wheelchair are capable of delivering a 1920 horizontal by 1200 vertical (1920x1200) maximum pixel resolution RGB image. The maximum tested resolution for the real time image-processing program is 960x600 for each camera. Higher resolutions produce slow results for real time autonomy of less than 2Hz operating frequency. The minimum tested pixel resolution from each camera is 120x75, maintaining the 16:10 aspect ratio of the camera digital sensor. The number of image segments tested on each computer ranges from one to four. The two side-by-side cameras are connected to each vision server computer to increase the horizontal field of view. Therefore, the tested image resolution for a single vision computer varies from 240x75 to 1920x600 pixel resolution and these dual camera images are split up to four segments. To minimize the network data traffic, the segmented images are combined into a single image on the vision computers for processing before transferring to the client computer over the LAN. The tested resolutions and image segmentation are based on the combined dual camera image processed on each vision computer.

Multiple cameras fixed at different orientations on the smart wheelchair is challenging as each camera is exposed to different light conditions in outdoors during transit. Each RGB image is processed using a separate image-processing pipeline to adapt to different lighting conditions. Image segmentation is used to improve computer CPU parallel processing capabilities of the vision programs for faster real time image processing. Segmented images and data from multiple RGB sensors are concatenated to produce a combined terrain map.

The RGB data processing pipeline incorporates data reduction for improved processing time. RGB data resolutions are selected for real time transit based on the computer cluster processing capabilities. The cluster processing results for one vision computer is discussed. Testing results indicate the performance is similar on both vision computers and the cluster processing time or CPU usage is unaffected by the changes in the terrain image.

5.1. Terrain Perception. The RGB data is converted to multiple gray scale images for different feature extractions. The gray scale data is filtered for noise and enhanced to improve feature contrast. Edge detection is used to enhance feature extraction. Isolated terrain obstacles are converted to binary representations. The terrain data is converted to a nearest obstacle edge histogram (Figure 7) and combined with LRF data for obstacle avoidance and path planning. Multiple, color coated, binary images are combined for the terrain map visual interface (Figures 4 and 5).

The consolidated terrain data is represented in binary image to separate passable and impassable terrain or color coded to represent concrete, dirt, undesired terrain, shadows

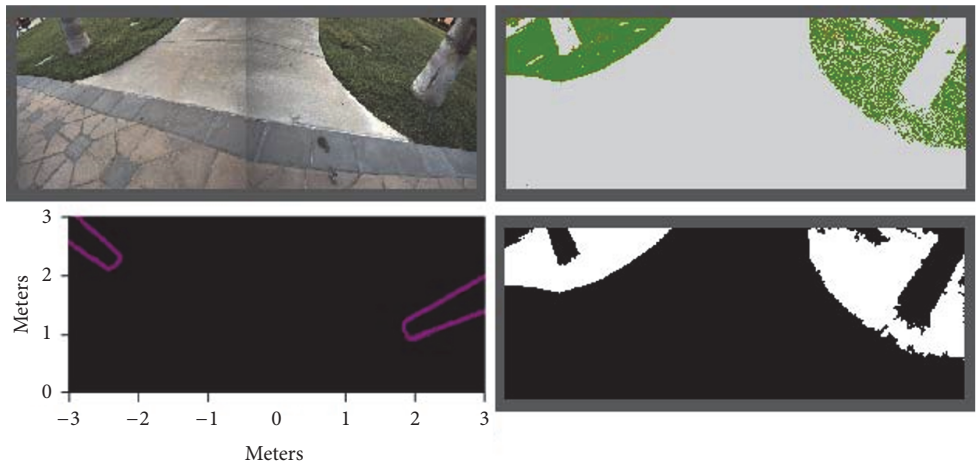
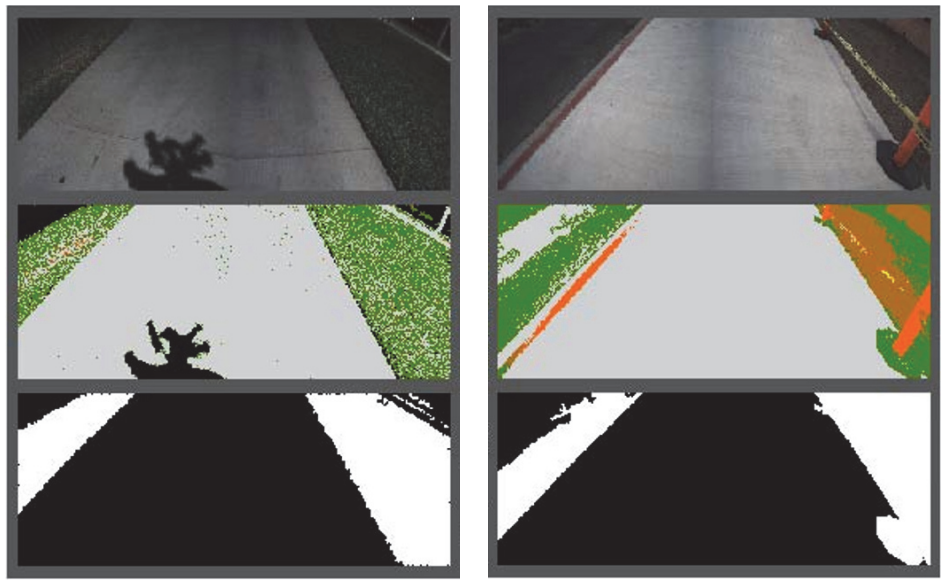


Figure 4: Image captured by dual camera, from top left clockwise: source image, color coded terrain, terrain obstacle binary image, LRF distance histogram [9].



(a) (b)

Figure 5: Dual camera images (a) shadows (b) construction marker. From top to bottom: source image, color-coded terrain obstacle visual interface, and terrain obstacle binary image [9].

and brightly colored construction markers in the pathway of the smart wheelchair (Figures 4 and 5). Figure 4 shows an outdoor daylight scenario captured by a dual RGB camera configuration. The image array data is used to combine the two images into a wider field of view representation as a single image. The LRF data shows the two tree trunks represented in distance histogram format, which is insufficient information for desirable scene for navigation. Therefore, the vision data is utilized to detect the edges of the grass as undesirable terrain, while filtering the three shades of concrete as a desirable open path. The data from the vision binary image is converted to a distance histogram shown in Figure 7 and combined with LRF histogram to create the nearest obstacle edge histogram for path planning. Since the trees are located within the grass terrain areas, the vision distance histogram indicates the nearest obstacle edge data for the scene.

Figure 5 shows two outdoor scenarios where the scene on the left side of the figure is a source image with shadow cast onto the terrain in front of the smart wheelchair. The color and gray scale histogram analysis used to filter the image detects the shadow contrast from the concrete path. The detected shadow is removed from the binary image representation and obstacle distance histogram. The right side in Figure 5 shows a sidewalk scene with road on one side and dirt with construction markers on the other. The red curb is detected along with the orange and yellow construction

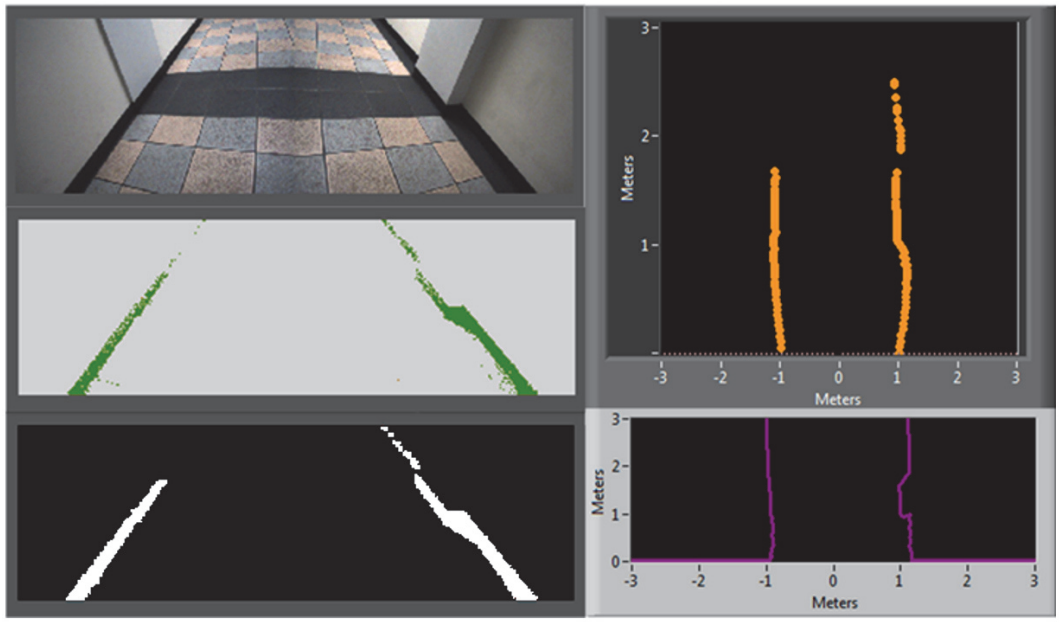


Figure 6: Camera images for indoor corridor and front panel display [9].

marker. In this scene, for real time operation the red curb and yellow construction tape are removed from the filtered binary image due to low resolution data. The dirt and base of the orange construction post is detected as side of the walkway along with the road and represented in the terrain binary image as seen in Figure 5.

5.2. Data Reduction Process. The smart wheelchair perception system uses 3D point cloud depth data and RGB images. The LRF hardware sensor is used for 3D depth data collection and data processing for real time operation. The raw 3D depth data processing has little effect on the real time performance of the system compared to RGB image processing (Figure 7). Consequently, data reduction is primarily applied to the data collected from RGB sensors. The RGB image data is converted to multiple gray scale images for filtering and feature extraction. Converting the data into binary image data further reduces the useful gray scale image. Additional filtering is applied to the binary images that contain smaller data set. Binary image data is further reduced to a distance histogram.

5.3. RGD and Depth Data Fusion. The RGB image data is combined with the LRF data for obstacle avoidance and path planning. The 3D point cloud depth data collected using the LRF sensors is filtered and converted to a stack of 2D depth data horizontal planes. The depth data planes are then compared and reduced to a single 2D nearest obstacle edge histogram for efficient path planning. The RGB image data is converted to high contrast gray scale and reduced to binary images in the data reduction process. Multiple binary images are combined to compile the terrain obstacle detection data. The stacked binary images are reduced to a nearest obstacle edge distance histogram to combine with the LRF distance histogram data. The camera distance histograms are calibrated to compensate for camera lens barrel and mounting angle perspective distortions. Distortion correction is incorporated into the distance histogram process to keep processing time low. The result is less powerful but faster calibration than image pixel mapping. The combined camera and LRF histogram extracts the closest edges of detected terrain obstacles.

5.4. Indoor and Outdoor Operation. Indoor and outdoor light conditions differ significantly. Outdoor daylight scenarios can often involve direct sunlight exposure, while indoor lighting conditions are typically low light in comparison. The LRF sensors are unaffected by light exposure from outdoor direct sunlight or indoor low light conditions. Bright and low light conditions have a more significant effect on RGB camera vision. Different calibration techniques for RGB data are required for indoor, low light and outdoor sunlight scenarios. Calibration of image contrast on RGB and gray scale images is used to compensate for changing light conditions.

Detecting the continuous smooth walls of the inside corridor shown in Figure 6 can be accomplished with a 2D LRF. The vision distance histogram shows the detection of the corridor walls compared to the multicolored tiles on the ground plane. Comparing the vision histogram to the LRF distance histogram shows the similarity in detection of the walls across multiple sensors. More complex object detection indoors is addressed in a similar fashion to outdoor scenarios with the combination of filtered RGB and 3D depth data.

6. Cluster Processing Results

The performance of the cluster processing is evaluated to determine the benefits of increased parallel computing for

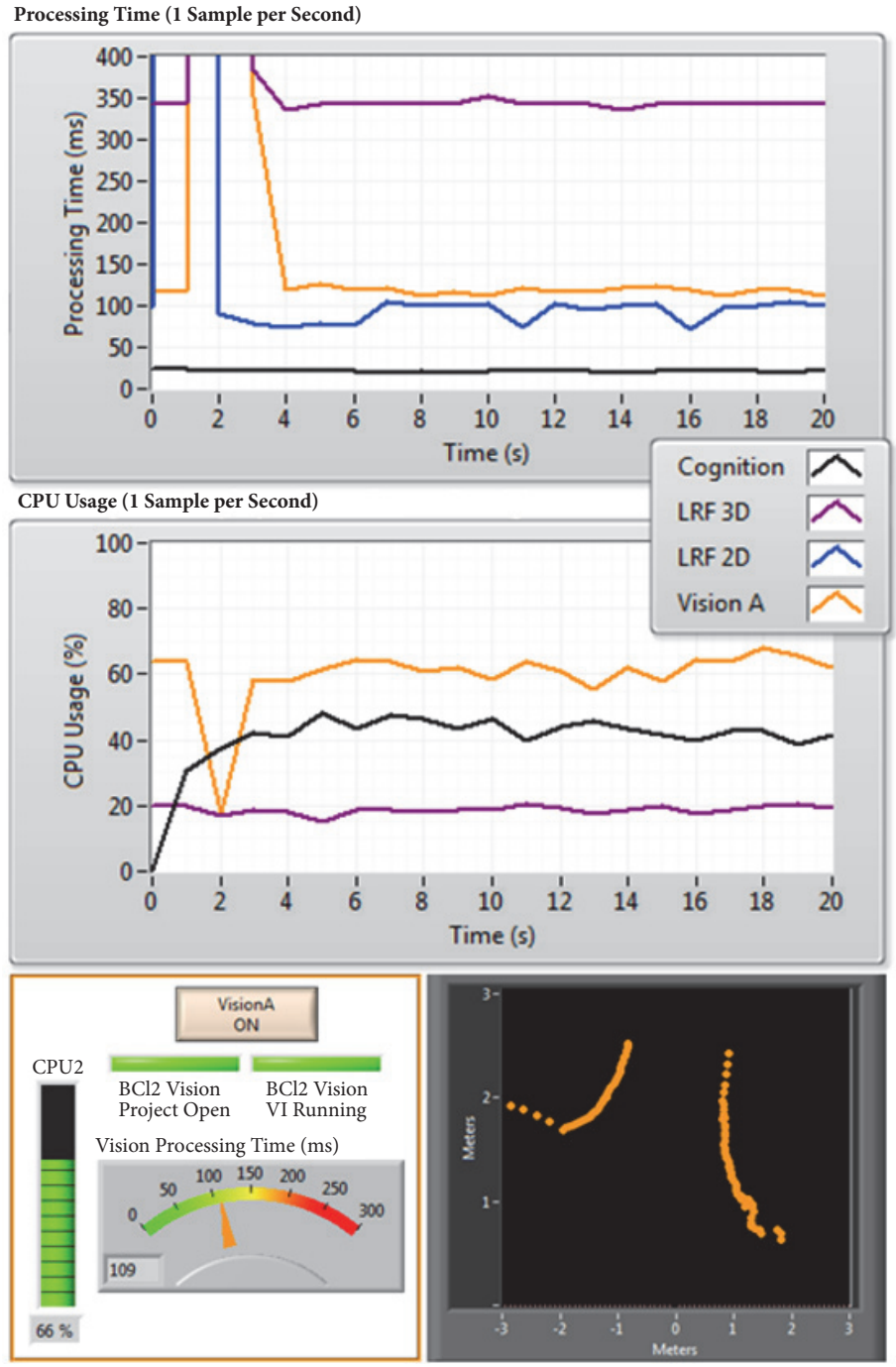


Figure 7: Performance monitor and user interface front panel display for three computers. Top: program processing time trace, Middle: computer CPU usage trace, Bottom left: simplified user interface with program status and Bottom right: camera vision distance histogram [9].

real time operation. Each portion of the distributed programs on a single computer is considered independently. The system as a whole is tested for the limiting factors in the overall performance of the cluster processing. The test results indicate that the client computer program performance and the dynamic 2D planer scan LRF program operates above 10Hz. Dynamic 2D refers to the scanning and processing of a single planer scan from the LRF. The LRF scanning plane is rotating around an additional pitch angle during the scanning process. The robot vision program is not able to operate above 10Hz, which is the maximum hardware resolution. Therefore, additional testing is conducted to determine the benefits of image segmentation on vision performance. Furthermore, the 3D LRF scanning is limited to less than 10Hz by the LRF hardware. The 3D LRF scanning process refers to the combination of a full sweep of single plane scans, and the

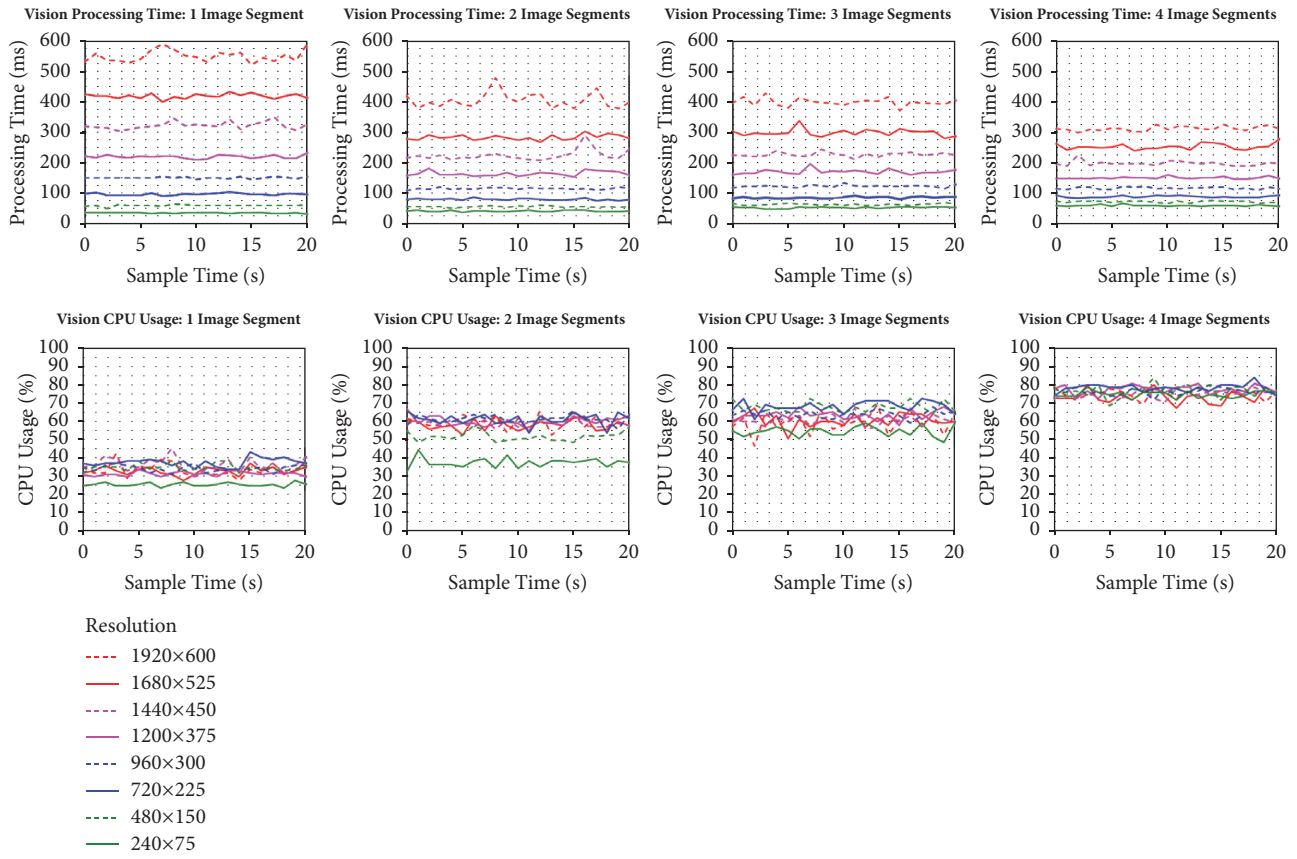


Figure 8: Vision program processing time (top) and vision computer CPU usage (bottom) of 20 sec samples [9].

stacking and processing of the combined data sets into a 2D representation of nearest obstacle edges.

6.1. Evaluation Metric. The real time performance of the cluster is considered for program processing time and computer CPU usage. The processing time includes the capture of a frame from the perception sensors, combining perception sensor data into a local map, instantaneous path determination and calculating necessary wheel speeds for motion control execution. Low processing time is critical for smooth real time operation of the smart wheelchair. The desired processing time for the smart wheelchair with an operating speed up to 4 MPH is 100 milliseconds (ms), or ten program iterations per second.

The CPU usage of each computer in the cluster is analyzed to determine the processing requirements of the programs. Increased CPU usage can represent improved task parallelism, while decreased CPU usage indicates less data processing or improved program efficiency. Accurate reading of the computer CPU usage for the methods used in LabVIEW is limited to one sample per second. While the processing time can be sampled effectively at a much higher frequency, the processing time sample rate is limited to one sample per second to correlate to the CPU usage.

6.2. Performance Monitor. A performance monitor is designed to show the status of computer cluster processing in a consolidated display for smart wheelchair real time operation. A LabVIEW front panel is created to show CPU usage, processing time and software activity of the individual computers in the cluster. The performance monitor is used for system evaluation during testing and development. Also the portions of the monitor are integrated into the user interface.

6.3. Startup Delay. There is a measurable startup delay between activating the distributed LabVIEW programs on the cluster and receiving updated information across the LAN. The delay using network automation and activating autonomous navigation from the client computer interface is within four seconds (Figure 8). After four seconds, normal runtime operation of the computer cluster is expected for autonomous navigation. The startup delay on the individual computers in the cluster is tested to be less than two seconds. For the start to stop operation of the smart wheelchair and switching between autonomous and manual control modes, the programs are kept running in the background to reduce the startup delay to a negligible amount. This comes with the cost of power consumption and therefore, reduced smart wheelchair operating time.

6.4. Runtime Performance. The runtime performance of the cognition, LRF and vision programs vary considerably (Figure 8). The cognition program on the client computer

Table 1: Vision processing near 10Hz operating frequency.

	720x225 Pixel Resolution (HxV)			960x300 Pixel Resolution (HxV)		
Image Segments	CPU Usage (%)	Processing Time (ms)	Operating Frequency (Hz)	CPU Usage (%)	Processing Time (ms)	Operating Frequency (Hz)
1	37	98	10.2	35	153	6.6
2	62	81	12.3	62	116	8.6
3	69	88	11.4	64	124	8.1
4	78	90	11.1	77	118	8.5

including path planning, motion control, and the user interface, is able to achieve a processing time of approximately 20ms. The cognition program uses approximately 40% of the client computer CPU during operation. Figure 8 shows the vision program processing time and vision computer CPU usage of 20 second samples for one, two, three, and four image segments at eight different dual camera resolutions for one vision computer.

The 2D LRF processing time is approximately 90ms, achieving 11Hz operating frequency, and the 3D LRF processing time is approximately 350ms, achieving only 3Hz operating frequency. The 2D LRF data is processed as part of the 3D LRF functionality and the 3D LRF utilizes 20% of the LRF computer CPU. The processing time and CPU usage of the vision program varies depending on image resolution and the number of parallel image segments. The general trend for the vision program resolutions ranging from 240x75 to 1920x600 is, an increase in resolution increases the processing time and an increase in image segments decreases the processing time. The benefit of reduced processing time from image segmentation is most significant at the highest tested resolution of 1920x600. At this resolution one image segment requires over 0.5 seconds to process a single image frame. Increasing the number of image segments to four, results in a decrease of approximately 43% in the average processing time to about 300ms. However, this processing time is well above the desired 100ms for smooth smart wheelchair operation.

At lower resolutions, the benefits of image segmentation decrease completely. At the lowest three tested resolutions, 720x225 and below, increased image segmentation negatively which impacts the processing time performance. This resulted in more image threads increasing the processing time. This presumably is due to the effect of processing overhead of segmenting and combining the images in addition to the rest of the image-processing pipeline. This can results in the minimum processing time of 37ms on average occurring at lowest resolution of 240x75 with only one image thread.

The CPU usage varies from about 35% to about 75% where the overall resolution has little effect on the CPU usage. Isolating the resolutions for processing time around 100ms yields the two resolutions of 720x255 and 960x300 (Table 1). With a resolution of 960x300 the processing time averages close to 120ms for two image segments (Figure 8), with one image segment performing significantly slower. Increasing the image segments above two shows no significant improvement. The 960x300 resolution configuration results in a 0.27 megapixel resolution operating at 8.5Hz for each vision computer, or 0.54 megapixel image processing using two vision computers in the cluster.

To achieve an operating frequency of at least 10Hz, the lower resolution of 720x225 is used for the current smart wheelchair operation. For a resolution of 720x225, the fastest average processing time of 81ms for the sample test is achieved with 2 image segments. Since a higher number of image segments produces no improvement on the processing time but does increase the computer CPU usage, two image segments is preferable. This configuration achieves 0.3 megapixel resolution image processing between two vision computers at over 12Hz operating frequency.

7. Conclusion

Advances are made on the technology of smart wheelchairs with sensors and driven by intelligent control algorithms to minimize the level of human intervention. The presented vision-based control interface allows the user to adapt and command the system at various levels of abstraction. The cognition motion control algorithms and the master remote control of the drive train is integrated in the control interface. The cluster consists of the distributed algorithms, the performance monitor and the network automation code that are executed from a single computer. With the research presented, the real time image processing is the limiting factor for processing speed in the current cluster configuration. From the results of the parallel processing capabilities of LabVIEW and the eight processing threads on the Intel I7 hyper-threading CPU, the task parallelism for the vision system can improve the CPU usage up to 80%. Currently in the demonstrated design, a dedicated computer is utilized for the LRF data processing cluster configuration that can be optimized. From the simulation results, it is predicted that combing the LRF algorithms onto the already fast performing cognition client computer can reduce the four computers to three. This modification is likely to achieve the same processing time results across the cluster. To improve the task parallelism of the vision algorithms or to provide space in the cluster for new algorithm development, the fourth computer could be utilized.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

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

Acknowledgments

The authors would like to show sincere gratitude to Dr. C. T. Lin for his continuous support, motivation, immense knowledge and enthusiasm in supporting the intelligent wheelchair research at California State University Northridge. This project would have been difficult to complete without his contribution and dedication.

References

[1] H. Soh and Y. Demiris, "Learning assistance by demonstration: smart mobility with shared control and paired haptic controllers," *Journal of Human-Robot Interaction*, vol. 4, no. 3, p. 76, 2015.

[2] H. Yuen, J. Pineau, and P. Archambault, "Automatically characterizing driving activities onboard smart wheelchairs from accelerometer data," in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5011–5018, Hamburg, Germany, September 2015.

[3] M. Ghorbel, J. Pineau, R. Gourdeau, S. Javdani, and S. Srinivasa, "A decision-theoretic approach for the collaborative control of a smart wheelchair," *International Journal of Social Robotics*, vol. 10, no. 1, pp. 131–145, 2018.

[4] C. Huang, Z. Wang, G. Chen, and C. Yang, "Development of a smart wheelchair with dual functions: Real-time control and automated guide," in *Proceedings of the 2017 2nd International Conference on Control and Robotics Engineering (ICCRE)*, pp. 73–76, Bangkok, Thailand, April 2017.

[5] F. Utaminingrum, M. A. Fauzi, R. C. Wihandika et al., "Development of computer vision based obstacle detection and human tracking on smart wheelchair for disabled patient," in *Proceedings of the 2017 5th International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 1–5, Dubai, United Arab Emirates, August 2017.

[6] S. A. Sheikh and D. R. Rotake, "An evolutionary approach for smart wheelchair system," in *Proceedings of the 2015 International Conference on Communications and Signal Processing (ICCSP)*, pp. 1811–1815, Melmaruvathur, India, April 2015.

[7] F. Utaminingrum, T. A. Kurniawan, M. A. Fauzi et al., "A laser-vision based obstacle detection and distance estimation for smart wheelchair navigation," in *Proceedings of the 2016 IEEE International Conference on Signal and Image Processing (ICSIP)*, pp. 123–127, Beijing, China, August 2016.

[8] A. Hartman, R. Gillberg, C. T. Lin, and V. K. Nandikolla, "Design and development of an autonomous robotic wheelchair for medical mobility," in *Proceedings of the 2018 International Symposium on Medical Robotics (ISMR)*, pp. 1–6, IEEE, Atlanta, GA, USA, March 2018.

[9] A. Hartman, *Design of a smart wheelchair for hybrid autonomous medical mobility [Thesis]*, 2018, http://scholarworks.csun.edu/handle/10211.3/201000.

[10] L. Fehr, W. E. Langbein, and S. B. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey," *Journal of Rehabilitation Research and Development* , vol. 37, no. 3, pp. 353–360, 2000.

[11] R. C. Simpson, "Smart wheelchairs: A literature review," *Journal of Rehabilitation Research and Development* , vol. 42, no. 4, pp. 423–435, 2005.

[12] J. Leaman and H. M. La, "A comprehensive review of smart wheelchairs: past, present, and future," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 486–489, 2017.

[13] C. T. Lin, C. Euler, P. Wang, and A. Mekhtarian, "Indoor and outdoor mobility for an intelligent autonomous wheelchair," in *Proceedings of the 13th ICCHP, International Conference on Computers Helping People with Special Needs*, 2012.

[14] C. A. Rockey, E. M. Perko, and W. S. Newman, "An evaluation of low-cost sensors for smart wheelchairs," in *Proceedings of the 2013 IEEE International Conference on Automation Science and Engineering, CASE 2013*, pp. 249–254, USA, August 2013.

[15] S. Desai, S. S. Mantha, and V. M. Phalle, "Advances in smart wheelchair technology," in *Proceedings of the 2017 International Conference on Nascent Technologies in Engineering, ICNTE 2017*, pp. 1–7, Navi Mumbai, India, January 2017.

[16] R. C. Simpson, D. Poirot, and F. Baxter, "The hephaestus smart wheelchair system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 2, pp. 118–122, 2002.

[17] P. Wang, N. R. Keyawa, and C. Euler, "Radial polar histogram: obstacle avoidance and path planning for robotic cognition and motion control," in *Proceedings of the SPIE 8301, Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*, Burlingame, Calif, USA, 2012.

[18] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets," RFC Editor RFC1918, 1996, Avaliable: https://tools.ietf.org/html/rfc1918.

[19] http://www.ni.com/white-paper/4679/en/.

[20] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2014.

[21] D. An and H. Wang, "VPH: A new laser radar based obstacle avoidance method for intelligent mobile robots," in *Proceedings of the WCICA 2004 - Fifth World Congress on Intelligent Control and Automation*, pp. 4681–4685, Tsinghua University, China, June 2004.

[22] J. Gong, Y. Duan, Y. Man, and G. Xiong, "VPH+: An enhanced vector polar histogram method for mobile robot obstacle avoidance," in *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, ICMA 2007*, pp. 2784–2788, Beijing Institute of Technology, China, August 2007.

[23] National Instruments, "Reference example of a system monitor displaying CPU and memory usage," Available: http://www.ni.com/example/31299/en/, 2011.

Hindawi
Journal of Robotics
Volume 2018, Article ID 6892153, 12 pages
https://doi.org/10.1155/2018/6892153

Research Article

Toward Dynamic Monitoring and Suppressing Uncertainty in Wildfire by Multiple Unmanned Air Vehicle System

Sharon Rabinovich, Renwick E. Curry, and Gabriel H. Elkaim

Computer Engineering, University of California Santa Cruz, Santa Cruz, California, USA

Correspondence should be addressed to Sharon Rabinovich; srabinov@ucsc.edu

Received 16 October 2018; Accepted 11 November 2018; Published 18 November 2018

Academic Editor: L. Fortuna

Copyright © 2018 Sharon Rabinovich 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.

Containing a wildfire requires an efficient response and persistent monitoring. A crucial aspect is the ability to search for the boundaries of the wildfire by exploring a wide area. However, even as wildfires are increasing today, the number of available monitoring systems that can provide support is decreasing, creating an operational gap and slow response in such urgent situations. The objective of this work is to estimate a propagating boundary and create an autonomous system that works in real time. It proposes a coordination strategy with a new methodology for estimating the periphery of a propagating phenomenon using limited observations. The complete system design, tested on the high-fidelity simulation, demonstrates that steering the vehicles toward the highest perpendicular uncertainty generates the effective predictions. The results indicate that the new coordination scheme has a large beneficial impact on uncertainty suppression. This study thus suggests that an efficient solution for suppressing uncertainty in monitoring a wildfire is to use a fleet of low-cost unmanned aerial vehicles that can be deployed quickly. Further research is needed on other deployment schemes that work in different natural disaster case studies.

1. Introduction

In any region undergoing some form of environmental distress, it is very important to detect changes occurring on the ground. In some cases, the environmental incident has spatial changes, and the incident spreads steadily. In other cases, it becomes difficult to follow the incident without knowing how it is evolving. Having a system that follows the event helps rescue human lives, monitor the incident, and allow the human responders to take better actions (as well as deploy assets in an optimal manner to mitigate the incident).

It is of great importance to monitor and respond to natural phenomena (e.g., fires) and national security disasters (e.g., emitting source). One needs to be able to explore a wide area and search for the source of hazardous substance emissions or the expansion of a fire front. In 2016 alone, Federal agencies reported 67,595 fires and an estimated cost of fire suppression of approximately 2 billion U.S. dollars [1]. In addition to financial loss and significant damage to the environment, wildfires threaten the lives of firefighters and civilians during these fire extinguishing operations [2].

There are two main reasons why a solution to fire tracking needs to be found. The first relates to modeling; it is very challenging to predict the fire frontier as a stochastic phenomenon dependent on weather conditions, terrain, and fuel (flammable materials) [3].

Secondly, operational aspects are exposed to severe limitations and constraints. The resources to respond to and monitor disasters are still quite limited. In the aviation section of the National Interagency Fire Center's annual summary of wildfire activity in 2012, there were many requests for large air tankers, which were Unable To be Filled (UTF). The number of cases of firefighters needing air tankers that were unavailable reached a high of 48 percent in 2012 [4] (This means that, in 2012, almost half of all requests for tankers to bomb fires were unanswered due to limited resources).

An incident with a dangerous spread area requires immediate exploration. Some examples are distributed fire spots and chemical threats; however, there are many others. This type of scenario requires surveillance to search for threats, but human observers are difficult to deploy because the task is dull, dirty, and/or dangerous. Wildfire monitoring missions

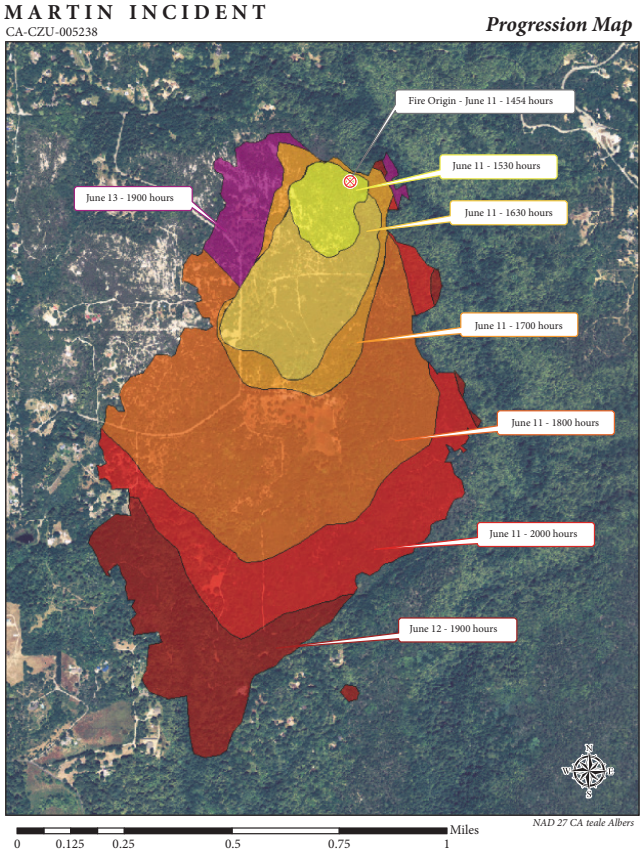


FIGURE 1: Progression map of the Martin Incident. The image is processed **after** the incident and relies on number of sources (from CAL FIRE).

are a perfect example of why a solution needs to be developed. Wildfires (and all natural and national threats phenomena) require urgent attention and an efficient response to monitor and contain their spread.

Figure 1 shows an example of a progression map of a wildfire incident. The periphery grows in time, and the boundary of the wildfire spreads. Knowing that the terrain is a dominant factor helps to understand the outcome in relation to the vegetation type, and knowing the actual weather explains the direction of propagation. The incident last for almost two days and spread out over a distance of 2km. It points to the necessity for a real-time monitoring system.

1.1. Existing Monitoring Systems. Previous studies have examined two different solutions: one on the ground and one in the air. In the first case, ground vehicles are used to explore the area. Use of ground vehicles depends on how passable is the area which needs to be explored. Ground vehicle capability is not necessarily suitable for scenarios with difficult terrain. In the second case, the deployment is in the air, the motion is smooth, and the area can be observed much more efficiently. In addition, most of those scenarios have a critical time factor. The systems phenomena are dealing with time, and because the existing system capabilities are limited, they cannot collect all spatial/temporal information at once. Whenever the observing vehicle is positioned in any one place, the system necessarily misses events in other places within the search area.

Ref. [6] describes existing projects that support disaster management in real time mainly systems that are space-based (e.g., GlobalStar), or high altitude and long duration (e.g., Global Hawk). The projects reinforce the importance of tracking events like floods and earthquakes, and how the tracking events help to monitor the incidents and handle them effectively from the ground control segment.

Ref. [7] describes remote sensing techniques and sensor packages that have been used on UAVs (Unmanned Air Vehicles). The author argues that these techniques can serve as the main solution for various disasters. Reviewing the literature on the development of UAVs, including projects with different types of sensors (IR/Visual) and platforms (fixed-wing/rotary-wing), he concludes that Multi-UAVs can be used to avoid the drawbacks of approaches that are based on either land or space.

1.2. Multi-UAV. The ideal mission for a UAV is monitoring a wide area and searching for the source of emission of a hazardous substance or expansion of a fire front. There has been a rising interest in increasing UAV efficiency and reliability. Autonomous vehicles have been used in a variety of applications for surveillance, search, and exploration [8]. In surveillance problems, the target space needs to be surveyed continuously. It differs from a coverage problem, which involves optimal deployment of the sensors for complete coverage of the target area. It also differs from an exploration problem, which deals with gaining more information about the bounded area [9]. This exploration research moves in two directions. The first focuses on how to pinpoint the source of an odor [10, 11]. In this area of research, robots are tasked with detecting, tracking and seeking the odor source efficiently.

The second direction of this exploration research focuses on how to establish the boundary or perimeter of a spreading phenomenon in order to monitor the area and prevent human exposure to risk [12]. Because of the spatial limitations of a single UAV, most research currently focuses on how to monitor large areas by operating multiple inexpensive simple UAVs simultaneously [13].

Though the studies mentioned above are significant, they focus on exploring the environment using clues (e.g., aerosol diffusion) for tracing emission sources. Moreover, the techniques used to detect the plume or periphery are strongly dependent on the spatial gradient change of the underlying tracked phenomenon. The research presented in our work proposes to explore the area by using approximate inference methods [14] and statistical reasoning [15]. The developed method takes into consideration the operational aspect of the mission in addition to the statistical characteristics of the underlying phenomenon.

1.3. Coordination. Most of the multi-UAV systems are designed to address problems related to specific research in a particular environment of interest. The UAVs cooperate and share data to obtain information on a certain aspect of the

environment. Regardless of the number of UAVs and size of the AOI (Area Of Interest), cooperative systems deliver an improved overall picture of the environment through coordination.

The design of cooperative systems mainly discusses the control strategy (e.g., centralized or decentralized) and the level of autonomy. Framework design inspired by a biological system has been a popular concept of research for some time (to name a few [16–18]). Nevertheless, achieving such complexity through control techniques is considerably challenging.

There are many studies on multi-UAV cooperative control systems that address coordination issues. These focus on designing a system to control and monitor a region. One of the earliest studies proposed using aerial photographs to monitor fires in order to combat them [19]. The objective was to use aerial photographs to map the fire and then coordinate the team on the ground. In the past few years, the literature has included more and more research of systems utilizing a team of small cooperating UAVs to get better surveillance; that is, better response time in missions where time is critical.

Recent studies have focused on special missions that can be efficiently performed with multi-UAV systems. Some address the problems of formation flight and some the problems of coordination. Fewer studies have been done on reconfiguring the coordination [20] or on coordination where the assigned tasks have uncertainty. This paper demonstrates that if the guidance system accounts for real-time events and is able to adjust the flight formation to incorporate changes, then the trajectories are more effective than traditional methods.

Closely related is the work that has been done on multi-UAV coordination for tracking missions for search and rescue or surveillance [21]. It presents a concept that relies on low altitude and short endurance (UAVs). The work explores tracking a fire line by using a team of UAVs following the perimeter of the wildfire area. The UAVs return periodically to the ground station location for downloading the collected data. The research focused on how to minimize the latency associated with the fire perimeter measurement when it is transferred to the ground station.

In [22], the design includes a coordination scheme to control a rotary-wing platform (Quadrotor) for a similar mission to the one above. Essentially the motion of the UAV patrols the propagating perimeter. Whenever one UAV approaches another UAV (rendezvous point), deconflict the rendezvous and resolve each UAV next flight direction. That research assumes, however, that the perimeter of the fire is circular. These studies (and similar ones) examine a specific scenario where the focus is on directly tracking the periphery point. This is limited, however, by focusing on the connection between the uncertainty of the spreading perimeter and the maneuverability of the fleet needed to maintain knowledge of the complete perimeter.

1.4. Observation. In coordination, one of the basic operations is observation sharing. Most of the recent studies in multi-UAV address the problem of partial information. It reflects the "real-world" problem where the UAV has limited communications (range or bandwidth). One UAV can communicate with one that is close by, but not with another that is far away. Ref. [23] presents a variety of research problems in which multivehicle systems agree on the value of observed data (consensus), and explores control strategies and a set of solutions for implementing them. Ref. [24] includes a chapter that suggests various deployment algorithms. They consider a distributed algorithm to address the physical limitations of the communication system for observation sharing.

If a coordination algorithm for an environment with uncertainty is available, the overall system still relies on individual sensing capabilities. Even if the system uses the best or most advanced sensors, the sensors can be restricted by environmental conditions, i.e., the sensors carried by the UAV do not have sufficient range [25], and the data measured can only be local and quantized.

The inefficiency of current systems with high-level-control creates significant timing difficulties for achieving the mission objectives. The ongoing mission can leave one vehicle loitering, resulting in a high latency of updates. Based on different studies [21], this represents a large time loss during a mission, with fewer updates, which in some cases can cause the mission to fail in its tracking objective.

1.5. Propagated Periphery Modeling. Disaster growth models, which predict the spatial and temporal dynamic spread rate, may help in evaluating the situation and deciding on a suitable response in a real-time deployment [3]. Appropriate representation and estimation of the spatial uncertainty can improve the prediction or help in developing a simplified model [26]. A mission with an uncertainty model for the AOI stands to benefit substantially from the predicted confidence envelope approach. For example, in expected high rate of spread (RoS) segments along the AOI perimeter, the allocation can use the availability and priority of the segment to get better results than if it were to assume that all segments along the perimeter are identical. Available UAVs can be redirected to new areas instead of merely loitering.

In one of the biggest wildfire research projects done by the *Joint Fire Science Program*, the researchers developed fire behavior models for operational use. Their main objective was to develop a detailed dynamic model to predict the physical behavior of the ground phenomenon. They considered two simple fire modeling approaches. In both models, the assumption was that the local spread at a point on the perimeter is perpendicular to the fire perimeter into an unburned environment and that the fire has a local RoS normal to the fire line.

2. Problem Definition

This research proposes a system design and implementation for quick deployment of a low-cost, low-power fleet of UAVs with a high-level ground control system. The results of this research project introduce the new methods which can serve as high-level-control for operational multiagent systems: a method to estimate a propagated boundary [5] and a scheme

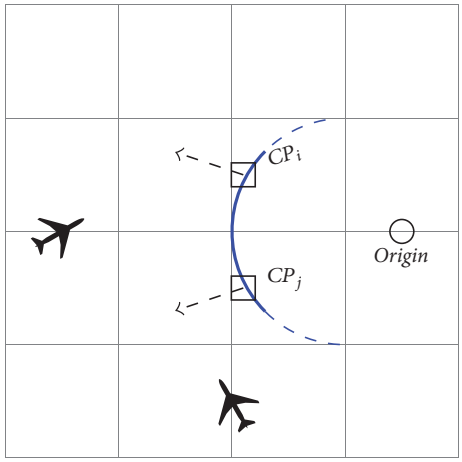


Figure 2: Setup of the boundary representation approach. CP_i and CP_j (square) are two of many grid points representing the close predicted periphery (in blue). The Origin (circle) is the starting point of the propagated phenomenon, and the UAVs are used to collect observations.

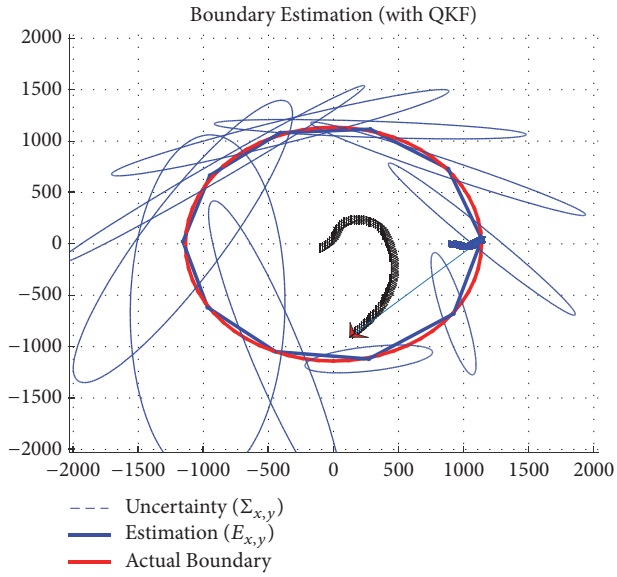


Figure 3: A periphery estimation with a single UAV for an autonomous mission is illustrated. The red line represents the actual periphery, and the blue line represents the estimated one. The UAV flies over the explored area autonomously. The line of sight to one of the CPs illustrates the directional effect of arbitrary CP. The QKF method is employed on all the CPs simultaneously, and the UAVs identify the current highest uncertainty to approach next (originally presented in [5]).

for optimal deployment of a fleet of agents in an exploration mission [27].

The problem is one of optimization with respect to time with sparse measurements detected by a fleet of UAVs. The UAVs have a dynamic process to monitor, as quickly as possible, a periphery represented by a set of Control Points (CPs). The complete system design considers the uncertainty of the bounded phenomenon, where each UAV fleet member carries an on-board sensor to distinguish between inside and outside areas.

Figure 2 illustrates the approach taken to represent the boundary with a set of CPs connected by straight lines. Each CP has a nominal spread rate that is considered relative to the origin point of the propagated phenomenon. That is, the spread rate is always pointed outward. The information is being gathered by a UAV to provide the observations that are noted as *IN* or *OUT* relative to the enclosed periphery. The optimum policy is derived from the decision of which CP the UAV should approach first to reduce uncertainty.

3. Periphery Estimation Methods

The estimation methods approach in this research project relies on an earlier developed technique for estimation of propagated boundary with quantized measurements [5]. The monitoring system involves large numbers of possibly randomly distributed inexpensive sensors, with limited sensing and processing. The estimator incorporates observations gathered by multiple observers and uses the Quantized Kalman Filter (QKF) estimation method [28] to update the expected location and unobserved spread rate. This technique has been extended and laid out the groundwork for the Greedy Uncertainty Suppression (GUS) strategy [27].

The estimation is meaningless in a situation where the available sensors are located inefficiently (e.g., considerably far or colocated). The GUS strategy searches for trajectories that improve the estimated boundary of a propagated phenomenon. The methodology designed for on-line look-ahead approaches to rerouting the UAVs is computationally intractable [8]. To improve the performance, even more, one can use the new approach to further suppress uncertainty. The UAV trajectory is changed to continuously reduce the uncertainty in the biggest covariance among all CPs, by flying directly to the tip of the major axis of that ellipse. Figure 3 illustrates the basic concept for reducing the uncertainty autonomously. Each associated uncertainty is represented as an ellipse (of 95% of confidence area). In previous work, it has been observed that reducing uncertainty is related to measuring distance as well as the arbitrary approach angle direction to an arbitrary analyzed CP. Moreover, uncertainty depends on measurements availability. Hence uncertainty grows with time when no significant observations have been incorporated. The UAV can approach a CP along the direction of its maximal uncertainty axis (direction of major axis of the covariance) and reduce the one-dimensional uncertainty. Interestingly the observed property affecting the uncertainty of CP is the line-of-sight direction.

4. Guidance Logic

4.1. Overview. The trajectory addresses the monitoring problem by adopting common principles. The first is a book-keeping. The mission planner keeps track of representative quantities of interest. Reference [29] suggests discretizing

the target space. Space gridding is typically performed after initial space decomposition which divides the target space into subspaces and later assigns them to different UAVs.

The second fundamental principle is the mission plan. The method considers the UAV deployment stage and preplanned search patterns (e.g., zigzag, spiral, and alternating) in case there is a need to switch back. The third principle is the replanning which adjusts the uncertainty models with observations.

The suggested strategy considers the perpendicular direction of the uncertainty major axis as the next search direction. The periphery is divided into segments based on the number of UAVs and the resolution of the periphery (number of CPs). The planner chooses important CPs which are distributed along the perimeter. Assigning a UAV to one of the CPs will influence the others and reduce the uncertainty accordingly.

Deploying the UAVs is based on the number of available resources. For example, for two UAVs the deployment is to two segments of the predicted polygon where one UAV is assigned to the highest look-ahead uncertainty (weighted by variance and time-to-go) and the second direction is the highest uncertainty on the remaining segments.

Three benefits are gained from that allocation policy: First, the solution avoids flyby trajectories and potential collision. Second, UAVs are not allocated to the same or a close area. Third, the trajectories are being evaluated for dynamic motion feasibility to be carried out by the assigned UAV.

4.2. Greedy Uncertainty Suppression (GUS). GUS strategy tends to minimize the maximum uncertainty over all CPs by incorporating observations over a long period. The policy achieves longer look-ahead with an on-line rerouting logic for the fleet members' task.

The implementation includes two main parts; coordination and allocation. The basic operation leads to coordination involves sharing information of the assigned tasks. The UAVs share their observations with a centralized entity. The observations incorporated sequentially in the estimation process. GUS algorithm is a step by step procedure to determine the best task to each UAV. The notation uses superscript j to label UAV and i as an index of an arbitrary CP.

The first step relies on a previously developed algorithm [5] (QKF). This procedure includes system coordinates transformation, scalar probability evaluation, and Kalman Filter to estimate the state ($\widehat{x}_{x,y}^{CP_i}$) and covariance ($\widehat{P}_{x,y}^{CP_i}$) of the CPs in the original coordinates frame.

The following steps determined the new policy. Step 2, sort the CPs estimation by their major variances. After correcting the state ($\widehat{x}_{x,y}^{CP_i}$) and updating the covariances ($\widehat{P}_{x,y}^{CP_i}$) the procedure continues to evaluate the major-axes of the projected uncertainties. By sorting the variances and adjusting the waypoints along the compass line the algorithm generates candidate destinations.

To evaluate all the alternatives, Step 3, run Dubins Vehicle algorithm which also provides the length of feasible trajectories and evaluates the time-to-go for each UAV. The associated trajectory for UAV_j weighted by the cost function and step 4 assigns the UAV to its best feasible task. The weights of the cost function address the need to consider additional restrictions (for example, deploying the UAVs to one side of the periphery) or tasks.

The propagation of the error between predicted and actual boundary can increase with no control. An additional task assigned where UAV had not cross the actual boundary for a long duration. The policy includes a special allocation mode, rerouting toward the origin point and searches for a crossing point. When that step is done the algorithm is back to the default allocation mode.

5. Monitoring System Architecture

5.1. Background. For many experimental and operational applications, UAVs can enable or enhance the efforts available to researchers or operational teams. Much work has been done to make UAVs useful in a myriad of scenarios. In some scenarios, operating in the environment requires special skills or training that operational teams do not have; here an autonomous system can enable access that was previously difficult to obtain. In recent years, there has been a rapidly increasing interest in UAVs where the operational problem requires an airborne platform.

Technological progress has made it possible to use inexpensive autopilots on small UAVs. The development of high-density batteries, long-range and low-power radios, cheap airframes, high-performance microprocessors, and powerful electrical motors all make experimental research or operational team with UAVs more practical than ever [30]. The availability of UAVs as a fast deployable resource allows teams to explore many new kinds of scenarios such as wildfire. The flexibility of the system design further allows for quick changes, reducing the project workload.

A modern UAV system consists of an on-board control system (i.e., autopilot) and Ground Control Station (GCS). The autopilot utilizes various sensors, communication modules, a power supply unit, and embedded software to control the UAV. The autopilot software is the real-time implementation of the guidance, navigation and control algorithm; one of the demands on designing a rapid prototype testbed is to enable control algorithms, discussed briefly in [31].

Autopilots control and guide the UAVs in flight. They rely on data gathered by various sensors and on a central processing unit (CPU), which carries out the instructions of the program. The objective of an autopilot system is to consistently guide the UAVs to follow reference paths or navigate through several waypoints. A UAV autopilot system is a closed-loop control system consisting of two parts: the state observer and the controller. A typical observer is designed to estimate the state (e.g., attitude) from sensor measurements (gyro, accel); advanced control techniques are used in the UAV autopilot systems to guarantee smooth, desirable trajectory navigation.

This paper focuses on the design of a multi-UAV system that is used in this research project and future projects. The emphasis is on the need for multi-UAV coordination and high-level-control. Ref. [31] provides a review of the existing

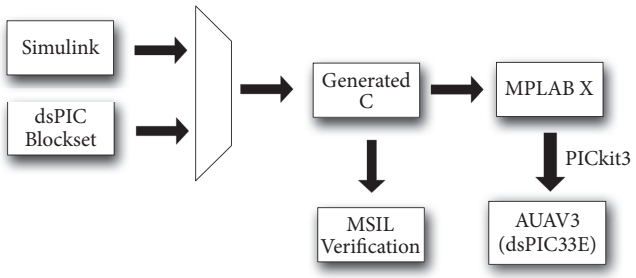


Figure 4: SLUGS II code generation workflow, with a new verification step, Multi-UAV Software in the loop (MSIL).

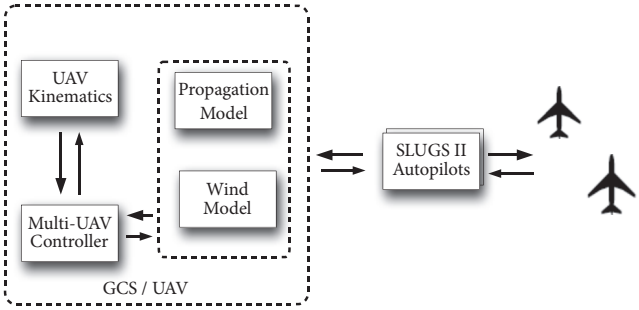


Figure 5: Multi-UAV Monitoring System, software block diagram.

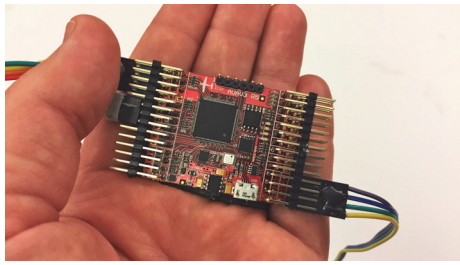


Figure 6: AUAV3 board.

autopilot and the migration process from the previous successful rapid prototyping concept to a new design.

SLUGS (Santa Cruz Low-cost Unmanned Aerial Vehicle Guidance, Navigation & Control System) is a platform that includes autopilot software and hardware components that enable a flexible environment for research in GNC applications [32]. The SLUGS was designed primarily for GNC research, and it has already been used in many flight tests. It is also part of the experimentation with fixed-wing UAV systems, presented in the following section.

The SLUGS II design improves on previous SLUGS design because it provides rapid prototyping control for multi-UAV systems. The control design process is made up of many iterations that can be verified and validated through both simulation in the Simulink environment and with autocode generation.

5.2. Software Design. The complete autopilot algorithm is implemented in Simulink using block diagrams and Matlab toolboxes (MPLAB X - Microchip Integrated Development Environment (IDE), and dsPIC - digital signal controllers support for Embedded Coder). Simulink blocks and Matlab routines are effective software that can be used to modify the algorithm and verify the design. Once the model is updated in the Simulink environment, it then generates the new code with the updated features. The R&D work in a model based environment makes the programming phase easier. Simulink includes tools that automatically generate and compile the code. The code is then deployed directly to the autopilot hardware [33].

SLUGS II modifies the design process to add a verification step for the generated code in a flexible and friendly environment that is committed to the sequence of events in software rather than to guarantee strong real-time performance execution of the code. Figure 4 demonstrates the code generation process and the design validation is discussed in detail in the implementation section.

The software design, as presented in Figure 5, introduces the first order constraints of a dynamical system where the vehicles are mobile and the environment domain changes.

The models are software oriented implementations in which the execution processes guarantees to reconstruct the outcome; hence, the process is deterministic. The UAV model in the software is a mathematical representation of the actual motion of nonholonomic systems.

The propagation model helps anticipate the boundary location in time. The developed model represents the fire-front propagation of a wildland fire. The implementation of this model is a greedy evaluation; each point out of a set of grid points along the periphery is evaluated. Wind velocity and ground slope are incorporated into the propagation model. The model combines the wind velocity and ground slope directly to each grid point of the boundary.

5.3. Hardware Design. The literature on COTS autopilots suggests that the minimum requirements for a research autopilot are robustness and attitude accuracy, enough for low altitude flight surveillance. Hardware must include sensors on-board and software for an attitude solution [34]. This hardware design makes an important contribution to the research framework because it introduces a new design. The SLUGS embedded system features two Microchip dsPIC33F microcontrollers. That design allows SLUGS to implement more complex and effective Guidance, Navigation and Control (GNC) algorithms. It provides a high level of safety and fault tolerance features, and it is designed such that the autopilot system would have more than enough processing power. However, it means more maintenance for the research autopilot Integrated Development Environment (IDE), and increased cost.

SLUGS II simplifies the existing design by using on a reliable commercial-off-the-shelf (COTS) hardware. The AUAV3 is a commercial open-hardware development board (all PCB layouts are provided) [35]. It features a single Microchip dsPIC33EP with twice the clock rate of the dsPIC33F. The AUAV3 board (see Figure 6) comprises peripheral circuits for

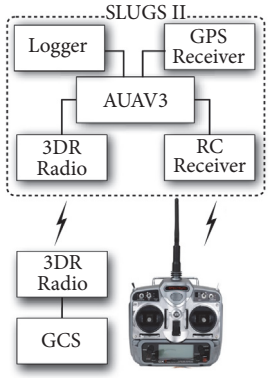


Figure 7: SLUGS II basic components.

IMU, Magnetometer, Barometer and the standard communication interfaces (SPI, CAN, UART and I^2C). Researchers have examined using the AUAV3 to replace the in-house SLUGS hardware. Ref. [31] discusses the SLUGS II Simulink model migration process in more detail, and the steps for verifying and validating the performance with a series of flight tests.

The design of SLUGS II is such that it can be adapted to various different scales. It provides a solution for multiple UAVs in the testing environment when they are needed for research or development. The major challenges of research in multi-UAV are handling duplicate systems with low maintenance cost, reliability, and researchers' insufficient skills. The UAV is linked with continually changing technology so that new infrastructure needs to be assessed and adopted in order to improve the existing system. The system design can help with this maintenance by enhancing the new R&D autopilot environment, eliminating the need to maintain multiple platforms can reduce overhead costs and development difficulties.

Figure 7 presents the basic hardware configuration per any single UAV employed by the monitoring system. The AUAV3 autopilot controls the GPS receiver, telemetry (recorded by a logger and transmitted to the GCS), remote-control (RC) inputs for a pilot safe mode and radio transceivers (3DR Radio) for manual and autonomous flight modes.

6. Monitoring System Implementation

6.1. SLUGS II Components. The AUAV3 addresses the issue of the skills needed to develop or maintain in-house hardware. Commercial hardware is constantly being updated, and for the R&D autopilot, this is an opportunity to put all the efforts into developing GNC algorithms and utilizing low-cost COTS hardware. The old hardware is difficult to integrate with newer sensors and sensing technology. Complex applications require a flexible and adaptive R&D autopilot to keep up with a dynamic environment.

The UAV model integrated within the Simulink development model is another challenging component. It needs specialized skills to tune and adjust to different platforms. Porting the X-Plane simulator improves the development effort and further reduces costs. Different airplane models can be found on the local simulator database instead of tuning the aerodynamic coefficients of a six-degree-of-freedom (6DOF) model by hand.

Two components are migrated as part of SLUGS II design. The benchmark configuration takes the MatrixPilot open-source autopilot and deploys the code on the AUAV3 board. Performance benchmarking ensures that the migrated components perform as well as or better than the old components. The new configuration is then evaluated in multi-UAV software in the loop (MSIL) simulation and in real flight tests.

Once the assessment of the AUAV3 board is completed, the Simulink model is then modified. The model adjusts to the new dsPic configuration. This integration phase includes eliminating the blocks that handle communication between the separate processors, improving the modeling style, optimization, removing dead code, and identifying incompatible porting issues. Configuring the Simulink model to the new AUAV3 board is based on the Microchip dsPIC toolbox (a new revision of the Lubins Blockset [36]). Although the complete process requires significant manual work, the main intellectual property (IP) of the R&D autopilot remains almost untouched.

In the final phase, the newly migrated autopilot is subjected to rigorous testing using test cases applied to the original design (SLUGS) and MatrixPilot [37]. Apart from the functional load testing, testing is carried out to ensure that the necessary performance level is achieved. The migrated autogenerated code is deployed, and parameters are fine-tuned for the new airframe (BixlerII).

6.2. Ground Control Station. The GCS is one of the most important components in a UAV system. It provides an operational interface to monitor and control the assigned task to the multiple UAVs. It presents any additional information that does not require the autopilot to complete its task; however, it supports the user who monitors the mission to coordinate with other systems for better decision making. The GCS includes indications for the mission showing the relevant spatial data (i.e., geodetic coordinates) associated with the map of the area of interest (see Figure 8).

The GCS communicate with the UAVs using a bidirectional data link (X-Bees transceivers). It runs on a mobile laptop computer that can easily be transported to the test site.

A complete process that supports a multi-UAV configuration is needed to be considered by the autopilot system for real-time identification and task allocation. To support a multi-UAV configuration, the SLUGS II design extended the tools for software verification. The multi-UAV IDE offers code verification with complete software in the loop (SIL) simulation.

6.3. Multi-UAV Software in the Loop. MSIL simulation is a higher level of fidelity for the final steps of developing the high-level-controller. MSIL simulation allows running the SLUGS II research autopilot on a computer before running

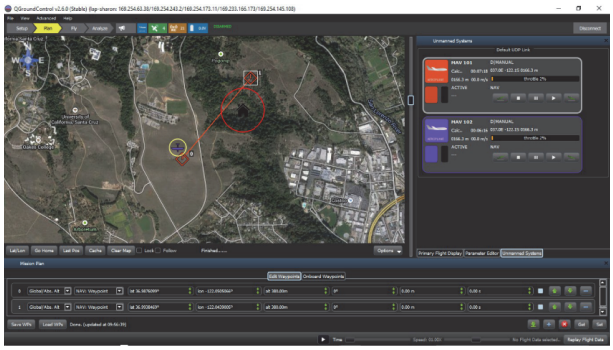


Figure 8: The graphical user interface (GUI) of the GCS is presented. The open-source software (Qt-Ground-Control: QGC) is adopted and extended to support the design of a multi-UAV monitoring system. The software supports the planning and visualization of the UAVs' trajectories in real time.

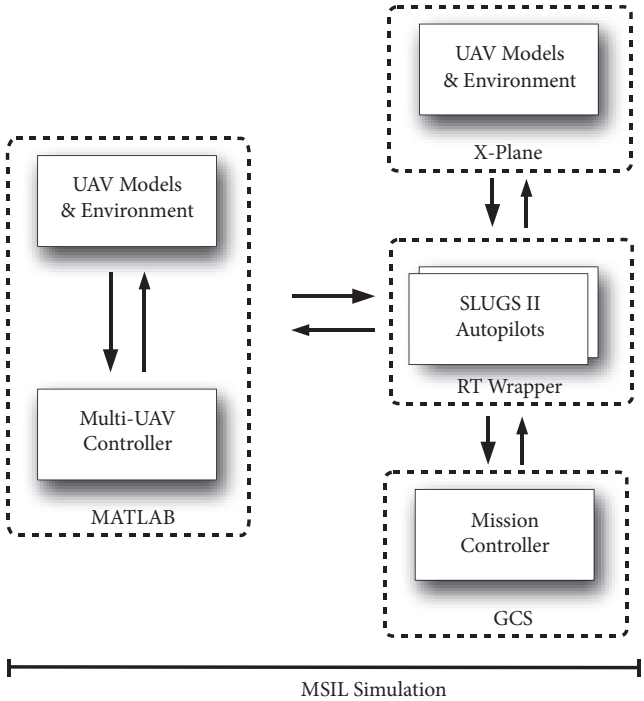


Figure 9: MSIL block diagram.

it on the target processor. It communicates with a simulator for simulating high-fidelity flight dynamics (X-Plane). The MSIL simulation is meant to run a single or multi-UAV configuration and support the external interfaces and built-in internal calls (for example, memory, timing and peripheral libraries) of every instance of the SLUGS II autopilot code.

The MSIL software includes the generated code, which is compiled together with a handling layer (real-time wrapper software). The RT Wrapper interfaces with the external software through a User Datagram Protocol (UDP) socket or a serial port. The MSIL simulation controls the simulated GPS, telemetry and remote-control (RC) inputs for a real RC controller (training mode). The autopilot researcher benefits from the ease of integrating the original generated code and having an easy, friendly environment for debugging.

The GCS unit controls the UAVs through a communication bridge to ensure two-way communication between the GCS and the SLUGS II autopilot. The autopilot can directly manage information from the serial port (or in case of MSIL from the buffer of the serial port). The RT Wrapper (in Figure 9) is responsible for managing the buffers and for distributing the MAVlink messages between real UAVs or simulated modules.

The coordination algorithm is executed in Matlab and works as an extension of the GCS. The RT Wrapper creates a tunnel between Matlab and the SLUGS II software through a physical communication link (UDP) using the MAVlink protocol.

6.4. Multi-UAV Hardware in the Loop. The Multi-UAV hardware In the Loop (MHIL) simulation runs the SLUGS II software stack on the AUAV3 flight controller using raw sensor data fed in from the simulated environment running on the desktop PC. HIL simulation replaces the UAV and the environment with a simulator (the simulator has a high-fidelity aircraft dynamics model and environment model for wind, turbulence, etc.) The physical autopilot hardware (AUAV3) is configured exactly as for flight and connects to a computer running the simulator rather than the aircraft. In this sense, the AUAV3 does not know it is flying a simulation.

Figure 10 shows the MHIL setup. The involved units in the MHIL configuration are depicted along with their associated interfaces. The AUAV3 and the GCS are connected physically by a telemetry link. The autopilot is connected to a computer running the simulator. The simulator is fed by the servo commands and responds with sensory values from the simulated airplane model. The generated sensor values are similar to the IMU output and injected to the navigation algorithm as the UAV autopilot flies the high-fidelity flight situation.

In the end, all of the various functionalities must work both as individual subsystems, but also integrated as part of the entire system: experiment with the UAV design, the basic multi-UAV flight formation, and the monitoring system control. Each one is a step in validating the complete system design which addresses the full multi-UAV monitoring problem.

The system architecture can be utilized in a centralized or a decentralized scheme of operation to enable coordination and information sharing. In a centralized system configuration, the UAVs relay real-time information between each other through the GCS. Alternatively, the UAVs could transmit real-time information between group members (a decentralized scheme configuration).

7. Simulation Results

7.1. Periphery Estimation Evaluation. The simulation is designed to evaluate all major components which involve

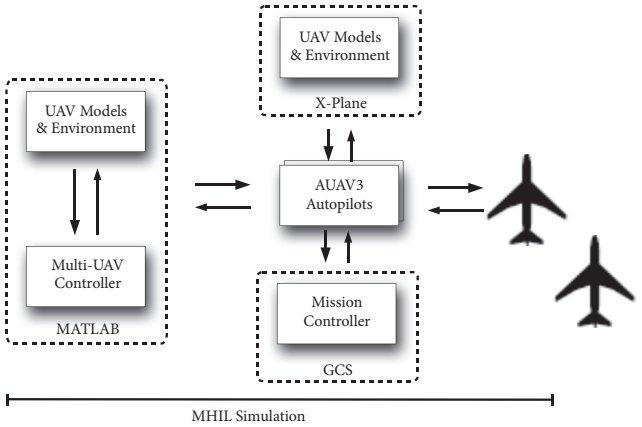


Figure 10: MHIL block diagram.

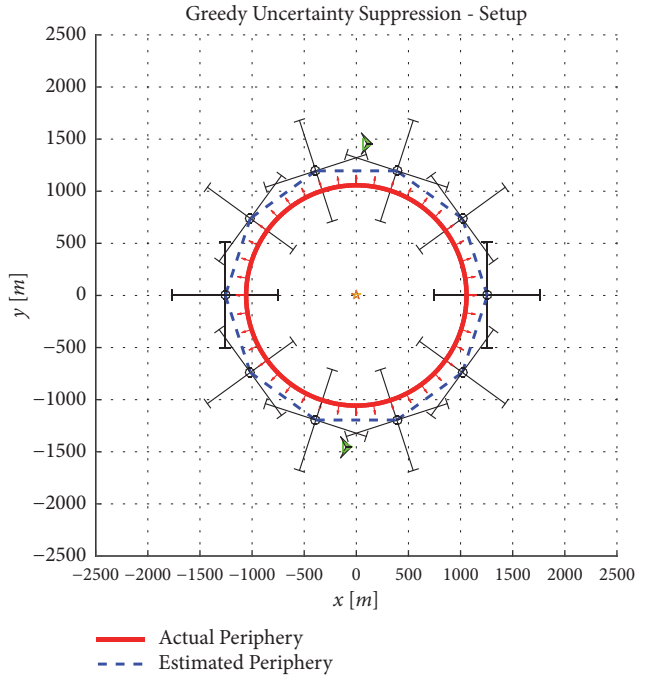


Figure 11: Initial setup. The UAVs are at the final stage of the deployment phase and located on the opposite sides of the boundaries. The actual periphery is a solid red line, and the predicted periphery is a dashed blue line. The error bar associated with an arbitrary CP represents its current perpendicular uncertainty (1σ). Note that the error bar is equal and results in a predetermined prediction that is based on a maximal spread rate.

in the GUS strategy. The environment conditions are being simulated based on a model of a propagated wildfire with a random and bounded spread rate (3 ± 0.1 $[m/sec]$). UAVs allocation is being implemented in a separate component which incorporates the observations gathered by the simulated UAVs.

The UAV dynamics model subject to constant-speed of $20[m/sec]$ which is the approximated speed of the platform that has been developed and examined for the experimental stage of this research. Moreover, the centralized controller comprises the QKF estimator which fuses the observations, and it is based on a previously derived technique. The following simulated scenarios explored the efficiency of the suggested concept.

The initial setup attempts to adhere to the real problem, and therefore a real-time data received from the CAL FIRE (San Mateo Santa Cruz Unit for the Martin Incident) is used. For example, the initial AOI is large ($1Km \times 1Km$) and the time scale is long (i.e., hours). Figure 11 shows the actual periphery with two UAVs deployed from both sides of it.

The propagation model used in the simulation is simplified. However, it allows investigating the major properties of fire spreading. The dynamic expansion of the boundary, the environmental effects (i.e., wind and slope) and the feasibility are all considered in the implementation and are utilized for different scenarios.

Figure 12 demonstrates the scenario with running GUS. A local error bar represents the uncertainty of each CP. The size of an error bar is correlated with the size of the perpendicular and tangential variances.

The performance measure offered in [5] accounts for two performance indicators: errors and uncertainty. The errors indicator comprised the mean-square-error, where the errors are between the predicted and the actual periphery. The uncertainty indicator is simply taking the mean of the CPs' major variances. Both indicators are weighted equally in the combined performance measure, $\sqrt{\sum_i^N (err_{CP_i}^2 + \sigma_{CP_i}^2)}$. In the figures, uncertainty is represented by an error bar in the global coordinates system, and the performances are evaluated relative to the perpendicular component of the local predicted periphery.

Figure 13 shows the combined performance measure with its two performance indicators.

If there were no errors and no uncertainty, then the traditional periphery tracking was an optimal approach. In practice, the uncertainty grows with time, and although errors are reduced to a minimum when the UAV crosses the CP, the spread rate is not observable, and the errors continue to grow shortly after updating the location with the nearby CP. The resulting trajectories and performance improve the benchmark strategy results (see [27]). Figure 14 demonstrates that the GUS and the benchmark have very different performance for a scenario with a wind. The GUS reduces the uncertainty much more over the time of the mission.

8. SLUGS II Validation

8.1. System Configuration. The platform used for the first flight tests was a Phoenix R/C aircraft; later the platform was changed to a Hobby King Bixler 2 (demonstrated in Figure 15). Both of the planes are low-cost foam kits and have a flying weight of approximately 2 lbs. The Phoenix and the Bixler 2 both feature a pusher propeller configuration that reduces vibration and increases overall robustness for a belly landing (neither aircraft has landing gear). The wings

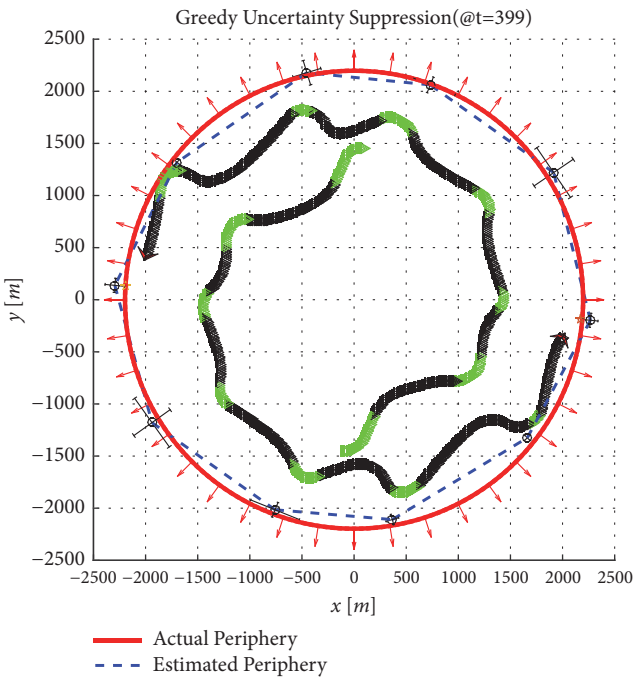


Figure 12: Estimation and coordination with the GUS method. The UAVs switched from the deployment phase to track the highest uncertainties. The actual periphery is a solid red line, and the predicted periphery a blue dashed line. The UAV trail is in green where the UAV is OUT and in black where the UAV is IN. The error bar associated with each CP represents its current uncertainty. Note that the error bar decreases as the UAV approaches a CP and that the observations cause the directional uncertainty of the other CPs to decrease.

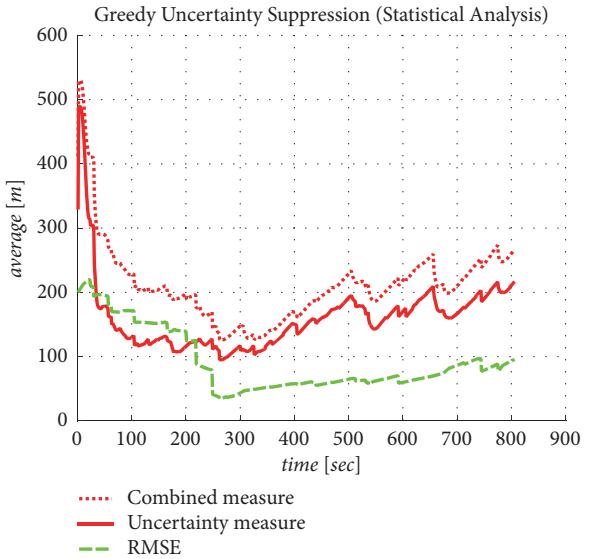


Figure 13: Performance analysis. The solid red line represents the average perpendicular standard deviation, the dashed green line shows the cumulative root mean squared error, and the dashed red line is the combined performance measure. Note that the mean value of the uncertainty is reduced during the mission, and the error increases as the periphery evolves since the number of crosses per AOI get smaller.

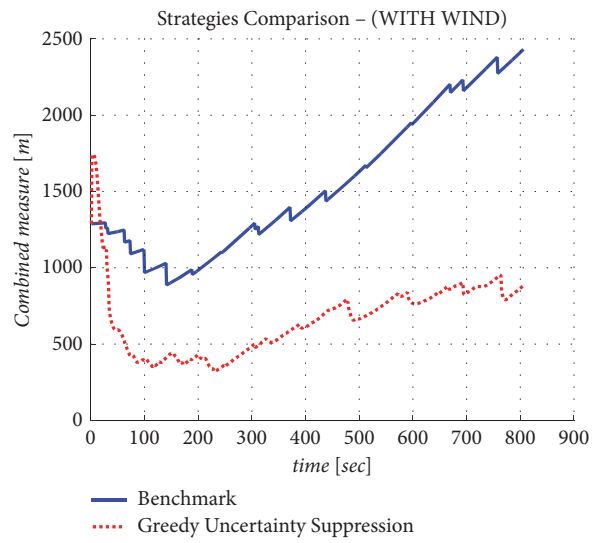


Figure 14: A comparison of strategies with a southwest wind. The solid blue line and the dotted red line represent the combined RMSE performance measure over time for the benchmark and the GUS strategies accordingly.

Figure 15: RC model plane: Hobby King Bixler 2.

and fuselage are reinforced with carbon fiber tubes that provide ample rigidity to the airframe [38]. The aircraft is hand launched for take-off. The Bixler 2 wings are almost an elliptical platform with curved winglets for increased flight efficiency. The power plant for the Bixler 2 aircraft is a 1200 kV brushless DC electric motor. The power source used is a 2200mAh Lithium Polymer battery. This battery provides sufficient current for the electric motor, servo, and the AUAV3 autopilot board, through the Electronic Speed Controller (ESC). The ESC provides a 5.0 volt supply to the servos and the AUAV3 autopilot through the Battery Eliminator Circuit (BEC) and also provides a control signal and power to the brushless motor. The BEC is designed to keep servos R/C receiver running while the baking has dropped too far in voltage to power the motor. The SLUGS II autopilot, like most other autopilots, uses a Proportional-Integral-Derivative (PID) control method for the low-level control loops [32]. The flight controller is developed as a Simulink model, and although it is relatively easy to alter its structure, it requires extensive knowledge about the inner and

Figure 16: Experiment hardware is shown. On the right, two airplanes model that have been used during the field test. On the left, the GCS deployed on the field.

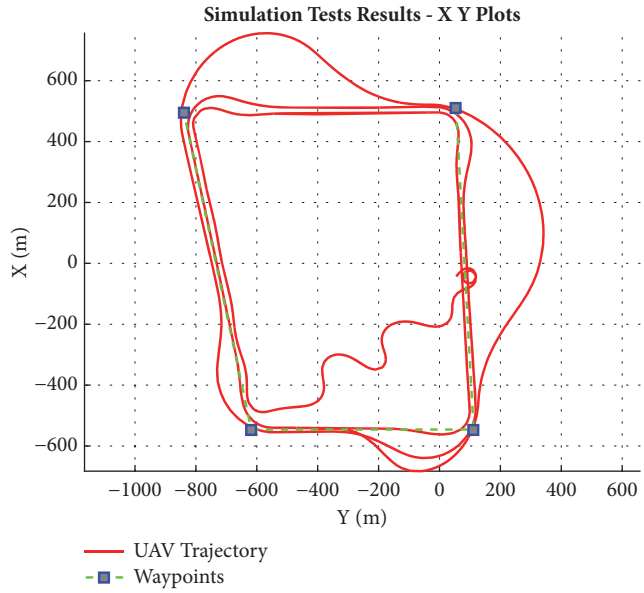


Figure 17: Simulated scenario with a single UAV is presented. The UAV trajectory is in X Y Cartesian coordinate frame and is relative to the Home position. The first segment of the trajectory started from take-off controlled manually by the safety-pilot (RC) and switched to autonomous mode after 23 seconds. Three laps were tested with different PID gains for tuning the roll command.

outer loop structure to redesign the controls. The simulation tests were devoted to validating the viability of the flight controller as flyable. This part of the testing covers the tuning process of the PID gains for the various autopilot control loops.

Figure 16 shows the outcome of the integration with the basic real-time components; autopilot, the UAV platforms, and the GCS a moment before performing a field test.

8.2. Flight Test. The goal of the SLUGS II validation is to support the R&D monitoring system development. The validation relies on several factors, including flight controller and path following performances. The flight controller has been extensively tested within the simulation. The environment supports parameter tuning which can accommodate hardware changes and flight mode extensions.

The most important feature of the SLUGS II autopilot for the R&D monitoring system is its autonomous waypoint navigation capabilities. The ground operator, through the GCS interface, can specify a sequence of waypoints to define the path the vehicle should follow. Figure 17 describes an example of a running scenario with four waypoints and shows how the vehicle follows the desired path while tuning PID gain parameters. Figure 17 shows that initially, the gains were too low, and the system had a slow response.

9. Conclusions

In summary, this paper has presented the design for monitoring system with a core methodology for coordinating a fleet of UAVs to suppress the uncertainty of a generic ground phenomenon. The coordination technique integrated with a R&D monitoring system which was designed carefully to improve the estimation of a propagated periphery supports decision making in an operational scenario.

The system design comprises the major components of a R&D monitoring system: high-level-controller, flight control, and ground control. The development process of the UAV flight controller (autopilot) has been improved with COTS board and a new development environment for software validation. The SLUGS II autopilot obtains the same functionality in the migrated Simulink model as found in the original model. The generated code uses on average a 60% CPU loading. The reserve computation time leaves enough computational resources for further enhancement and evolution.

MSIL simulation tests the generated code in a flexible and friendly environment that is committed to the sequence of events in the software rather than to guarantee strong real-time performance execution of the code. The system is designed to be agnostic as to the type of phenomenon that is being tracked and can be made to work well for a number of different scenarios.

Wildfire incidents are an example of a stochastic phenomenon, and knowing the fire boundary with high certainty would improve decision making by the ground team.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] National Interagency Fire Center. Federal Firefighting Costs (Suppression only). page 1987, 2016.

[2] Fire. NIFC, "Initial Attack," Tech. Rep., 2016.

[3] a Nalyses P Art, Forest Service, Martin E Alexander, David a Thomas, and Dale Bosworth. Fire Management W ILDLAND F IRE S TUDIES AND. Management, 63(3):1–96, 2003.

[4] NIFC, "National Interagency Coordination Center Wildland Fire Summary and Statistics," Technical report, 2013.

[5] S. Rabinovich, R. E. Curry, and G. H. Elkaim, "A methodology for estimation of ground phenomena propagation," in *Proceedings of the 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 1239–1244, Monterey, CA, April 2018.

[6] UAV Over-the-Horizon Disaster Management Demonstration Projects Project Manager: Steve Wegener February 2000 Contents. (February), 2000.

[7] C. Yuan, Y. Zhang, and Z. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Canadian Journal of Forest Research*, vol. 45, no. 7, pp. 783–792, 2015.

[8] N. Nigam, "The Multiple Unmanned Air Vehicle Persistent Surveillance Problem: A Review," *Machines*, vol. 2, no. 1, pp. 13–72, 2014.

[9] X. Cui, T. Hardin, R. K. Ragade, and A. S. Elmaghraby, "A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization," in *Proceedings of the 2004 IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 194–203, USA, October 2004.

[10] G. Kowadlo and R. A. Russell, "Robot odor localization: a taxonomy and survey," *International Journal of Robotics Research*, vol. 27, no. 8, pp. 869–894, 2008.

[11] B. Ristic, D. Angley, B. Moran, and J. L. Palmer, "Autonomous multi-robot search for a hazardous source in a turbulent environment," *Sensors*, vol. 17, no. 4, 2017.

[12] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 47–53, ACM, May 1998.

[13] W. Burgard, M. Moors, and F. Schneider, "Collaborative Exploration of Unknown Environments with Teams of Mobile Robots," in *Advances in Plan-Based Control of Robotic Agents*, vol. 2466 of *Lecture Notes in Computer Science*, pp. 52–70, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[14] S. Russell and P. Norvig, *Artificial Intelligence A Modern Approach*, 2013.

[15] R. Elaine, K. Knight, and B. Shivashankar, *Artificial Intelligence-Rich-Knight-Nair.pdf*, 2009.

[16] M. T. Rashid, M. Frasca, A. A. Ali, R. S. Ali, L. Fortuna, and M. G. Xibilia, "Artemia swarm dynamics and path tracking," *Nonlinear Dynamics*, vol. 68, no. 4, pp. 555–563, 2012.

[17] M. T. Rashid, A. A. Ali, R. S. Ali, L. Fortuna, M. Frasca, and M. G. Xibilia, "Wireless underwater mobile robot system based on ZigBee," in *Proceedings of the 2012 International Conference on Future Communication Networks, ICFCN 2012*, pp. 117–122, Iraq, April 2012.

[18] A. A. Ali, L. Fortuna, M. Frasca, M. T. Rashid, and M. G. Xibilia, "Complexity in a population of Artemia," *Chaos, Solitons & Fractals*, vol. 44, no. 4-5, pp. 306–316, 2011.

[19] K. Arnold, "Uses of aerial photographs in control of forest fires," *Journal of Forestry*, vol. 49, pp. 26–31, 1951.

[20] A. Richards, J. Bellingham, M. Tillerson, and J. How, "Coordination and control of multiple UAVs," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit 2002*, USA, August 2002.

[21] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. Mc lain, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.

[22] K. Alexis, G. Nikolakopoulos, A. Tzes, and L. Dritsas, "Coordination of Helicopter UAVs for Aerial Forest-Fire Surveillance," *Applications of Intelligent Control to Engineering Systems*, vol. 39, Article ID 169193, pp. 169–193, 2009.

[23] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*, 2008.

[24] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*, 2009.

[25] N. Dadkhah and B. Mettler, "Survey of motion planning literature in the presence of uncertainty: Considerations for UAV guidance," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 65(1-4), pp. 233–246, 2012.

[26] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[27] S. Rabinovich, E. Renwick, and H. Gabriel Elkaim, "Multiple Unmanned Air Vehicle Coordination for Monitoring of Ground Phenomena Propagation," in *Proceedings of the ION GNSS+ 2018. IEEE*, 2018.

[28] C. E. Renwick, *Estimation and control with quantized measurements*, MIT press, 1970.

[29] A. Girard, A. Howell, and J. Hedrick, "Border patrol and surveillance missions using multiple unmanned air vehicles," in *Proceedings of the 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, pp. 620–625 Vol.1, Nassau, Bahamas, December 2004.

[30] H. Chao, Y. Cao, and Y. Chen, "Autopilots for small unmanned aerial vehicles: A survey," *International Journal of Control, Automation, and Systems*, vol. 8, no. 1, pp. 36–44, 2010.

[31] S. Rabinovich, *Multi-UAV Coordination for Uncertainty Suppression of Natural Disasters*, Phd, UC Santa Cruz, 2018.

[32] I. Mariano and Lizarraga F., *DEsign, Implementation and Flight Verification of A Versatile and Rapidly Reconfigurable Uav Gnc Research Platform*, 2009.

[33] E. Geaney, UC Santa Cruz Electronic Theses and Dissertations.

[34] R. Garcia and L. Barnes, "Multi-UAV simulator utilizing x-plane," *Journal of Intelligent & Robotic Systems*, vol. 57, no. 1–4, pp. 393–406, 2010.

[35] Nick Arsov. AUAV3.

[36] Lubin. Kerhuel, Simulink - embedded target for pic.

[37] MatrixPilot. MatrixPilot Auto-Pilot, 2016.

[38] P. Kumar, J. E. Steck, and S. G. Hagerott, "System identification, HIL and flight testing of an adaptive controllerona small scale unmanned aircraft," in *Proceedings of the AIAA Modeling and Simulation Technologies Conference 2015*, USA, January 2015.

Hindawi
Journal of Robotics
Volume 2018, Article ID 7806854, 9 pages
https://doi.org/10.1155/2018/7806854

Research Article

Particle Filter and Finite Impulse Response Filter Fusion and Hector SLAM to Improve the Performance of Robot Positioning

Amin Bassiri,[1] **Mohammadreza Asghari Oskoei**,[2] **and Anahid Basiri**[3]

[1]*Qazvin Islamic Azad University, Iran*
[2]*Allameh Tabataba'i University, Iran*
[3]*University College London, UK*

Correspondence should be addressed to Amin Bassiri; amin.bassiri@gmail.com

Received 13 April 2018; Revised 30 July 2018; Accepted 6 September 2018; Published 11 November 2018

Guest Editor: Ling-Ling Li

Copyright © 2018 Amin Bassiri 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.

Indoor position estimation is essential for navigation; however, it is a challenging task mainly due to the indoor environments' (a) high noise to signal ratio and (b) low sampling rate and (c) sudden changes to the environments. This paper uses a hybrid filter algorithm for the indoor positioning system for robot navigation integrating Particle Filter (PF) algorithm and Finite Impulse Response (FIR) filter algorithm to assure the continuity of the positioning solution. Additionally, the Hector Simultaneous Localisation and Mapping (Hector SLAM) algorithm is used to map the environment and improve the accuracy of the navigation. The paper implements the hybrid algorithm that uses the integrated PF, FIR, and Hector SLAM, using an embedded laser scanner sensor. The hybrid algorithm coupled with Hector SLAM is tested in several scenarios to evaluate the performance of the system, in terms of continuity and accuracy of the position estimation, and compares it with similar systems. The scenarios where the system is tested include reducing the laser sensor readings (low sampling rate), dynamic environments (change in the location of the obstacles), and the kidnapped robot situation. The results show that the system provides a significantly better accuracy and continuity of the position estimation in all scenarios, even in comparison with similar hybrid systems, except where there is a high and constant noise, where the performance of the hybrid filter and the simple PF seems almost the same.

1. Introduction

The positioning technologies have faced several challenges for indoor applications. Reference [1] studied several most widely used positioning technologies, including Wireless Local Area Network (e.g., Wi-Fi) based systems, Bluetooth Low Energy (BLE), Ultra-Wide Band (UWB), Inertial Navigation Systems (INS), Radio Frequency Identification (RFID), and Tactile based systems, from fitness-for-purpose point of view, and concluded that, for almost all of the indoor Location Based Services (LBS) applications, including indoor navigation, none of these stand-alone positioning technologies could yet provide the required level of positional accuracy and continuity.

One of the biggest challenges of indoor localisation is the existence of a relatively high ratio of noise to signal. Also, Non-Line-Of-Sight (NLOS), where the received signal does not traverse the direct path between the receiver and the transmitter, introduces a big challenge for the ranging-based positioning systems. This is a common situation where the positioning technology is being used for indoor applications. To handle some of these issues, up to some degrees, many of the positioning systems use a state estimator for an estimated but accurate position solution in a cluttered and noisy environment [2], such as inside the buildings. The state estimator, which is also called the stochastic filter, is a mathematical algorithm that estimates the state variables of a system from noisy and biased measurements [3].

One of the most widely used state estimators is Kalman filter (KF), which functions particularly well for the linear systems with a Gaussian noise [4]. For indoor localisation, the state-space model is typically nonlinear; therefore, the nonlinear filters such as the Extended (or Enhanced) Kalman filter (EKF) and the Particle Filter (PF) could respond better [2, 3]. The PF, e.g., Monte Carlo Localization (MCL), can provide with a better performance, in comparison with the

EKF, in the highly nonlinear environments [5]. While the EKF requires the initial position and can only solve the relative positioning solutions [2], e.g., for tracking purposes [6], the ability of the PF to function and provide the positioning solutions with no initial (a priori) data allows many applications to apply and use it. In addition, the PF is relatively easier to implement, particularly in comparison with the EKF [7]. However, the PF performance is highly associated with the sample distribution and diversity, which can be an issue in systems that are based on the low rate sampling. The loss of diversity among the samples may result in failing to estimate the state or potentially the large estimation errors [8]. This is referred as the sample impoverishment problem and can happen when the measurement noises are small and/or the number of particles is not enough. To handle this, several attempts have been made and some solutions are provided. They include the Regularized Particle Filter (RPF) [9], Markov Chain Monte Carlo (MCMC) move step [10], and combined/integrated Particle and Kalman filters [11]. Despite these, the enhanced versions of the PF may still not be able to prevent and/or cop with the sample impoverishment completely. In addition, there may a compromised performance for the PF, and in the extreme cases, the filter can fail to function completely [8]. Such extreme cases can include the situations that have got a very small measurement noise or a very low number of particles [12]. There are various preventive methods against sample impoverishment and PF failures. However, they concluded that an effective and general remedy to cure a completely failed (or diverging) PF has yet to be proposed [13].

This paper applies a hybrid filter that integrates PF and Finite Impulse Response filtering (FIR), which has been proposed by [13], and couples it with a Hector SLAM algorithm to improve the performance, i.e., accuracy, reliability, and continuity of the localisation under harsh conditions. This system is tested using a laser scanner sensor embedded in a robot to measure the performance of the system in different challenging scenarios and compare the results with the systems that use (a) PF/FIR only [13], and (b) Particle Filter only. While laser scanner can be considered as a standalone positioning technology, the hybrid filter uses a PF as the core main filter, integrated and enhanced by another robust FIR filter, based on the proposed scheme by [13]. The PF estimates the state of the robot in the 'normal' condition, i.e., where there is no sample impoverishment or positioning solution failure. Once an enormous estimation error is detected, the FIR filter starts functioning and helps the PF to recover. Note the hybrid system does not enhance the PF and so it can be integrated into currently developed and deployed PF-based systems. The novelty of the paper is to use this integrated PF-FIR system [13] with a Hector SLAM for simultaneously localisation and mapping purposes. Also, this paper uses Manhattan distance for PF failure detection as it seems more compatible with the grid-based environment where the experiments are conducted. So, the system uses a hybrid filter to integrate PF and FIR, which prevent the PF failure, while a laser scanner embedded on a robot is used to apply Hector SLAM and improve the continuity, usability of the system at different scenarios. This will be examined through some experiments where the system is tested with different sampling rates and noises levels imposed to the system and/or the environments.

The used hybrid filter coupled with Hector SLAM is implemented and several experiments and tests are conducted in both simulation and real-world environments in different scenarios, including environments with dynamic/changing obstacles and at different levels of noise. The proposed system also compared against (a) PF only and (b) the hybrid only systems. The results of the experiments show that the hybrid system coupled with Hector SLAM improves the continuity and reliability of the positioning solutions, particularly in harsh scenarios where the number of particles and/or level of measurement noise may be low. Also, it seems to function with better robustness in the case of extreme noise in comparison with FP/FIR, where usually PF fails to function. Also, the tests show that the used hybrid filters (both with and without Hector SLAM) continue functioning and so are able to solve the kidnapped robot situations.

This paper is organised as follows: the next section is about the mathematical background of the hybrid filter, explaining the principles of the PF and FIR separately. Then the hybrid filter is explained in detail. Then the implementations of the used hybrid filter and the experiments at different scenarios are discussed. And finally, there is a conclusion and a forward-looking discussion on future work.

2. System Design

In order to estimate the state of a system, one of the most widely used approaches is Bayesian filter framework, which uses the observed values and the corresponding confidence coefficients, i.e., the covariance matrix, to estimate the state of the system [14]. For many linear variables, the Kalman filter can accurately estimate the state using a Gaussian distribution [2]. However, for nonlinear variables, the KF may not be able to provide an accurate estimation and so an enhancement is needed. The EKF algorithm is used for the linear approximation of a nonlinear system using the Gaussian distribution [2]. In contrast, the Particle Filter estimates the state of the nonlinear (non-Gaussian) systems using a set of the particles distributed in the state space [14, 15]. So, the Particle Filter provides a numerical approximation for the nonlinear problem [16].

While the Particle Filter can approximate the nonlinear systems numerically, PF may fail due to a wide range of reasons [3, 8] and so resetting the failed PF is important to have a continuous state estimation. Resetting the failed PF also means generating new particles. To generate new particles, this paper uses Finite Impulse Response.

FIR has been applied by several positioning systems, particularly as an alternative to the filters with Infinite Impulse Response structures such as Kalman filter [13, 17–21]. FIR can improve the overall performance of such systems [18–20] as it provides with a robust response despite the availability of noise and/or model parameter uncertainty [18]. While the PF is basically an Infinite Impulse Response (IIR) filter, which can potentially provide a more accurate position

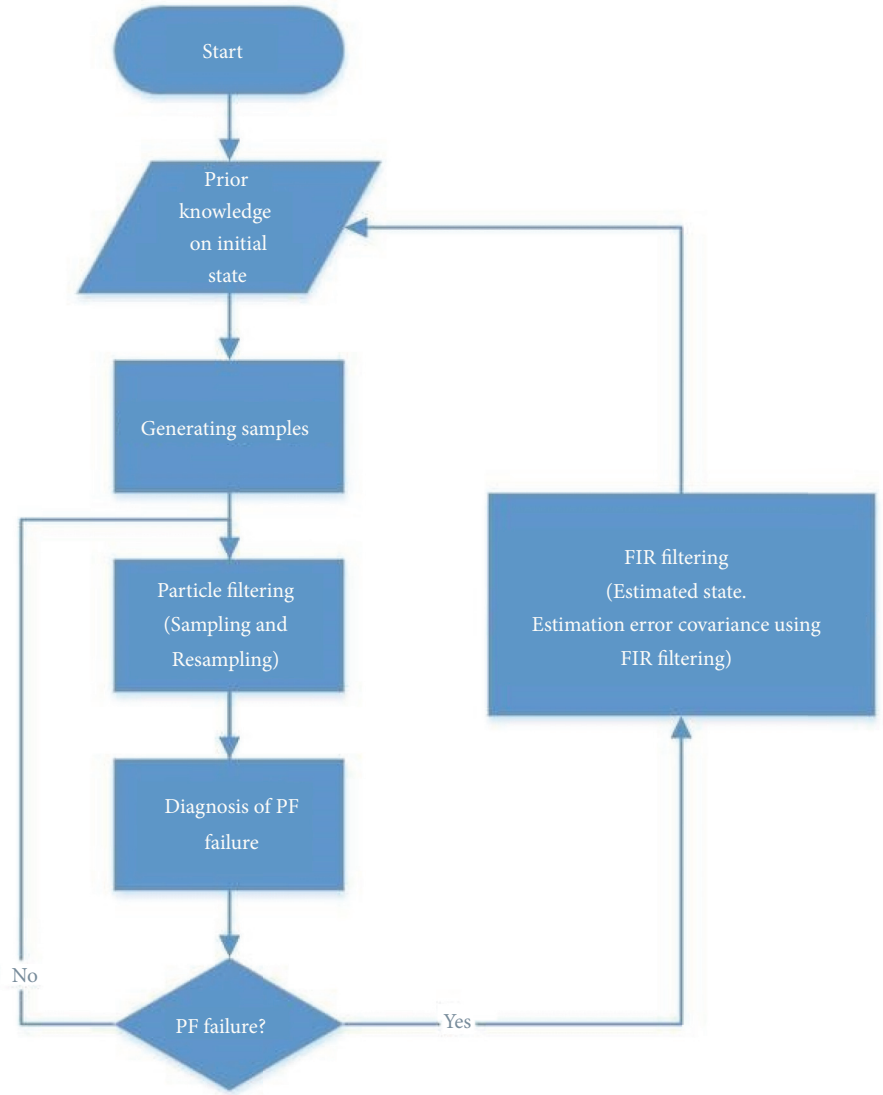


Figure 1: Flowchart of hybrid particle/FIR filtering algorithm [13].

estimation than FIR filters generally, it may fail to estimate the state if there is the case of sample impoverishment. To keep the balance hybrid filter (PF and FIR) may provide a good solution. The hybrid filter allows overcoming the issues and limitations of either the stand-alone versions of the PF and the FIR filters [13]. This paper uses the hybrid filter, combining the PF and the FIR, to provide a reliable, continuous, and robust position estimation. The hybrid filter acts like a PF as long as it can estimate the state. As soon as an extreme noise or any other circumstances that could fail the functionality of the PF, the FIR filter takes over and assures the continuity of the position estimation. However, this paper couples the hybrid filter with a Hector SLAM for even more continuity and autonomy of the whole system.

The hybrid filter uses the PF as its main filter in normal conditions, and as soon as any abnormality occurs it uses FIR to recover the whole system. The hybrid filter, integrating PF and FIR, was initially proposed by [13] and its general process is represented in Figure 1. As the flowchart (Figure 1) illustrates, having been initialised and calibrated, the PF starts functioning. As long as there is no failure diagnosed in the PF, the system relies solely on the PF. This means the relative position of the robot, which can be used for calculation of heading and absolute position, is estimated by the PF. This is mainly because the PF can provide the system, as a whole, with a better performance in comparison with the hybrid filter [13]. However, in the case of PF failure, the FIR takes over and estimates the state and error. FIR is practically an auxiliary or a backup filter that only acts as a plan B, i.e., when the PF fails to estimate the state. The estimation of the state and error from the FIR are fed into the 'initialisation' phase and based on these a secondary initial sample set is generated.

In order to identify a failure in the PF, a diagnosis algorithm which uses the Manhattan distance is proposed and used by this paper. This is basically because the maps

are available as a raster file, i.e., Pixels and grids with values, where Manhattan distance seems more compatible choice [22, 23].

2.1. The Failure Diagnosis. The PF failure can be caused by several reasons; one of the most common reasons behind it is the sample impoverishment, which is also a relatively difficult case to handle [24]. This paper used an algorithm that diagnoses the failure if either (a) the majority of the predicted states fall outside the uncertainty ellipse or (b) the distance between the prediction and the actual samples is too big. These two categories of the PF failure symptoms can be associated with the concepts of accuracy and bias, respectively.

The first category uses the concept of uncertainty ellipse; this paper uses Manhattan distance for the outlier detection. While Manhattan distance is relatively simple to implement, it is powerful way to detect the outliers. It is particularly compatible with the grid/pixel-based input data, i.e., raster maps and spatiotemporal intervals for location updates. The points, which fall outside the 99% confidence interval (3 standard deviations from the average of the distance between the predicted points and the actually measured locations) can be identified as outliers. After identification of outlier points and recognising the PF failure, FIR takes over.

One of the important inputs for robot navigation is the map of the environments. In some cases, the maps can be extracted from the building plans while in other cases the map plan is being generated simultaneously while the rover is moving and sensing. In such cases, the autonomous moving object, here the robot, must be able to both estimate the position and also create the map. While the position estimation itself requires a map and for mapping, the localisation is essential too. In this regard, this paper uses the simultaneous approach; i.e., the processes of mapping and the robot position estimation are conducted simultaneously. This is called Simultaneous Localization and Mapping [25]. SLAM uses the correlation between the estimated robot position and the 'as-built' or the under-construction map [25, 26]. The map, generated while navigating the robot, is fed into the system, recursively. The implementation of the whole system is explained in the next section.

3. Implementations

In this section, the implementation of the system, i.e., the hybrid Particle Filter and the Finite Impulse Response filter, using a mobile robot with a laser sensor for Hector SLAM, is described (see Figure 2). The hardware and then the implementation of the system using the Robot Operating System (ROS), which is the software used for the customization and the development of the hybrid filter algorithm, are described in this section.

As shown in Figure 2, a robot with an embedded LIDAR sensor is used to implement and test the position obtained by the hybrid filter algorithm. The operating system is run on the Raspberry Pi3 and an Arduino Uno board to feed and steer the motors. It also has got two 35rpm motors and

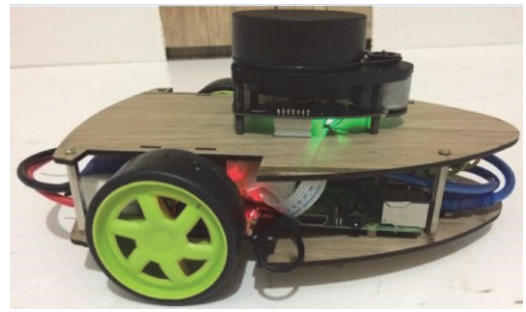


Figure 2: Robot used for implementation and test. Equipped with a RPLIDAR A1 scanner.

one 40Ah drivers. An A1 RPLIDAR scanner is used, which has been the most recent version at the time of development and applied by many new systems. The RPLIDAR A1 scanner provides an angular resolution of 1440 scans per second and a diagnostic length of 6 meters with a measurement range of 360 degrees. The robot controller, Raspberry Pi3, is embedded on an ARM-A7 processor runs on the ROS system under the Ubuntu Linux where all robot software runs. The main purpose of the robot controllers is to calculate the position of the robot and also to send commands to the robot.

The implementation of the used system results in the development of a piece of software, which is in charge of controlling the robot and estimating the position, as shown in Figure 3.

In this paper uses a hybrid filter algorithm for the indoor positioning system for robot navigation using PF and FIR to assure the continuity of the positioning solution, as proposed and implemented by [13]. To improve this hybrid filter, however, this paper uses Manhattan distance for PF failure detection and also couples it with Hector SLAM for more autonomy. First, the map of the environment is created, either manually or using SLAM. In this paper, the initial map is generated manually but it is updated by SLAM. This process is explained in more detailed in the next subsection. The data capture is initiated by the laser scanner. Collected data are sent to the ROS system, which includes a particle filtering module. As it mentioned, in this project ROS is the basis of the customization and development. This is mainly because it is very well developed in terms of most the packages and libraries that are essential or useful for positioning and navigation purposes, e.g., locating. This minimizes the software development and programming phase, as some basic functionalities already exist. Also, it allows sending data using a variety of standardized message formats, which could be useful for any further development and test of the used system by other sensors.

The developed piece of software represented in the ROS node with a publisher name, which is responsible for sending all measured data from the robot. Figure 4 shows the sender (node) of all ROS data.

The LIDAR scanner with a laser scan node captures data and sends it to ROS. The RPLIDAR A1 scanner reads data with the angular resolution of 360-degree omnidirectional

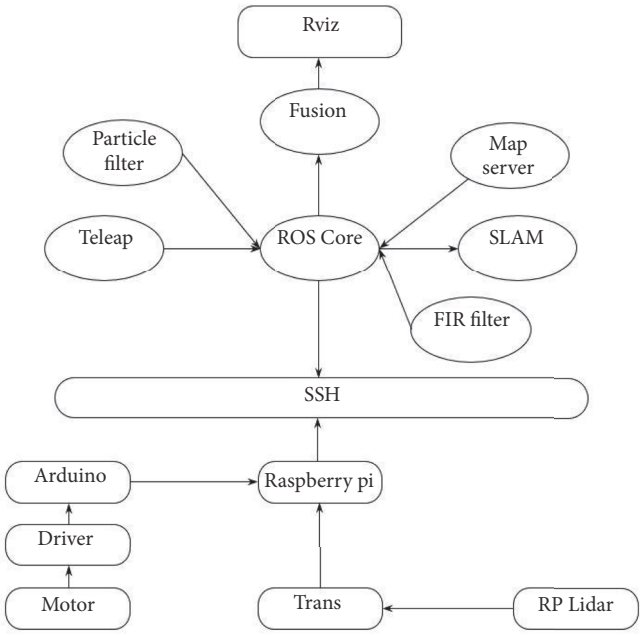


Figure 3: View of the software structure of robot controller.

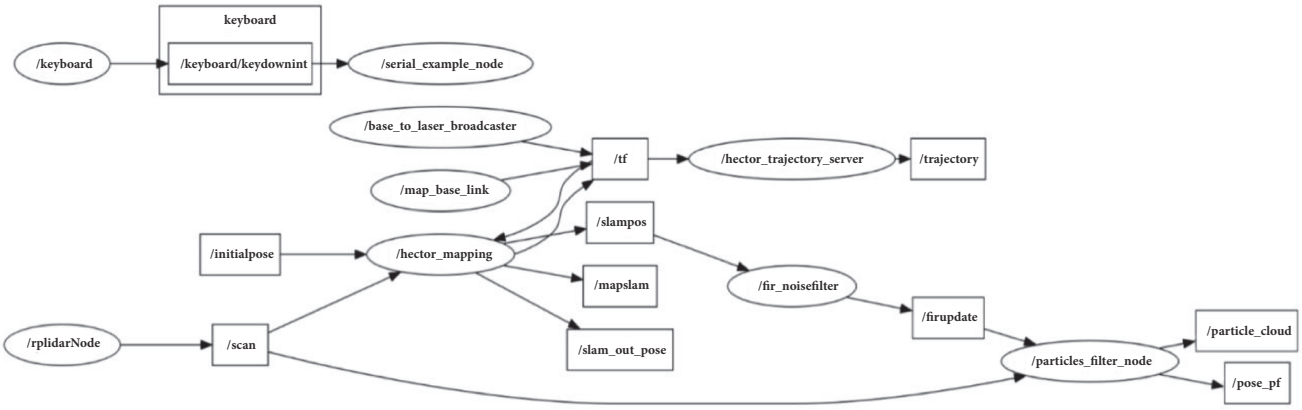


Figure 4: Diagram of all implemented ROS nodes.

and the frequency of 5.5 Hz. However, the software is designed to work with lower resolution and/or frequency at other scenarios.

The physical position of the sensors in relation to other parts of the robot is very important; TF in ROS allows nodes to communicate with each other in a distributed computing environment. ROS packages usually use the system clock as a time source for synchronization. In this project, the system with synchronized SSH clock has a static and fast structure.

3.1. Mapping. As explained earlier, this paper uses the simultaneous approach, i.e., the processes of mapping and the robot position estimation are conducted simultaneously. Hector SLAM algorithm is used to correlate the estimated robot position and the 'as-built' or the under-construction map [26].

To create the map, Hector SLAM modules, which have been made available by the software package, are used at different instances. Hector SLAM functions based on different sensors samples, along with a metadata specifying the number of parameters such as map frames and sensor data format; see Figures 5 and 6.

4. Results

This subsection explains the implementations of the hybrid filter applying a laser scanner data. Different scenarios that may result in the Particle Filter failure are designed and four experiments were conducted. They include the cases examining the performance of the hybrid filter exposed to (a) low sampling of the laser scanner readings, (b) the kidnapped robot, and (c) in a dynamic environment. The results are then compared with both normal situations where

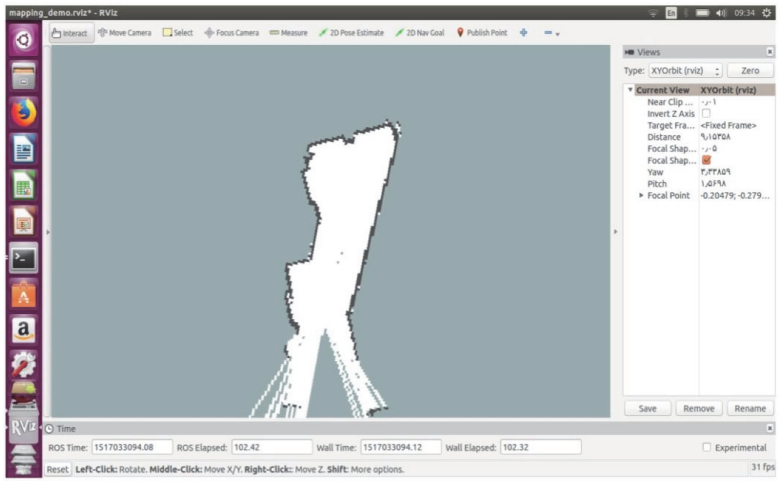


Figure 5: Map Construction by Hector SLAM.

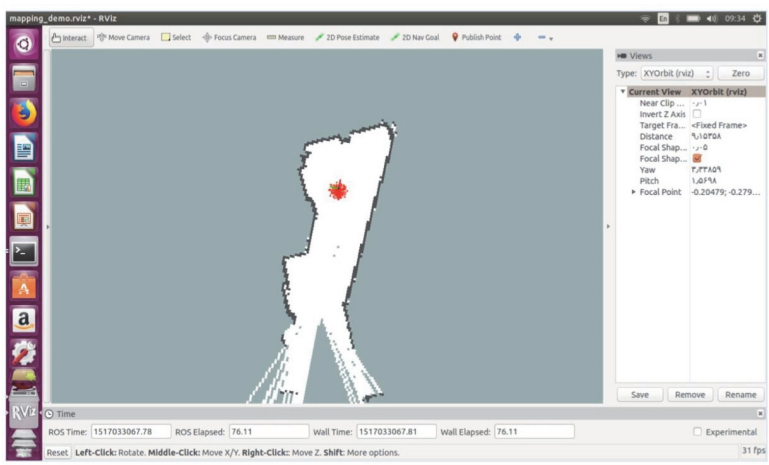


Figure 6: Representation of robot position using hybrid Particle Filter and Finite Impulse Response in Rviz.

the Particle Filter solely can estimate the state with no failure.

To have a benchmark, firstly, the hybrid system run on the robot is tested under normal conditions. The robot moves in a counter clockwise direction in a rectangular trace (1m $*$0.5m). For each scenario, the same planned path is taken by the robot and the error is calculated based on the same formula.

4.1. Robot Position Estimation with Low Samples. In this scenario, the laser scanner sensor measures low samples, i.e., the number of measured points reduced to 360 samples in 360°. Note that, in the normal scenarios, the laser scanner sensor measures 1440 samples in 360° (noise: δ=0.5).

As shown in Figure 7, the Particle Filter provides poor performance due to the low number of sample points, while the hybrid filter offers a lower location error. The estimated path is shown in Figure 8.

4.2. The Kidnapped Robot Problem. The kidnapped robot problem is one of the most challengeable problems in the robot positioning. To handle this, a position estimation algorithm is needed that is able to recover from a high level of error and noise. The kidnapped robot problem is studied in three phases:

(1) The mobile robot starts from the beginning point and moves on the straight line.

(2) After traversing 30cm, the robot suddenly jumps (kidnapped) to 80 cm.

(3) Then continuous to move on the straight line in a similar direction.

Since the sudden jump is not feasible for the robot applied here, while possible in other autonomous rovers such as drones, this scenario is tested in a simulated environment. However, the simulation allows the position estimation algorithm to be tested under any other unpredictable behaviours, including the kidnapped robot problem. Figure 9 shows the trajectory of the movements, i.e., actual path, estimated by PF, and estimated by the hybrid filter.

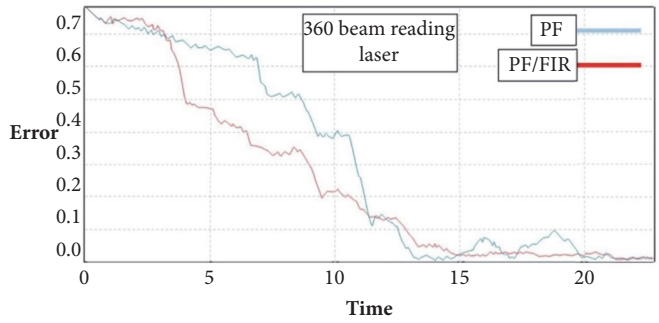


Figure 7: The Particle Filter only versus the hybrid against Particle Filter and Finite Impulse Response (PF/FIR) error with 360 laser reading points.

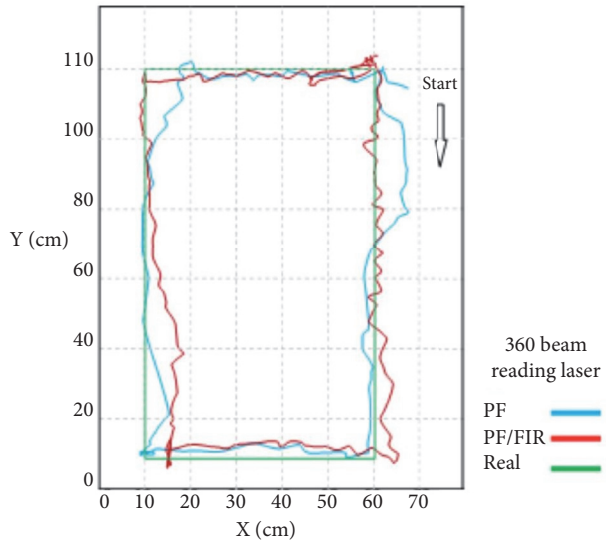


Figure 8: The estimated path in Particle Filter only versus the hybrid (PF/FIR) with 360 laser reading points.

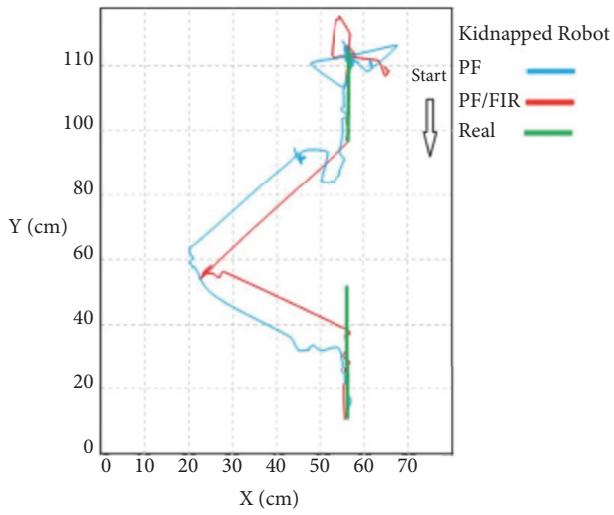


Figure 9: The path based on the estimation of the PF only and the hybrid filter.

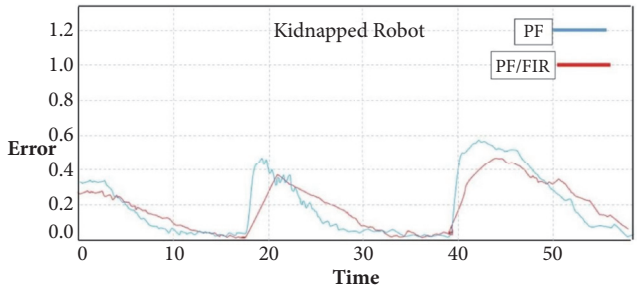


Figure 10: Position errors in the kidnapped robot problem.

Figure 10 illustrates the level of error from the position estimation of the two filters, i.e., PF only and the hybrid (PR/FIR), before and after the kidnapped robot occurrence, i.e., two peaks. As shown in Figure 10, the hybrid PF/FIR filter provides with better accuracy and also continues to track the robot after a shorter transient period.

4.3. Dynamic Environments. The last experiment is to test the performance of the hybrid filter in a dynamic and changing map. Any changes to the robot map would be very difficult to handle for the positioning and tracking system as the most of indoor positioning technologies are based on relative (dead reckoning) localisation and so the change can have a significant impact on the position estimation. In this experiment, the robot moves through the map where an existing map is fed to the system, instead of creating the map on the fly (SLAM). Then a sudden change occurs in the surroundings of the robot. This experiment, i.e., the change in the environment map, is conducted for each of filters, and the position estimation error is measured. As shown in Figure 11 the hybrid filter provides with a better accuracy compared to the Particle Filter.

As discussed above, in the three scenarios of a lower sampling rate, kidnapped robot, and dynamic environment, the position estimation of the robot with an embedded laser scanner sensor has got a lower level of overall error. Table 1 shows the improvement (percentage) of the overall accuracy of the hybrid filters used with respect to the PF only estimation.

Figure 12 compares the average accuracy of the PF only, hybrid filter (without Hector SLAM) and the proposed

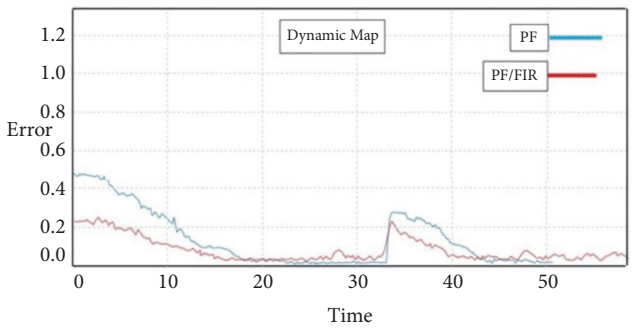


Figure 11: Position errors of PF and hybrid (PF/FIR) filters in a dynamic environment.

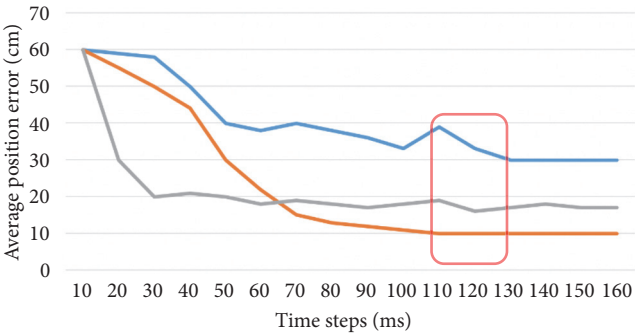


Figure 12: Average error for PF only (blue), hybrid filter only (grey), and hybrid filter coupled with Hector SLAM (orange).

Table 1: Decrease or increase of the hybrid PR/FIR filter in relation to PF only scenarios.

Experiment (scenario)	Improvement
Kidnapped Robot	14.56%
Dynamic map	12.33%
360 beam reading laser	11.8%

system which couples the hybrid (PF and FIR) with Hector SLAM. As it is shown in Figure 12, the proposed system provides an overall higher accuracy and better robustness (particularly with high noise where both PF only and hybrid filter-only systems have got a peak, see the transparent box).

5. Conclusion

This paper used a hybrid (PF/FIR) algorithm for robot positioning in harsh environments, where there are more noise and sudden changes. The hybrid filter algorithm is implemented in three different scenarios; each could potentially fail the PF, which is the most commonly used filter in nonlinear cases. Those scenarios include the kidnapped robot, changing/dynamic environments, high noise to signal ratio, and lower sampling rate. The results of the implementations using a laser scanner sensor show the hybrid filter provides a more accurate, continuous, and reliable position estimation. The flexibility of the hybrid filter algorithm to be applied by any PF and/or any FIR filter allows taking this research to another level and perhaps overcoming the NLOS situation.

Data Availability

The tracking data and all other info used or produced in this project will be available for the publisher.

Conflicts of Interest

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

References

[1] A. Basiri, E. S. Lohan, T. Moore et al., "Indoor location based services challenges, requirements and usability of current solutions," *Computer Science Review*, vol. 24, pp. 1–12, 2017.

[2] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proceedings of the Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068 of *Proceedings of SPIE*, pp. 182–193, Orlando, Fla, USA, 1997.

[3] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proceedings Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, 1999.

[4] G. Bishop and G. Welch, *An introduction to the Kalman filter*, Course 8, Proc of SIGGRAPH, 2001.

[5] M. Rekleitis Ioannis, *A particle filter tutorial for mobile robot localization*, Centre for Intelligent Machines, McGill University3480, 2004.

[6] M. A. Oskoei, "Adaptive Kalman filter applied to vision based head gesture tracking for playing video games," *Robotics*, vol. 6, no. 4, 2017.

[7] J. González, J. L. Blanco, C. Galindo et al., "Mobile robot localization based on ultra-wide-band ranging: a particle filter approach," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 496–507, 2009.

[8] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni, "Particle filter-based prognostics: Review, discussion and perspectives," *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 2–31, 2016.

[9] N. Oudjane and C. Musso, "Progressive correction for regularized particle filters," in *Proceedings of the Third International Conference on Information Fusion*, pp. THB2/10–THB2/17 vol.2, Paris, France, July 2000.

[10] W. R. Gilks and C. Berzuini, "Following a moving target—Monte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 1, pp. 127–146, 2001.

[11] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[12] T. Li, T. P. Sattar, and S. Sun, "Deterministic resampling: unbiased sampling to avoid sample impoverishment in particle filters," *Signal Processing*, vol. 92, no. 7, pp. 1637–1645, 2012.

[13] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, and M. T. Lim, "Improving reliability of particle filter-based localization in wireless sensor networks via hybrid particle/FIR filtering," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1089–1098, 2015.

[14] J. M. Pak, C. K. Ahn, M. T. Lim, and M. K. Song, "Horizon group shift FIR filter: Alternative nonlinear filter using finite recent measurements," *Measurement*, vol. 57, pp. 33–45, 2014.

[15] F. Gustafsson, F. Gunnarsson, N. Bergman et al., *Particle filters for positioning, navigation and tracking*, Linköping University Electronic Press, 2001.

[16] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–81, 2010.

[17] W. Yan and X. Li, "The IMU/UWB Fusion Positioning Algorithm Based on a Particle Filter," *ISPRS International Journal of Geo-Information*, vol. 8, no. 6, article 235, 2017.

[18] A. H. Jazwinski, "Limited Memory Optimal Filtering," *IEEE Transactions on Automatic Control*, vol. 13, no. 5, pp. 558–563, 1968.

[19] Y. Yamamoto, B. D. O. Anderson, M. Nagahara, and Y. Koyanagi, "Optimizing FIR approximation for discrete-time IIR filters," *IEEE Signal Processing Letters*, vol. 10, no. 9, pp. 273–276, 2003.

[20] S. Y. Cho and B. D. Kim, "Adaptive IIR/FIR fusion filter and its application to the INS/GPS integrated system," *Automatica*, vol. 44, no. 8, pp. 2040–2047, 2008.

[21] Y. S. Shmaliy, "Linear optimal FIR estimation of discrete time-invariant state-space models," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3086–3096, 2010.

[22] D. H. Pandya, S. H. Upadhyay, and S. P. Harsha, "Fault diagnosis of rolling element bearing with intrinsic mode function of acoustic emission data using APF-KNN," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4137–4145, 2013.

[23] N. Olivier-Maget, G. Hétreux, J. M. Le Lann, and M. V. Le Lann, "Model-based fault diagnosis for hybrid systems: Application on chemical processes," *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1617–1630, 2009.

[24] M. E. Orchard and G. J. Vachtsevanos, "A particle-filtering approach for on-line fault diagnosis and failure prognosis," *Transactions of the Institute of Measurement and Control*, vol. 31, no. 3-4, pp. 221–246, 2009.

[25] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[26] D. Törnqvist, T. B. Schön, R. Karlsson, and F. Gustafsson, "Particle filter SLAM with high dimensional vehicle model," *Journal of Intelligent & Robotic Systems*, vol. 55, no. 4-5, pp. 249–266, 2009.