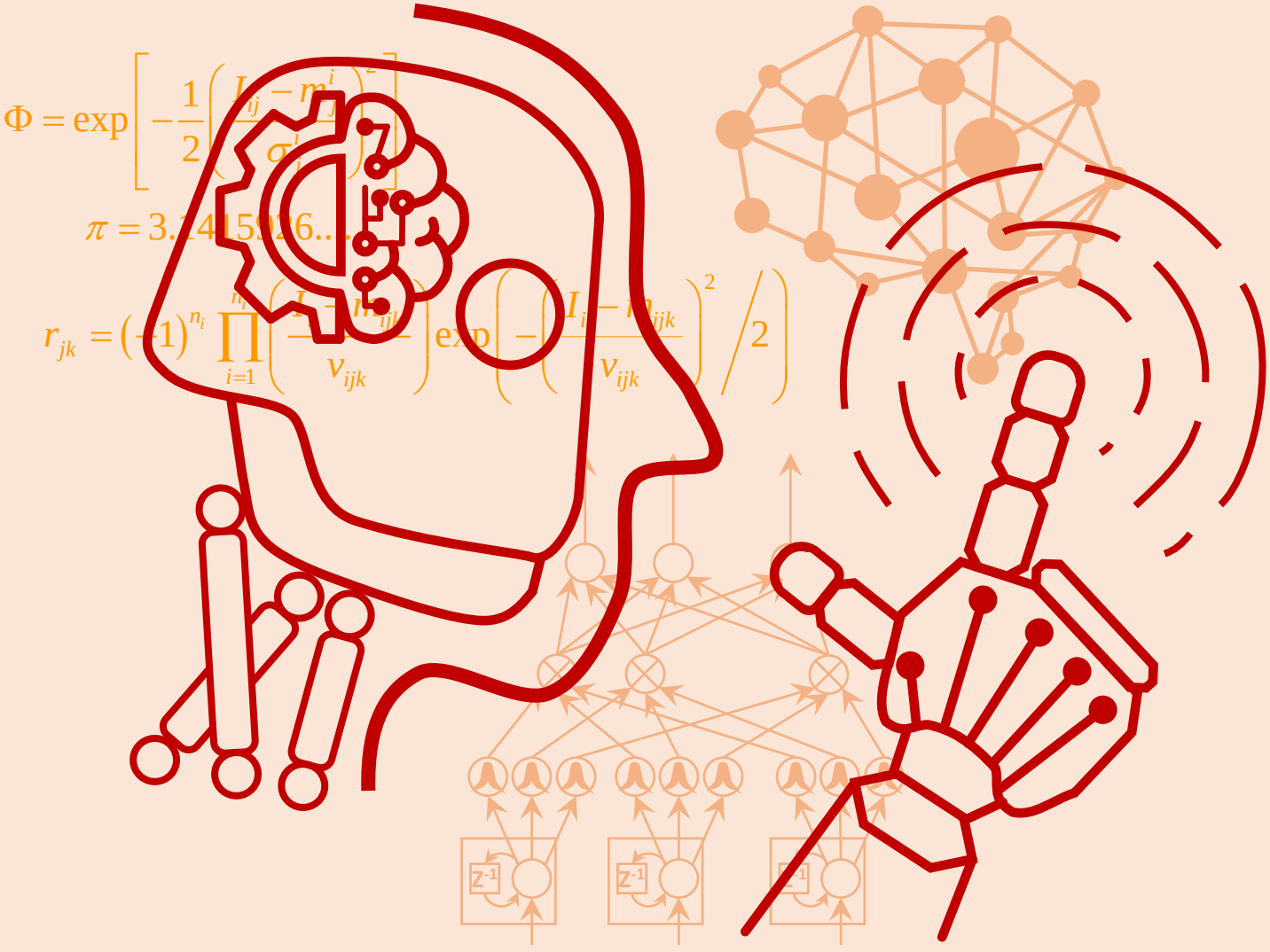
ISSN 2616-8170

Robotics Society of Taiwan

iRobotics

VOLUME 1
NUMBER 4
DECEMBER 2018



PUBLISHED BY THE ROBOTICS SOCIETY OF TAIWAN

*i*Robotics

PUBLISHER

Robotics Society of TAIWAN (RST)
Society President: Ching-Chih Tsai

Department of Electrical Engineering, National Chung Hsing University
Taichung, Taiwan

Tel: +886-4-2285-1549#601
URL: http://www.rst.org.tw

The ***i*Robotics** is published quarterly each year by the Robotics Society of Taiwan (RST). Institutional rate: US$140 annually; individual annual subscription rate: US$50 for nonmembers, US$25 for members (including postage). Note that another US$100 is needed if the express is required.

Development and Design of a Collaborative Air-Ground Robotic System in Indoor Warehouse

Xin-Cheng Lin and Ching-Chih Tsai

***Abstract*-This paper presents a collaborative air-ground robotic system with an indoor GPS-denied quadrotor working together with a Mecanum-wheeled mobile robot in indoor warehouse environments, in order to find and recognize cardboard boxes using deep neural networks on inventory shelfs. The quadrotor is used to fly in the warehouse environment and scan cardboard box's tags on the shelf. The Mecanum-wheeled mobile robot with one Nvidia TX2 AI computing module is employed to co-work with the quadrotor not only by stitching the images which are scanned by the quadrotor, but also by using tiny YOLOv2 network to detect and find the cardboard boxes on the shelf. This collaborative robotic system is operated under the ROS environment such that it is easily deployed in any real warehouse system. Experimental results are conducted to show the feasibility and effectiveness of the proposed method.**

***Index Terms*—Collaborative robotic system, inventory management, deep neural networks, YOLOv2 neural network, ROS.**

I. Introduction

In recent years, the use of robotic systems in inventory environments has been increasingly popular due to cost reduction and automation requirements. Therefore, there have been many related investigations and public competitions, such as the Amazon's robots used widely in their own warehouses, and Amazon's pick-up challenge. Furthermore, there are some studies presenting various methods investigating the use of robotic system in inventory environments. The authors in [1] proposed a mobile robot, which is able to deal with one inventory task, in the indoor environments; their proposed system had an ability to generate a costmap, which was an occupancy gird map, by using ultrasound, Lidar and RGB camera to sense and model the environment such that the used mobile robot was shown able to scan RFID tags for the inventory environment. However, this kind of robot was manually controlled without any navigation algorithm and automatic robot movement. The researchers in [2] presented an inventory checking system based on an outdoor storage yard environment, where the UAV of the inventory system which carried a portable PDA as a RFID reader and data collector was flew manually to scan RFID tags. Furthermore, air-ground robotic systems have been proposed by researchers to process and achieve inventory management; for example, the authors in [3] used a quadrotor flying above a ground robot, and utilized the marker-based method to scan bar-codes attached on the cardboard boxes. In [3], the ground robot navigated and localized itself by using a dead-reckoning technique and detecting the land-markers in the working environment, and the quadrotor flew over the environment by tracking the ground robot via its onboard bottom camera and scanned the bar-codes attached on the cardboard boxes. However, the flexibility of the quadrotor in [3] is restricted by the ground robot, namely that once the quadrotor fails to track the ground robot, it won't be able to localize and navigate itself even to navigate itself from one area to another, in order to process inventory management. To the best of our knowledge, there are few researchers focusing on how to process a complete inventory management via multi-robot systems in spite of the fact that there still exist many problems of the multi-robot systems working for warehouse environments. Thus, this motivates us to seek for a more flexible and reliable approach to allowing one kind of multi-robot system, especially for air-ground robotic system, to process a complete inventory task.

Deep learning neural networks have been widely applied to recognize various objects in huge amount of images. In particular, deep convolution neural networks (DCNNs) have been extensively used for image recognition and object understanding effectively and efficiently. Among DCNNs, YOLO NN in [8] has been shown as faster and more efficient neural networks, and has also been implemented into a mobile embedded system for real-time imaging applications. YOLOv2 NN is an improved version and function more accurately than its original version, YOLO NN, does [9]. In the paper, tiny YOLOv2 NN will be used for efficient object recognition in inventory good recognition.

This paper aims to propose a collaborative multi-robot system not only allowing the quadrotor to fly in indoor environments with the ground mobile robot, but also completing scanning task of desired tags attached on the cardboard boxes efficiently and precisely. The quadrotor, which is benefited from its efficient motion performance, is able to fly in front of the shelf and scan the cardboard box images. Meanwhile, the ground robot, which is benefited from its properties of high payload and omnidirectional mobility, is suitable to be equipped with a high-performance computer board, one Nvidia TX2 AI computing module, to execute high computational loads. Therefore, the quadrotor of the proposed system will be responsible to scan the images in front of the shelf and send back the scanned images to the computation center. Once these images received by the computation center on the ground robot, these images will be stitched into a complete image in order to find and detect the cardboard boxes with the attached tag by using YOLOv2 network and tag recognition algorithm.

Xin-Cheng Lin and Ching-Chih Tsai are with the Department of Electrical Engineering, National Chung Hsing University, Taichung, 40227, Taiwan.
(Corresponding author Ching-Chih Tsai, email: cctsai@nchu.edu.tw)
(email: d15321854@gmail.com)

The authors gratefully acknowledge financial support from the Ministry of Science and Technology (MOST), Taiwan, the Republic of China, under contract MOST 107-2221-E-005 -073 –MY2.

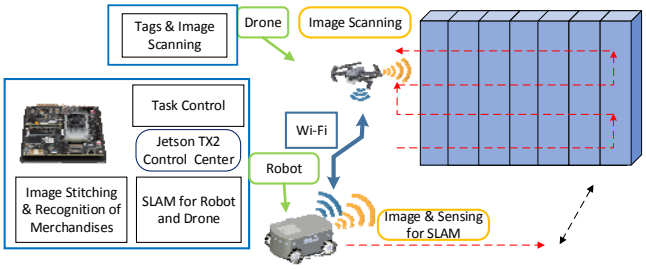


Fig. 1. Working scenario of the collaborative air-ground robotic system.

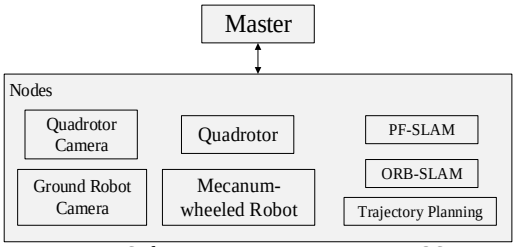


Fig. 2. Software system structure using ROS.

The rest of this paper is constructed as follows. The proposed system structure and description is stated in Section II. Section III addresses the indoor cardboard box tags scanning by using the quadrotor. Section IV delineates the cardboard box detection via YOLOv2 deep neural network. In Section V, experiments are conducted to verify the effectiveness and feasibility of the proposed method. Section VI draws the conclusions of the paper.

II. System Structure and Description

This section aims to introduce the structure of the proposed system, describe system configurations of the quadrotor, the Mecanum-wheeled robot and the ROS (Robot Operating System) software environment. The ROS will be used to be a robot framework and software platform, with each sensor or module being registered as a node and communicating with each other via the Master of the used ROS. The collaborative air-ground robotic system for the inventory management will be implemented in the ROS environment.

Fig. 1 shows the overview of the collaborative air-ground robot system in the indoor warehouse. The quadrotor is responsible for capturing images in front of the shelf. Meanwhile, the Mecanum-wheeled vehicle which carries a powerful computing module, NVIDIA Jetson TX2, is responsible for executing all the computations, including SLAM algorithm, quadrotor flying trajectory calculation and cardboard box tag recognition. Once the quadrotor scans the images in front of the shelf, the images will be stitched into a complete image in order to find and detect the cardboard boxes with the attached tags by using YOLOv2 neural network.

Fig. 2 depicts the overall system structure in the ROS environment. In Fig. 2, the quadrotor will be registered as a node, and it will publish its navigation data, the images of front and bottom cameras to the Master of ROS. The Mecanum-wheeled robot together with its on-board camera and Lidar will be registered as three nodes, respectively. Hence, they will publish the odometry data, camera images and Lidar

Fig. 3. Picture of the collaborative air-ground robotic system.

data to the Master of ROS. Once the odometry data, camera images and Lidar data have been published to the Master of ROS successfully, the PF-SLAM and ORB-SLAM algorithm in [4] as two nodes will be able to subscribe these data and process the computations of the algorithms. Meanwhile, the trajectory planning module will also be registered as a node in order to command the quadrotor and Mecanum-wheeled robot to the proper position by publishing control commands of both robots. Fig. 3 shows the collaborative air-ground robotic system, where the Mecanum-wheeled which has omnidirectional movement ability is able to assist the indoor quadrotor to process the cardboard box tags scanning task in indoor warehouse environments.

III. Indoor Cardboard Box Scanning Task

This section describes how to process the indoor cardboard boxes scanning task of the quadrotor by using image stitching technique and tiny YOLOv2 deep neural network. In order to perform a proper indoor quadrotor flight, two SLAM approaches, including ORB-SLAM and particle-filter SLAM, are deployed to localizing and automatically building the environmental map of the quadrotor and ground robot respectively [4]. Furthermore, to ensure the safety of the quadrotor while it flies close to the shelf, the land markers will be attached on the shelf. The SLAM technique guarantees the quadrotor and ground robot to navigate themselves safely in the warehouse environment, and the land markers technique ensures the safety and accuracy of the quadrotor to process the scanning task in the way of closing to the shelf. Once the quadrotor has flown in front of the shelf, the images of the cardboard boxes will be captured simultaneously. Moreover, these images will be stitched into complete images in order to form a complete shape of each cardboard box. Therefore, the tiny YOLOv2 neural network will be employed to find the cardboard boxes.

A. Image Scanning via the Quadrotor

Due to the image resolution, the size of each cardboard box and tag, the quadrotor must fly nearby each cardboard box in order to recognize the tag successfully. However, after the quadrotor flies close to a cardboard box, it's possible that the quadrotor cannot recognize it because it cannot see its complete shape. Moreover, it will not find other cardboard boxes because they cannot be seen.

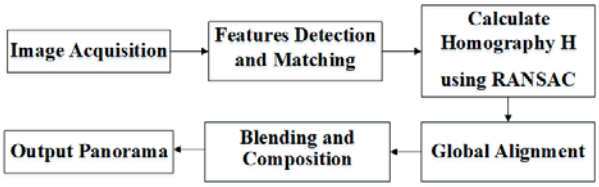


Fig. 4. Block diagram of the image stitching method.

For efficiency, we propose a novel but simple method to allow the quadrotor to fly only one time to collect all images of the shelf including the tags of the cardboard boxes. During the quadrotor flying in front of the shelf, the image of the shelf will be captured and these images will be stitched into a complete image. Once these images are stitched into a complete image, the cardboard boxes will be detected and its attached tag can be scanned as well. Furthermore, the complete image of the shelf is also able to provide the inventory information to the inventory manager in order to proceed with process inventory management. The scanning task is designed in the following four steps.

Step 1: Localize itself by assistance of the ground robot

The ground robot which processes SLAM algorithm navigates itself to the shelf, and simultaneously the quadrotor flies above the ground robot and localize itself assisted by the ground robot [4].

Step 2: Fly close to the shelf

Once the ground robot has found the shelf corner by detecting the QR-code attached on the shelf, the quadrotor will fly close to the shelf and start to scan the shelf image.

Step 3: Scan the cardboards' images via the flight sequence

The quadrotor will fly horizontally from one side of the shelf to another side. Afterwards, it will fly to the next flight height and repeat to fly from one side to another. During the flight sequence, the quadrotor is able to localize itself more accurately by using the QR-code markers attached on the shelf. At the same time, the locations of scanned images can be recorded as well.

Step 4: Complete the image scanning

Once the quadrotor completes the scanning task, it will fly back to stop atop the ground robot and wait for future scanning tasks.

B. Image Stitching for Cardboard Boxes Detection

Due to the limitations of the scanned images, it's hard to identify the cardboard box through the partial cardboard box image accurately. Therefore, it's necessary to stitch the continuous images of the cardboard box in order to identify and recognize the attached tag. Since the image stitching techniques have been developed for several years, one of the key methods are the two stitching methods in [5], where one is direct techniques and the other is feature-based method. In this paper, we deploy a feature-based method which matches images by establishing correspondences between points, lines, edges, corners or other geometric entries. Fig. 4 depicts the block diagram of the proposed stitching method, where the algorithm is divided into the following five steps.

Step 1: Image acquisition

The quadrotor which has an onboard camera is responsible for scanning the images of the shelf.

Step 2: Features detection and matching

To process the image stitching, it is necessary to detect the features of the images. Once the features of the images are detected, the images will be matched. A well-known algorithm, scale-invariant feature transform (SIFT) [6], is used to detect the features and match the images.

Step 3: Compute homography **H** using RANSAC

To compute the homography **H** matrix which projects each image onto the same surface for image stitching, this step deploys the Random sample consensus (RANSAC) algorithm [7] to estimate the homography **H** matrix. The first step is to choose number of sample *N*, and then choose 4 random potential matches. The second step is to compute **H** by using normalized Direct Linear Transform (DLT), and then to project points from $\mathbf{x}$ to $\mathbf{x}'$ for each potentially matching pair.

Step 4: Global alignment

To combine multiple images of the same scene into an accurate 3D reconstruction, we deploy bundle adjustment which applies an iterative algorithm to compute optimal values for the 3D reconstruction of the scene and camera positions.

Step 5: Blending and composition

This step is to select the final compositing surface, such as planar, cylindrical, spherical and cubic, and to blend the images together. We then select the cylindrical surface and blend the images of shelf together for cardboard box detection.

IV. Cardboard Box Detection by using YOLOv2 Neural Network

This section will investigate how to detect the cardboard boxes by using tiny YOLOv2 neural networks [8]-[9], and how to achieve fine-tuned pre-trained weights of the network in order to recognize the cardboard boxes in our own inventory environments more accurately. Due to the image deformation of image stitching, traditional object detection may not recognize the cardboard boxes successfully. To circumvent the difficulty, YOLO neural network, benefitting from the robust, power tool, and deep neural network, can be deployed to solve this problem. The YOLO neural network with the light-weight and real-time performance is particularly suitable to be implemented in an embedded system in this proposed system to detect the cardboard boxes with image deformation in the stitched image.

A. Cardboard Boxes Detection

To achieve the real-time detection performance on the embedded system board, NVIDIA Jetson TX2, we deploy the tiny YOLOv2 neural network to detect the cardboard boxes on the shelf. The tiny YOLOv2 neural network has the ability to recognize and localize the cardboard boxes in real-time with acceptable accuracy. The principle of the YOLO neural network is to divide the input image into a **S**×**S** gird, and then predict and detect objects through a single convolutional neural

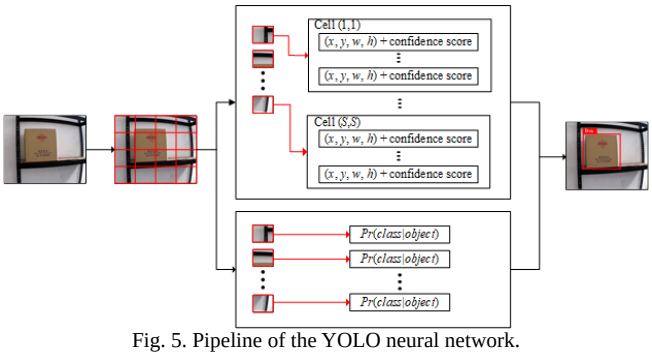


Fig. 5. Pipeline of the YOLO neural network.

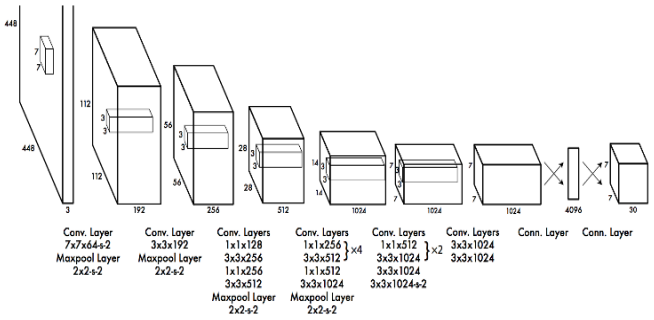


Fig. 6. Structure of the YOLOv2 deep neural network.

network. Each gird cell predicts a fixed number of bounding boxes, **B**, and each box has one box confidence score which reflects how likely the box contains an object and how accurate is the boundary box. Furthermore, each gird cell also predicts **C** conditional class probability (one per class for the likeliness of the object class) which is the probability that the detected object belongs to a particular class. Thus, the bounding boxes with high box confidence scores will be kept then to make a final prediction.

Once the box confidence box and conditional class probability are estimated, the class confidence score for each predicted box will be computed as:

$$P_\gamma\left(class_i\right)\cdot IoU = P_\gamma\left(object\right)\cdot IoU \times P_\gamma\left(class_i\left|object\right.\right)$$

$$class\ confidence\ score \equiv P_\gamma\left(class_i\right)\cdot IoU$$

$$box\ confidence\ score \equiv P_\gamma\left(object\right)$$

$$conditional\ class\ probability \equiv P_\gamma\left(class_i\left|object\right.\right)$$

where $P_\gamma(object)$ is the probability that the box contains an object, $P_\gamma(class_i|object)$ is the probability that the object belongs to $class_i$ given an object is presence, IoU is the IoU (intersection over union) between the predicted box and the ground truth, and $P_\gamma(class_i)$ is the probability that the object belongs to $class_i$. For more details, each bounding box contains 5 elements: offsets to the corresponding cell x and y; width w, height h of the image, and a box confidence score (x, y, w, h). In Fig. 5, it shows the pipeline of the YOLO neural network.

TABLE 1. STRUCTURE OF THE TINY YOLOV2 DEEP NEURAL NETWORK.

Layer	Filters	Size/Stride	Input	Output
0 conv	16	3 x 3 / 1	416 x 416 x 3	416 x 416 x 16
1 max		2 x 2 / 2	416 x 416 x 16	208 x 208 x 16
2 conv	32	3 x 3 / 1	208 x 208 x 16	208 x 208 x 32
3 max		2 x 2 / 2	208 x 208 x 32	104 x 104 x 32
4 conv	64	3 x 3 / 1	104 x 104 x 32	104 x 104 x 64
5 max		2 x 2 / 2	104 x 104 x 64	52 x 52 x 64
6 conv	128	3 x 3 / 1	52 x 52 x 64	52 x 52 x 128
7 max		2 x 2 / 2	52 x 52 x 128	26 x 26 x 128
8 conv	256	3 x 3 / 1	26 x 26 x 128	26 x 26 x 256
9 max		2 x 2 / 2	26 x 26 x 256	13 x 13 x 256
10 conv	512	3 x 3 / 1	13 x 13 x 256	13 x 13 x 512
11 max		2 x 2 / 1	13 x 13 x 512	13 x 13 x 512
12 conv	1024	3 x 3 / 1	13 x 13 x 512	13 x 13 x 1024
13 conv	512	3 x 3 / 1	13 x 13 x1024	13 x 13 x 512
14 conv	425	1 x 1 / 1	13 x 13 x 512	13 x 13 x 425
15 softmax				

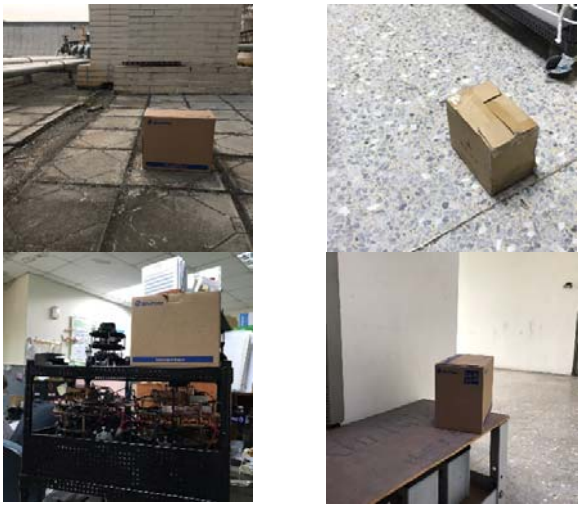


Fig. 7. Cardboard box image of our own datasets.

B. Design of the Neural Network's Architecture

The regular YOLOv2 neural network has 24 convolutional layers followed by 2 fully connected layers, as can be seen in Fig. 6. In this section, we only implemented a faster, light-weight version YOLOv2, tiny YOLOv2 network which only used 9 convolutional layers with shallower feature maps. Table 1 shows the structure of the tiny YOLOv2 neural network.

C. Fine-Tuning with the Pre-Trained Weights

To detect the customized object and increase the accuracy of the pre-trained tiny YOLOv2, we proceed with fine-tuning of the network by our own datasets. First of all, the "cardboard box" dataset was created by using our own images from different scenes including outdoor, indoor and warehouse scene, and then labeling the cardboard boxes by hands. To date, about 200 images have been constructed with labelled data in our dataset. Fig. 7 displays the four images from our "cardboard box" dataset. After creation of our own dataset, the network was fine-tuned with its weights by using NVIDIA Jetson TX2 GPU board.

V. Experimental Results and Discussion

This section presents the experimental results of the proposed collaborative air-ground robotic system in a demonstrative warehouse environment. The four experiments aim to show that the collaborative air-ground robotic system will complete the tags scanning task in the warehouse environment successfully. More importantly, the quadrotor will be shown able to fly in front of the shelf and scan the images of the shelf, and the ground robot will be shown able to carry high-performance computer board and detect cardboard boxes by using image stitching technique and tiny YOLOv2 neural network.

A. Image Stitching

In the first experiment, the image stitching module is verified if it has the ability to stitch the images with partial cardboard boxes and shelfs into a complete image by using the proposed stitching algorithm. Fig. 8 displays the images with partial shape of the cardboard box and shelf, and Fig. 9 shows the results of the stitching algorithm. Fig. 10 presents the images scanned by the onboard camera of the quadrotor. As can be seen in Fig. 11, the cardboard box was stitched successfully despite of the fact that there is a little image deformation. As shown in Fig. 12, the images were shown with QR-code attached on the cardboard box and shelf. The results in Fig. 13 reveal that the cardboard box image was stitched successfully as well.

Fig. 8. Images of the cardboard box.

Fig. 9. Stitched image of the cardboard box.

Fig. 10. Partial images of the cardboard box.

Fig. 11. Stitched image of the cardboard box.

Fig. 12. Partial images of the cardboard box.

Fig. 13. Stitched image of the cardboard box.

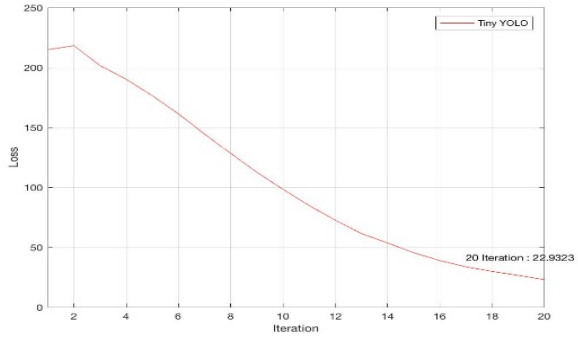


Fig. 14. Earlier stage of the training process.

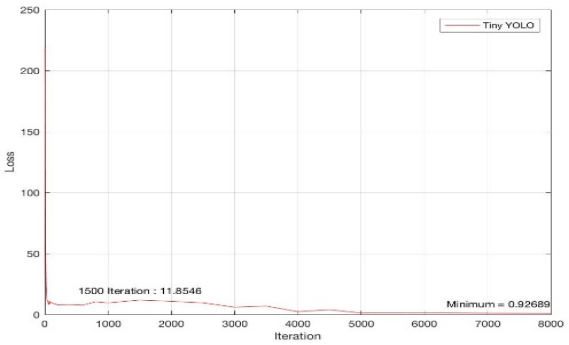


Fig. 15. Complete stage of the training process.

B. Fine-tuning of the Tiny YOLOv2 NN

The second experiment is carried out to examine whether the tiny YOLOv2 is successfully find-tuned with our own dataset. Fig. 14 shows the training curve of the network on the earlier stage, and Fig. 15 depicts that the loss value continuously decayed until the training process was stopped. The results in Figs. 14 and 15 indicate that the tiny YOLOv2 has been successfully find-tuned with our own dataset.

C. Cardboard Box Detection via Tiny YOLOv2 NN

The third experiment is conducted to investigate if the fine-tuned tiny YOLOv2 network has the ability to detect a cardboard box in the image. Fig. 16 depicts the testing results of the object detection using the tiny YOLOv2 NN, thereby showing that the network is capable of detecting and localizing the cardboard box in different view angles and scenes. Furthermore, Fig. 17 shows that this network is capable of recognizing the cardboard box in the stitched image with image deformation as well.

D. Cardboard Box Tags Scanning Task

Finally, the last experimental is aimed to explore if the quadrotor is able to complete the scanning task in the warehouse environment. In Fig. 18, six pictures showed that the quadrotor first flew close to the shelf and started to scan the shelf image, and then flew horizontally from one side of the shelf to the other; afterwards, the quadrotor flew to the next flight height and repeated to fly from one side to the other. Meanwhile, the Mecanum-wheeled robot was used to carry the computation center and assist the quadrotor to process the scanning task in the indoor warehouse environment. The experimental results in Fig. 18 that both robots cooperated to achieve the scanning task successfully.

Fig. 16. Testing of the tiny YOLOv2 neural network.

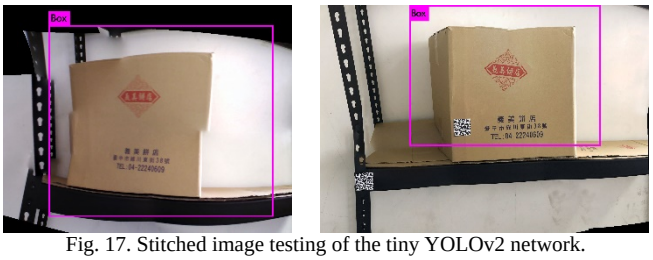


Fig. 17. Stitched image testing of the tiny YOLOv2 network.

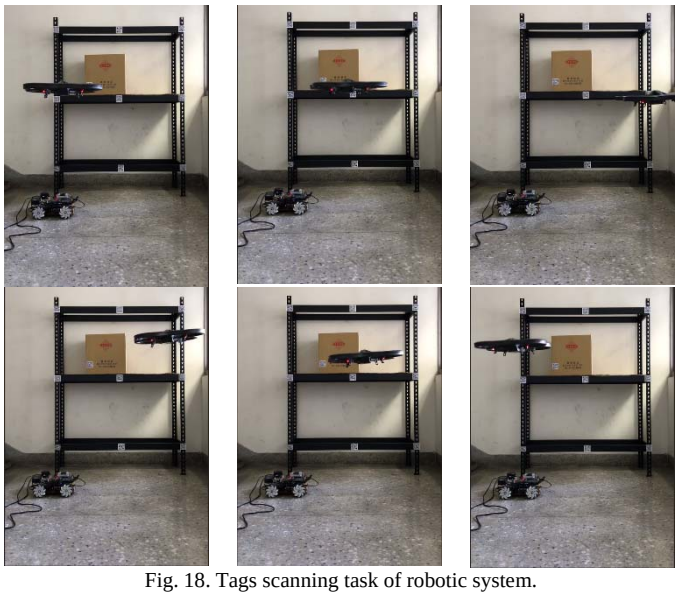


Fig. 18. Tags scanning task of robotic system.

VI. CONCLUSIONS

This paper has presented a collaborative air-ground robotic system in indoor warehouse allowing one quadrotor to scan cardboard box tag and one Mecanum-wheeled robot to process the image stitching algorithm, tiny YOLOv2 object detection, and SLAM algorithms. The collaborative air-ground robotic system has been implemented in the ROS environment. The quadrotor with an onboard camera has been responsible for scanning the images of the shelf and sending back these images to the ground robot. The Mecanum-wheeled robot carrying a high-performance computing module has been shown to work

well to process the image stitching algorithm and the tiny YOLOv2 neural network. Through experimental results, the system has been shown its effectiveness in an indoor warehouse. An interesting topic fir future work would be to integrate a QR-code detection and recognition method and an inventory management system into a more complete system.

REFERENCES

[1] J. Zhang, Y. Lyu, T. Roppel, J. Patton, and C P Senthilkumar, "Mobile robot for retail inventory using RFID," in *Proc. of 2016 IEEE Intern. Conf. on Industrial Technology (ICIT)*, Taipei, Taiwan, 2016.

[2] S. M.n Bae, K. H. Han, C. N. Cha, and H. Y. Lee, "Development of Inventory Checking system based on UAV and RFID in open storage yard," in *Proc. of 2016 International Conference on Information Science and Security (ICISS)*, Pattaya, Thailand, 2016.

[3] E. H. C. Harik, F. Guérin, F. Guinand, J. F. Brethé, and H. Pelvillain, "Towards an autonomous warehouse inventory scheme," in *Proc. of 2016 IEEE Symp. Series on Computa. Intelligence (SSCI)*, Athens, 2016, pp. 1-8.

[4] X. C. Lin and C. C. Tsai, "Cooperative SLAM of an indoor quadrotor flying together with a differential-driving automatic guided vehicle", in *Proc. of* 2018 *Intern. Conf. on Advanced Robotics and Intelligent Systems*, Taipei, Taiwan, 28-31 August, 2018.

[5] P. D. Kale and K. R. Singh, "A technical analysis of image stitching algorithm," *International Journal of Computer Scienceand Information Technologies (IJCSIT)*, vol. 6, no. 1, pp. 284-288, 2015.

[6] D.G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, Sept., 1999.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol.24 no.6, p.381-395, June 1981.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection." In *Proc. of the IEEE Intern. Conf. on computer vision and pattern recognition*, 2016, pp. 779-788.

[9] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *2017 IEEE Intern. Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA, 2017, pp. 6517-6525.

Xin-Cheng Lin he is currently pursuing the M.S. degrees in Department of Electrical Engineering from National Chung Hsing University, Taichung, Taiwan, ROC. His current research interests include mobile robots, intelligent control, SLAM and their applications to industrial processes and machines.

Ching-Chih Tsai received the Diplomat in Electrical Engineering from National Taipei Institute of Technology, Taipei, Taiwan, ROC, the MS degree in Control Engineering from National Chiao Tung University, Hsinchu, Taiwan, ROC and the Ph.D degree in Electrical Engineering from Northwestern University, Evanston, IL, USA, in 1981, 1986 and 1991, respectively. Currently, he is currently a Distinguished Professor in the Department of Electrical Engineering, National Chung-Hsing University, Taichung, Taiwan, where he served the Chairman in the Department of Electrical Engineering from 2012 to 2014. He is a Fellow of IEEE, IET and CACS.

Dr. Tsai served as the Chair, Taipei Chapter, IEEE Control Systems Society, from 2000 to 2003, and the Chair, Taipei Chapter, IEEE Robotics and Automation Society from 2005 to 2006. In 2007, he was the program chair of 2007 CACS international automatic conference sponsored by Taipei chapter, IEEE control systems society. In 2010, he served as the program co-chair of SICE 2010 annual conference in Taiwan, which was technically sponsored by IEEE CSS; in 2011, he served as the General Chair, 2011 International conference on service and interactive robotics; in 2012, he has served as the General Chair, 2012 International conference on Fuzzy Theory and Its Applications, the General Chair, 2012-2015 CACS International Automatic Control Conferences, and the General Chair, 2016-2017 International Conference on Advanced Robotics and Intelligent Systems. Dr. Tsai served the two-term President, Chinese Institute of Engineers in Central Taiwan, Taiwan from 2007 to 2011, and two-term President of Chinese Automatic Control Society from 2012 to 2015. Since 2008, he has been the Executive Directors in Boards of Government of thee professional associations, including Robotic Society of Taiwan, Taiwan Fuzzy Systems Association, and Taiwan Systems Association. He has served as the Chair, Taichung Chapter, IEEE Systems, Man, and Cybernetics Society since 2009, the Chair of IEEE SMC Technical Committee on intelligent learning in control systems since 2009, the President of Robotics Society of Taiwan since 2016, the steering committee of Asian Control Association since 2014, a BOG member of IEEE Nanotechnology council since 2012, the Vice President of International Fuzzy Systems Association since 2015, and a BOG member of the IEEE SMCS since 2017.

Dr. Tsai has published more than 500 technical papers, and seven patents in the fields of control theory, systems technology and applications. Web of Science has indexed his paper entitled "Adaptive Neural Network Control of a Self-Balancing Two-Wheeled Scooter" in the category Automation Control Systems, where the paper was ranked 408 out of 37607 articles (published between 2010 to 2014). Dr. Tsai is respectively the recipients of the Third Society Prize Paper Award from IEEE Industry Application Society in 1998, the Outstanding Automatic Control Engineering Award in 2008 from Chinese Automatic Control Society (CACS), and the Outstanding Engineering Professor Award in 2009 from the Chinese Institute of Engineers in 2009, the IEEE Most Active SMC Technical Committee (TC) Award in 2012 from IEEE SMC Society, the Outstanding Electrical Engineering Professor Award from the Chinese Institute of Electrical Engineering in 2014, Outstanding Industry Contribution Award from Taiwan Systems Association in 2016, the best paper award in the International Journal of Fuzzy Systems in 2017, and many best paper awards from many international conferences technically supported by IEEE. He is the advisor, IEEE SMC student branch chapter at National Chung Hsing University; this chapter was the recipient of certificate of appreciation from IEEE SMCS in 2009. He has served as the associate editors of International Journal of Fuzzy Systems, and IEEE Transactions on Systems, Man and Cybernetics: Systems, IEEE Transactions on Industry Informatics, and International Journal of Electrical Engineering. Recently, he has served as the Editor-in-Chief of a new international robotics journal called "iRobotics". His current interests include advanced nonlinear control methods, deep model predictive control, fuzzy control, neural-network control, advanced mobile robotics, intelligent service robotics, intelligent mechatronics, intelligent learning control methods with their applications to industrial processes and intelligent machinery.

Intelligent Motion Control Using Broad Learning System for Uncertain Inverse-Atlas Ball-Riding Robots

Bing-Yang Chen, Ching-Chih Tsai, and Feng-Chun Tai

***Abstract*-This paper presents a novel motion control method using broad-learning-system (BLS) for station keeping and trajectory tracking of an uncertain Inverse-Atlas ball-riding robot (IABRR) driven by three omnidirectional wheels. After brief description of the dynamic model of the robot with viscous and Coulomb frictions, a BLS-based compensator along with the backstepping sliding–mode controller is proposed to accomplish robust self-balancing and trajectory tracking of the robot in the presence of frictions variations, viscous and Coulomb frictions with unknown parameters and uncertainties. The proposed motion controller is proven asymptotically stable using Lyapunov stability theory. Computer simulations are conducted for illustration of the effectiveness of the proposed control method. The results show that the proposed controller is more effective and efficient than Chan's method [16].**

***Index Terms*—Backstepping, ball-riding robot, sliding-mode control, broad-learning-system (BLS), trajectory tracking.**

I. Introduction

SEVERAL ball robots have been developed to achieve mobile servicing in human living and work environments [1-7] in recent years. The ball-riding robots devised by the authors in [5-7] were constructed using three omnidirectional wheels driven by three DC servomotors. Moreover, ball-riding robots [5-7] have been designed as mobile platforms for indoor autonomous service robots because such robots have high center of gravity, narrow bases of support, agile mobility toward any poses, and dynamical stability. Hence, such mobile platforms have been shown useful in achieving efficient navigation around the human living space, and establish friend and convenient man-machine interactions.

Although the ball-riding robots have received much attention recently [5-7], the stability of the proposed state feedback control in [5-6] has not been analyzed because the complete dynamic model of the ball robot has not been established yet. On the other hand, the modeling and control of the ball robot developed in [6] have not been seen in the literature although the authors released the experimental videos to show how the robot worked successfully. The researchers in [8] presented a dynamic model and an LQR control method for a ball-riding robot, in order to achieve station keeping and point stabilization; however, the trajectory tracking stable control problems have not been solved yet [8].

Bing-Yang Chen, Ching-Chih Tsai and Feng-Chun Tai are with the Department of Electrical Engineering, National Chung Hsing University, Taichung, 40227, Taiwan, R.O.C. (Corresponding author Ching-Chih Tsai, email: cctsai@nchu.edu.tw) (email: g106064024@mail.nchu.edu.tw).

The authors gratefully acknowledge financial support from National Science Council, Taiwan, ROC, under contract NSC 106-2218-E-005-003-.

From viewpoints of control systems technology, the control of the ball-riding robot can be thought of as an under-actuated control problem, which has been investigated by several researchers [9-11]. In particular, Lo and Kuo [9] provided a decoupled sliding-mode control to stabilize a nonlinear system with four state variables, Lin and Mon [10] offered a hierarchical decoupling sliding-mode control to regulate a more general class of under-actuated control systems, and Wang *et al*. [11] presented two systematic sliding-mode design methods for a class of under-actuated mechanical systems. On the other hand, backstepping control method has been one important branch of nonlinear control approaches for model-based systems, aiming at a class of nonlinear systems whose dynamic models are governed by strict feedback forms. Backstepping sliding-mode control has been well studied in [12] by taking advantages of backstepping and sliding mode control. This control approach has been shown particularly useful in a class of second-order dynamic systems.

In recent years, the Broad Learning System (BLS) is proposed as an original flat structure established based on the random vector functional-link neural network (RVFLNN)[13]; the BLS inherits the major features of RVFLNN, and it can be expanded in a wide sense [14]. Chen *et al.* [15] further discussed the general approximation capability of the BLS and some variants with their mathematical models. The BLS is indeed different from some popular deep neural networks which have high compute cost and suffer from a time consuming learning for excessive parameters, thereby providing a much faster method with high accuracy. Due to these characteristics, it makes the BLS very efficient and much less time-consuming in function approximation and regression. However, the intelligent motion control augmented by BLS method has not been proposed for motion control of uncertain IABRRs.

Hence, the objective of the paper is to propose an intelligent motion control using BLS along with backstepping sliding–mode methodology for station keeping and trajectory tracking of the ball-riding robot with uncertain parameters. The presented contents are written in two main contributions. One is the rigorous derivation of the BLS-based intelligent motion controller using Lyapunov stability theory for the IABRR, the other is the comparative study on effectiveness, efficiency and superiority of the proposed method by comparing to an existing control method by Chan [16].

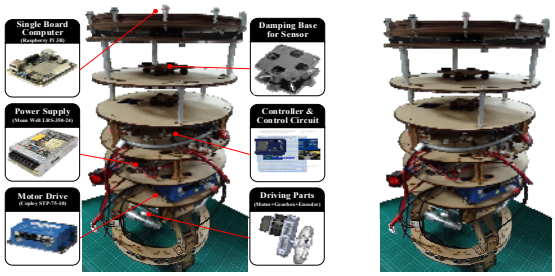
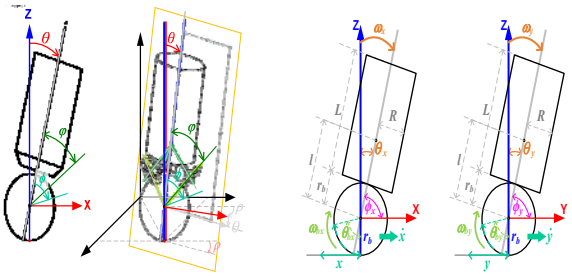


Fig. 1. The experimental ball-riding robot; (a) physical diagram; (b) laboratory-built prototype.



(a) (b) (c)

Fig. 2. Illustration of (a) the elevation angle θ, the zenith angle φ; (b) the tilt angle θ_x and the motors' angular positions ϕ_x in the median sagittal plane; (c) the tilt angle θ_y and the motors' angular positions ϕ_y in the median coronal plane.

The rest of the paper is organized as follows. Section II briefly describes the dynamic model of the IABRR with uncertainties, and Section III shows the structure of the used proposed BLS. In Section IV the proposed intelligent BLS controller is synthesized to achieve the design goals. Computer simulation is performed in Section V to illustrate the effectiveness and superiority of the proposed control method. Section VI concludes the paper.

II. System Modeling

The section aims to describe the system structure and physical configuration of the designed IABRR, and then recall its dynamic model. Table 1 lists the parameters of the ball-riding. Note that the modeling process is based on a simplification assumption that the ball-riding robot is constructed by two major components: the body and ball.

A. Kinematic and Dynamic Models of the IABRR

To steer the designed a stable controller for the IABRR, it is necessary to develop its dynamic model. As mentioned in [5-6,8], the IABRR is realized using two mechanisms: one is a two-dimensional inverted pendulum and the other is an omnidirectional mechanism using an inverse Atlas spherical motion platform for driving the ball. For the control purposes, two decoupled, independent dynamic equations for the two-dimensional mobile inverted pendulum are required in order to easily achieve trajectory tracking.

The basic concept in finding kinematic and dynamic model of the robot is to first obtain a relationship between the angular velocity vectors of the center of mass of the sphere $\bar{\Omega}$ and the angular speeds of the three omnidirectional wheels $\bar{\omega}_i$, which accounts for zero kinematic slip between

TABLE 1. LIST OF THE ROBOT PARAMETERS AND THEIR ACTUAL VALUES.

Symbol	Parameter description	Actual value
m_B	mass of the body	13.6kg
m_b	mass of the ball	2.7kg
l	distance between the center of the ball and the center of mass of the body	0.225m
r_B	radius of the body	0.15m
r_b	radius of the ball	0.105m
I_B	moment of inertia of the Body with respect to the center of the body	0.2678
I_b	moment of inertia of the ball with respect to the center of mass	0.0119 [kg·m^2]

the sphere and omnidirectional wheels. Since the related mathematical work about the Atlas spherical motion platform has been well described in [8], some related variables and functions are presented as follows. The relationship between the angular velocities, $\omega_i, i=1,2,3,$ of three omnidirectional wheels and the linear velocities, v_x, v_y in both x and y axes, and the rotational speed ω_z of the robot is expressed by

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}=\frac{1}{r}\begin{bmatrix}-v_y\cos\varphi+K_z\omega_z\\ (\frac{\sqrt{3}}{2}v_x+\frac{1}{2}v_y)\cos\varphi+K_z\omega_z\\ (-\frac{\sqrt{3}}{2}v_x+\frac{1}{2}v_y)\cos\varphi+K_z\omega_z\end{bmatrix} \tag{1}$$

Moreover, the angular velocity vector $\bar{\Omega}$ of the ball is given by

$$\begin{bmatrix}\Omega_1\\ \Omega_2\\ \Omega_3\end{bmatrix}=\frac{2r}{3r_b}\begin{bmatrix}\frac{-2\omega_1+\omega_2+\omega_3}{2\cos\varphi}\\ \frac{\sqrt{3}}{2\cos\varphi}(-\omega_2+\omega_3)\\ \frac{1}{2\sin\varphi}(\omega_1+\omega_2+\omega_3)\end{bmatrix}=\begin{bmatrix}v_y/r_b\\ -v_x/r_b\\ \omega_z\end{bmatrix} \tag{2}$$

Let the contact point $\bar{R}_s$ between the sphere and the ground have the position vector defined by $\bar{R}_s=-r_b\bar{k}$, where $\bar{i}$, $\bar{j}$ and $\bar{k}$ are respectively the unit normal vectors in the x, y and z axes. Hence, the velocity $\bar{\mathbf{V}}$ of the contact point between the driving ball and the ground is computed by

$$\begin{aligned}\bar{\mathbf{V}}=\bar{\Omega}\times\bar{R}_s&=\begin{bmatrix}\bar{i}&\bar{j}&\bar{k}\\ \Omega_1&\Omega_2&\Omega_3\\ 0&0&-r_b\end{bmatrix}\\ &=-r_b\Omega_2\bar{i}+r_b\Omega_1\bar{j}+0\bar{k}=V_x\bar{i}+V_y\bar{j}+V_z\bar{k}\end{aligned} \tag{3}$$

which leads to

$$V_x=\dot{x}=v_x,\quad V_y=\dot{y}=v_y,\quad V_z=0 \tag{4}$$

Taking the time derivative of Eq. (4), one easily obtains the dynamic equations of the inverse Atlas spherical motion platform in the following.

$$\ddot{x} = \dot{V}_x = \dot{v}_x = u_x, \quad \ddot{y} = \dot{V}_y = \dot{v}_y = u_y \tag{5}$$

where u_x and u_y are the acceleration control commands in both x and y axes.

B. Dynamic Modeling of the Two-Dimensional Mobile Inverted Pendulum

The subsection is aimed to describe a simplified planner dynamic model of the two-dimensional mobile inverted pendulum using Lagrangian mechanics. Similar to the modeling process investigated by Nagarajan *et al.* [3-4], our modeling approach makes the following three assumptions: (*i*) there is no slip between the spherical wheel and floor, (*ii*) the motion in the median sagittal plane and median coronal plane is decoupled, and (*iii*) the equations of motion in these two planes are identical. With these assumptions, one obtains two decoupled, independent planar dynamic models for the robot, thereby easily designing two independent motion controllers. In the following, as per the assumptions, the equations of motion in the median sagittal plane (x-z) are derived and then the vehicle dynamics in the median coronal plane (y-z) is assumed to have identical dynamics but with different symbols.

1) Reduced Dynamics in the Median Sagittal plane

To describe the dynamics of the robot in the sagittal plane, one needs to obtain two expressions of the kinetic and potential energy of the body and ball, form the Lagrangian function and finally obtain the Euler-Lagrange equations of motion of the two-dimensional ball-riding robot in the following vector matrix form

$$\mathrm{M_x}(\mathbf{q_x})\ddot{\mathbf{q}}_\partial + \mathrm{C_x}(\mathbf{q_x},\dot{\mathbf{q}}_x)\dot{\mathbf{q}}_x + \mathrm{G_x}(\mathbf{q_x}) + \mathrm{D_x}(\dot{\mathbf{q}}_x) = \begin{bmatrix} 0 & \tau_x \end{bmatrix}^T \tag{6}$$

where

$$\mathbf{q}_x = \begin{bmatrix} \theta_x & \phi_x \end{bmatrix}^T$$

$$\mathrm{M_x}(\mathbf{q_x}) = \begin{bmatrix} \alpha+\gamma+2\beta\cos(\theta_x) & \alpha+\beta\cos(\theta_x) \\ \alpha+\beta\cos(\theta_x) & \alpha \end{bmatrix}$$

$$\mathrm{C_x}(\mathbf{q_x},\dot{\mathbf{q}}_x)\dot{\mathbf{q}}_x = \begin{bmatrix} -\beta\dot{\theta}_x^{\,2}\sin(\theta_x) & -\beta\dot{\theta}_x^{\,2}\sin(\theta_x) \end{bmatrix}^T$$

$$\mathrm{G}_x(\mathbf{q}_x) = \begin{bmatrix} -\dfrac{\beta g}{r_b}\sin(\theta_x) & 0 \end{bmatrix}^T, \quad D_x(\dot{\mathbf{q}}_x) = \begin{pmatrix} D_1 \\ D_2 \end{pmatrix}$$

and $\alpha = I_b + (m_B + m_b)r_b^{\,2}$; $\beta = m_B r_b l$; $\gamma = m_B l^2 + I_B$. I_b, m_b and r_b are, respectively the moment of inertia, mass, and radius of the ball. μ_ϕ represents the viscous coefficients and $\mu_{c\phi}$ denotes the static friction coefficient.

Since $\dot{\phi}_x$ and $\ddot{\phi}_x$ can be calculated from the kinematic equations of motion of the ball, i.e., $\dot{\phi}_x = \dfrac{r}{\sqrt{3}r_b\cos\varphi}(\omega_2 - \omega_3) = \dfrac{v_x}{r_b}$ and $\ddot{\phi}_x = \dfrac{r}{r_b\sqrt{3}\cos\varphi}(\dot{\omega}_2 - \dot{\omega}_3)$ where $a_x = dv_x/dt$, then we have the following second- order equation of motion of θ_x from Eq. (6)

$$\begin{aligned} M_{x11}\ddot{\theta}_x - \beta\dot{\theta}_x^{\,2}\sin(\theta_x) - \frac{\beta g}{r_b}\sin\theta_x + D_{1x} &= -M_{x12}\ddot{\phi}_x \\ &= -\frac{M_{x12}}{r_b}u_x \end{aligned} \tag{7}$$

where $M_{x11} = \alpha + \gamma + 2\beta\cos(\theta_x)$, $M_{x12} = \alpha + \beta\cos(\theta_x)$ and $u_x = a_x$. Considering the uncertainty coming from the system parameters, one decomposes system parameters into nominal terms and perturbed terms as below;

$$M_{x11} = M_{x110} + \Delta M_{x11}, \quad \beta = \beta_0 + \Delta\beta, \quad M_{x12} = M_{x120} + \Delta M_{x12}$$

where perturbed terms are assumed to be bounded. Then, Eq. (7) turns out

$$\begin{aligned} M_{x110}\ddot{\theta}_x - \beta_0\dot{\theta}_x^{\,2}\sin(\theta_x) - \frac{\beta_0 g}{r_b}\sin\theta_x + U_x &= -M_{x120}\ddot{\phi}_x \\ &= \frac{-M_{x120}}{r_b}u_x \end{aligned} \tag{8}$$

where

$$\begin{aligned} U_x &= D_{1x} + \Delta M_{x11}\ddot{\theta}_x - \Delta\beta\dot{\theta}_x^{\,2}\sin(\theta_x) - \frac{\Delta\beta g\sin\theta_x}{r_b} + \Delta M_{x12}\ddot{\phi}_x \\ &= D_{1x} + \zeta \end{aligned}$$

represents the dynamic equation of the uncertain inverted pendulum in the Sagittal plane or in the x axis.

2) Reduced Dynamics in the Median Coronal plane

Similar to the previous section, the Euler-Lagrange equations of the robot in the median coronal plane can be derived and governed by the following matrix form

$$\mathrm{M_y}(\mathrm{q}_y)\ddot{\mathrm{q}}_y + \mathrm{C_y}(\mathrm{q}_y,\dot{\mathrm{q}}_y)\dot{\mathrm{q}}_y + \mathrm{G_y}(\mathrm{q}_y) + \mathrm{D_y}(\dot{\mathrm{q}}_y) = \begin{bmatrix} 0 & \tau_y \end{bmatrix}^T \tag{9}$$

In addition, because $\dot{\phi}_y$ and $\ddot{\phi}_y$ can be calculated from the kinematic equations of motion of the ball, i.e.,

$$\dot{\phi}_y = \frac{r(-2\omega_1 + \omega_2 + \omega_3)}{3r_b\cos\varphi} = \frac{v_y}{r_b} \quad \text{and} \quad \ddot{\phi}_y = \frac{r(-2\dot{\omega}_1 + \dot{\omega}_2 + \dot{\omega}_3)}{3r_b\cos\varphi} = \frac{a_y}{r_b}$$

where $a_y = dv_y/dt$, then we have the following second- order equation of motion of θ_x from Eq. (9)

$$\begin{aligned} M_{y11}\ddot{\theta}_y - \beta\dot{\theta}_y^{\,2}\sin(\theta_y) - \frac{\beta g}{r_b}\sin\theta_y + D_1 &= -M_{y12}\ddot{\phi}_y \\ &= -\frac{M_{y12}}{r_b}u_y \end{aligned} \tag{10}$$

where $M_{y11} = \alpha + \gamma + 2\beta\cos(\theta_y)$, $M_{y12} = \alpha + \beta\cos(\theta_y)$, and $u_y = a_y$. Similarly, we have

$$M_{y11} = M_{y110} + \Delta M_{y11}, \beta = \beta_0 + \Delta\beta, M_{y12} = M_{y120} + \Delta M_{y12}$$

Then, (10) is rewritten by

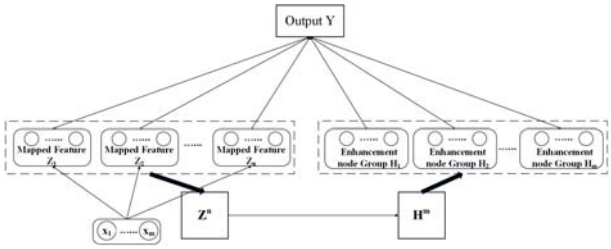


Fig. 3. Illustration of the BLS.

$$M_{y110}\ddot{\theta}_y - \beta_0\dot{\theta}_y{}^2\sin(\theta_y) - \frac{\beta_0 g}{r_b}\sin\theta_y + U_y = -\frac{M_{y120}}{r_b}u_y \quad (11)$$

where

$$\begin{aligned} U_y &= D_{1y} + \Delta M_{y11}\ddot{\theta}_y - \Delta\beta\dot{\theta}_y{}^2\sin(\theta_y) \\ &\quad - \frac{\Delta\beta g}{r_b}\sin\theta_y + \Delta M_{y12}\ddot{\phi}_y \\ &= D_{1y} + \zeta \end{aligned} \quad (12)$$

and Eq. (11) represents the dynamic equation of the uncertain ball-riding robot in the median coronal plane.

III. Review of BLS

A. Introduction to BLS

This subsection will briefly introduce the broad learning system structure as shown in Fig. 3. For the input data $\mathbf{X} \in R^{m-1}$, , the output data $\mathbf{Y} \in R^n$, and the nonlinear transformation ϕ_i, the output of the i[th] mapped feature is expressed by, $i = 1,...,n$,

$$\begin{aligned} \mathbf{Z_i} &= \phi_i(\bar{\mathbf{W}}_{fi}{}^T\mathbf{X} + \boldsymbol{\beta}_{fi}) \\ &= \phi_i(\left[\bar{\mathbf{W}}_{fi}{}^T \quad \boldsymbol{\beta}_{fi}\right]\left[\mathbf{X}^T \quad 1\right]^T) = \phi_i(\mathbf{W}_{fi}{}^T\mathbf{x}) \end{aligned} \quad (13)$$

where $\phi_i(\mathbf{x}) = \mathbf{x}$; $\bar{\mathbf{W}}_{fi}$ and $\boldsymbol{\beta}_{fi}$ are generalized weights and biases, respectively; $\mathbf{W}_{fi} = \left[\bar{\mathbf{W}}_{fi}{}^T \quad \boldsymbol{\beta}_{fi}\right]^T \in R^{m\times 1}$ and $\mathbf{x} = \left[\mathbf{X}^T \quad 1\right]^T \in R^m$. Let $\mathbf{Z} = [Z_1,...,Z_n]^T \in R^n$ and $\mathbf{W}_f = \left[\mathbf{W}_{f1}, \mathbf{W}_{f2}, ..., \mathbf{W}_{fn}\right] \in R^{m\times n}$.

Hence, it follows that

$$\mathbf{Z} = \phi(\mathbf{W_f}{}^T\mathbf{x}) \quad (14)$$

Next, we group the first n feature maps together and establish the j^{th} enhancement node group whose output is obtained from, $j = 1,...,m$,

$$\begin{aligned} \mathbf{H}_j &= \xi_j(\bar{\mathbf{W}}_{ej}{}^T\mathbf{Z} + \boldsymbol{\beta}_{ej}) \\ &= \xi_j(\left[\bar{\mathbf{W}}_{ej}{}^T \quad \boldsymbol{\beta}_{ej}\right]\left[\mathbf{Z}^T \quad 1\right]^T) = \xi_j(\mathbf{W}_{ei}{}^T\mathbf{z}) \end{aligned} \quad (15)$$

where $\xi_j = \tanh(\bullet)$; $\bar{\mathbf{W}}_{ej}$ and $\boldsymbol{\beta}_{ej}$ are also respectively generalized weights and biases. $\mathbf{W}_{ei} = \left[\bar{\mathbf{W}}_{ei}{}^T \quad \boldsymbol{\beta}_{ei}\right]^T$ and $\mathbf{z} = \left[\mathbf{Z}^T \quad 1\right]^T \in R^{n+1}$. Moreover, let us represent the first m groups of the enhancement nodes by $\mathbf{H} = [H_1,...,H_m]^T \in R^m$. let $\mathbf{W}_e = \left[\mathbf{W}_{e1} \quad \mathbf{W}_{e2} \quad \cdots \quad \mathbf{W}_{em}\right]$, and one obtains

$$\mathbf{H} = \xi(\mathbf{W}_e{}^T\mathbf{z}) \quad (16)$$

and the output of the broad learning system as

$$\mathbf{Y} = \mathbf{W}^T\boldsymbol{\Psi} \quad (17)$$

where $\mathbf{W}$ is the weighting matrix connects mapped features and enhancement node groups to output, and $\boldsymbol{\Psi} = [\mathbf{Z}^T \mid \mathbf{H}^T]^T$.

B. Vector Uncertain Function Approximation by BLS

The BLS has been shown capable of uniformly approximating any real continuous vector function $f(x)\in R^1$ on a compact set $\mathbf{U}$ to any arbitrary accuracy b_ε, i.e., there exists an ideal BLS with ideal parameters $\mathbf{W}$ such that

$$\sup_{x\in\mathbf{U}}\left\|f(x) - \mathbf{W}^T\boldsymbol{\Psi}\right\| < b_\varepsilon \quad (18)$$

Therefore, vector function $f(x)$ can be represented as

$$f(x) = \mathbf{W}^T\boldsymbol{\Psi} + \varepsilon \quad (19)$$

where $\|\varepsilon\| \le b_\varepsilon$. In this paper, all norms of vectors and matrices adopt Frobenius norm. The norms of the ideal parameters should satisfy the following assumption.

Assumption 1: The norms of ideal parameters, $\|\mathbf{W}_e\|$, $\|\mathbf{W}_f\|$ and $\|\mathbf{W}\|$, are bounded by positive real values, i.e. $\|\mathbf{W}_e\| \le b_e$, $\|\mathbf{W}_f\| \le b_f$ and $\|\mathbf{W}\| \le b_w$. Clearly, we need to estimate the ideal BLS by an estimate BLS

$$\hat{f}(x) = \hat{\mathbf{W}}^T\hat{\boldsymbol{\Psi}} \quad (20)$$

The weight updating law will be stated in the following. By defining $\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}$, $\tilde{\mathbf{W}}_f = \mathbf{W}_f - \hat{\mathbf{W}}_f$ and $\tilde{\mathbf{W}}_e = \mathbf{W}_e - \hat{\mathbf{W}}_e$, we have

$$f(\mathbf{x}) = \hat{f}(\mathbf{x}) + \tilde{f}(\mathbf{x}) \quad (21)$$

where

$$\begin{aligned} \tilde{f}(\mathbf{x}) &= \tilde{\mathbf{W}}^T\frac{\partial\tilde{f}}{\partial\mathbf{W}} + \frac{\partial\tilde{f}}{\partial\mathbf{W_f}}\tilde{\mathbf{W}}_\mathbf{f} + \frac{\partial\tilde{f}}{\partial\mathbf{W}_e}\tilde{\mathbf{W}}_\mathbf{e} + \varepsilon_t \\ &= \tilde{\mathbf{W}}^T\boldsymbol{\Psi} + \hat{\mathbf{W}}^T\frac{\partial\boldsymbol{\Psi}}{\partial\mathbf{W_f}}\tilde{\mathbf{W}}_\mathbf{f} + \hat{\mathbf{W}}^T\frac{\partial\boldsymbol{\Psi}}{\partial\mathbf{W_e}}\tilde{\mathbf{W}}_\mathbf{e} + \varepsilon_t \end{aligned} \quad (22)$$

where $\varepsilon_t = \tilde{\mathbf{W}}^T\tilde{\boldsymbol{\Psi}} + \varepsilon$ and ε_t is assumed to satisfy $\|\varepsilon_t\| < \varepsilon_{\max}$.

IV. Intelligent Motion Control Using BLS

This section will present the procedures of designing the proposed motion controller for tracking $x_d(t)$ and $y_d(t)$, and stabilizing both variables θ_x and θ_y at $\theta_{xd}(t)$ and $\theta_{yd}(t)$, respectively. Since the dynamic equations of motion of the

robot in both sagittal and coronal planes are identical except the notations. Hence, only the controller in the sagittal plane is designed during the controller design process. The same controller can be straightforward applied to control the robot in the other plane.

The design procedure is divided into two steps; the first step uses backstepping to design a virtual controller, and the second step defines the sliding surface and designs the sliding-mode controller.

By defining the following four state variables, $x_{x1} = x$, $x_{x2} = \theta_x$, $x_{x3} = \dot{x}$, $x_{x4} = \dot{\theta}_x$, the dynamic model (8) in the median sagittal plane can be re-expressed by

$$\begin{aligned}\dot{\mathbf{x}}_x &= \begin{bmatrix}\dot{x}_{x1} & \dot{x}_{x2} & \dot{x}_{x3} & \dot{x}_{x4}\end{bmatrix}^T \\ &= \begin{bmatrix}\dot{x} & \dot{\theta}_x & u_x & \left(A_x u_x + B_x + U_b\right)\end{bmatrix}^T\end{aligned} \tag{23}$$

where

$A_x = \dfrac{-M_{x120}}{M_{x110} r_b}$, $B_x = \left(\dfrac{\beta g}{M_{x110} r_b} + \dfrac{\beta \dot{\theta}_x^{\,2}}{M_{x110}}\right)\sin(\theta_x)$, $U_b = \dfrac{U_x}{M_{x110}}$,

and $u_x = \tau_x$.

A. Virtual Control Design

In order to achieve the control goal, one defines the following tracking error

$$x_{x1e} = x - x_d \quad \Rightarrow \quad \dot{x}_{x1e} = \dot{x}_1 - \dot{x}_d = x_3 - \dot{x}_d \tag{24}$$

$$x_{x2e} = \theta_x - \theta_{xd} \quad \Rightarrow \quad \dot{x}_{x2e} = \dot{x}_2 - \dot{\theta}_{xd} = x_4 - \dot{\theta}_{xd} \tag{25}$$

Consider the two state variables, x_{x3} and x_{x4}, as two virtual (ideal) controls and set

$$\phi(x_1, x_2) = \begin{pmatrix} x_{x3\_virtual} \\ x_{x4\_virtual} \end{pmatrix} = -K_p \begin{bmatrix} x_{x1e} \\ x_{x2e} \end{bmatrix} + \begin{bmatrix} \dot{x}_d \\ \dot{\theta}_{xd} \end{bmatrix} \tag{26}$$

where $K_p = \begin{bmatrix} k_{p1} & 0 \\ 0 & k_{p2} \end{bmatrix} > 0$

Hence, we have

$$\dot{x}_{1e} = -k_{p1} x_{1e} \Rightarrow \ddot{x}_{1e} + k_{p1}\dot{x}_{1e} = 0 \tag{27}$$

$$\dot{x}_{2e} = -k_{p2} x_{2e} \Rightarrow \ddot{x}_{2e} + k_{p2}\dot{x}_{2e} = 0 \tag{28}$$

By introducing the Lyapunov function

$$\mathrm{V}_1 = \frac{1}{2}\begin{bmatrix} x_{x1e} & x_{x2e} \end{bmatrix}\begin{bmatrix} x_{x1e} & x_{x2e} \end{bmatrix}^T \tag{29}$$

one obtains time derivative of the Lyapunov function

$$\dot{\mathrm{V}}_1 = -\begin{bmatrix} x_{x1e} & x_{x2e} \end{bmatrix} K_p \begin{bmatrix} x_{x1e} & x_{x2e} \end{bmatrix}^T \le 0 \tag{30}$$

According to Barbalat's lemma, $\dot{\mathrm{V}}_1$ is negative semi-define function and it implies that

$$\begin{pmatrix} x_{x1e} & x_{x2e} \end{pmatrix}^T \to 0 \text{ as } t \to \infty \Rightarrow x_{x1e}, x_{x2e} \to 0 \text{ as } t \to \infty \tag{31}$$

Next, let the backstepping error vector be defined as follows;

$$\eta_x = \underbrace{\begin{pmatrix} x_{x3} \\ x_{x4} \end{pmatrix}}_{\text{True}} - \underbrace{\phi(x_{x1}, x_{x2})}_{\text{Virtual control}} = \begin{pmatrix} x_{x3} \\ x_{x4} \end{pmatrix} + K_p \begin{bmatrix} x_{x1e} \\ x_{x2e} \end{bmatrix} - \begin{bmatrix} \dot{x}_d \\ \dot{\theta}_{xd} \end{bmatrix} \tag{32}$$

With the definition of the backstepping error vector, the dynamics of $\begin{bmatrix} \dot{x}_1 & \dot{x}_2 \end{bmatrix}^T$ can be rewritten by

$$\begin{bmatrix} \dot{x}_{1e} & \dot{x}_{2e} \end{bmatrix}^T = \eta_x - K_p \begin{bmatrix} x_{1e} & x_{2e} \end{bmatrix}^T \tag{33}$$

In order to start with the design of the control law for u_x, it is necessary to obtain the time derivative of the backstepping error vector $\eta_x = \begin{bmatrix} \eta_{1x} & \eta_{2x} \end{bmatrix}^T$

$$\begin{aligned}\dot{\eta}_x &= \begin{pmatrix} \dot{x}_{x3} \\ \dot{x}_{x4} \end{pmatrix} + K_p \begin{bmatrix} \dot{x}_{x1e} \\ \dot{x}_{x2e} \end{bmatrix} - \begin{bmatrix} \ddot{x}_d \\ \ddot{\theta}_{xd} \end{bmatrix} \\ &= \begin{pmatrix} u_x \\ A_x u_x + B_x \end{pmatrix} + \begin{pmatrix} k_{p1}\dot{x}_{x1e} - \ddot{x}_d \\ U_b + k_{p2}\dot{x}_{x2e} - \ddot{\theta}_{xd} \end{pmatrix}\end{aligned} \tag{34}$$

B. Backstepping Hierarchical Aggregated Sliding Surface and Control

In order to make the backstepping error vector η to converge to zero, it is necessary to construct the subsequent sliding surface by obeying the design procedure of the hierarchical aggregated sliding-mode controller in [11],

$$S_x = S_{1x} + rS_{2x}, \quad S_{1x} = \eta_{1x}, \quad S_{2x} = \eta_{2x} \tag{35}$$

where r is a real constant. The time derivative of the second-layer sliding surface S is given by

$$\begin{aligned}\dot{S}_x &= \dot{S}_{1x} + r\dot{S}_{2x} = (\dot{\eta}_{1x}) + r(\dot{\eta}_{2x}) \\ &= \left(u_x + k_{p1}\dot{x}_{x1e} - \ddot{x}_d\right) + r\left(A_x u_x + B_x + U_b + k_{p2}\dot{x}_{x2e} - \ddot{\theta}_{xd}\right) \\ &= (1 + rA_x)(u_x + f_x) - \ddot{x}_d - r\ddot{\theta}_{xd}\end{aligned} \tag{36}$$

where $f_x = -(1 + rA_x)^{-1}\left(k_{p1}\dot{x}_{x1e} + r\left(B_x + U_b + k_{p2}\dot{x}_{x2e}\right)\right)$

Let the control be decomposed into $u_x = u_{x\_eq} + u_{x\_sw}$, the equivalent and switching control, respectively. The equivalent control u_{x_eq} can be found such that $\dot{S}_x = 0$,

$$u_{x\_eq} = -f_x \tag{37}$$

Since the function f_x in (37) is unknown, one proposes that the control is learned by the BLS as follows;

$$u_{x\_eq} = -\left(\hat{f}_x + \varepsilon_{tx}\right) \tag{38}$$

where the function $\hat{f}_x$ in (38) is online learned by the BLS. According to the result in Subsection 3.2 the ideal function u_{x_eq} is expressed as

$$u_{x\_eq} = -\begin{pmatrix} \hat{f}(\mathbf{x}) + \tilde{\mathbf{W}}\boldsymbol{\Psi} + \hat{\mathbf{W}}^T \dfrac{\partial \boldsymbol{\Psi}}{\partial \mathbf{W_f}} \tilde{\mathbf{W}}_\mathbf{f} \\ + \hat{\mathbf{W}}^T \dfrac{\partial \boldsymbol{\Psi}}{\partial \mathbf{W_e}} \tilde{\mathbf{W}}_\mathbf{e} + \varepsilon_t \end{pmatrix} \tag{39}$$

and the actual output of the BLS is then given by

$$\hat{u}_{x\_eq} = -\hat{f}_x = -\hat{\mathbf{W}}^T \hat{\boldsymbol{\Psi}}$$

The inclusion of the switching-like control which is

$$u_{x\_sw} = \left(k_1 + c_1 \left| r_1 \ddot{\theta}_{xd} + r_2 \ddot{\phi}_{xd} \right| + g\right) \mathrm{sgn}(S_x) + k_2 S_x \tag{40}$$

gives the resultant torque u_x,

$$\begin{aligned} u_x &= \hat{u}_{x\_eq} + u_{x\_sw} \\ &= -\hat{f}_x + \left(k_1 + c_1 \left| r_1 \ddot{\theta}_{xd} + r_2 \ddot{\phi}_{xd} \right| + g\right) \mathrm{sgn}(S_x) + k_2 S_x \end{aligned} \tag{41}$$

where k_1 and c_1 are two positive and constant gains.

In order to prove the stability of the controller Eq. (41), the Lyapunov function is selected as below;

$$\begin{aligned} V_2 = \frac{1}{2} {S_x}^2 + \frac{1}{2r_w} \tilde{\mathbf{W}}^T \tilde{\mathbf{W}} + \frac{1}{2r_{wf}} \tilde{\mathbf{W}}_f^T \tilde{\mathbf{W}}_f \\ + \frac{1}{2r_{we}} \tilde{\mathbf{W}}_e^T \tilde{\mathbf{W}}_e + \frac{1}{2r_{\varepsilon x}} \tilde{\varepsilon}_{\max}{}^2 \end{aligned}$$

where r_w, r_{wf}, r_{we} and $r_{\varepsilon x}$ are real and positive. Let $g = \hat{\varepsilon}_{\max}$, and the time derivative of $S_x^2/2$ is expressed by

$$\begin{aligned} \frac{d}{dt}\left(\frac{S_x^2}{2}\right) &= S_x \left((1 + rA_x)\left(u_x + f_x\right) - \ddot{\theta}_{xd} - r\ddot{\phi}_{xd}\right) \\ &\le -S_x \left|(r_1 + r_2 A_x)\right| \begin{pmatrix} \tilde{\mathbf{W}}\boldsymbol{\Psi} + \hat{W}^T \dfrac{\partial \boldsymbol{\Psi}}{\partial \mathbf{W_f}} \tilde{\mathbf{W}}_\mathbf{f} \\ + \hat{\mathbf{W}}^\mathbf{T} \dfrac{\partial \boldsymbol{\Psi}}{\partial \mathbf{W_e}} \tilde{\mathbf{W}}_\mathbf{e} - \varepsilon_{\max} \end{pmatrix} \\ &\quad - S_x \left|(r_1 + r_2 A_x)\right| \left(\begin{pmatrix} k_1 + \hat{\varepsilon}_{\max} \\ + c_1 \left| r_1 \ddot{\theta}_{xd} + r_2 \ddot{\phi}_{xd} \right| \end{pmatrix} \mathrm{sgn}(S_x) + k_2 S_x \right) \\ &\quad + \left|S_x\right| \left| \left(r_1 \ddot{\theta}_{xd} + r_2 \ddot{\phi}_{xd}\right) \right| \\ &\le -\left|(r_1 + r_2 A_x)\right| \begin{pmatrix} S_x \begin{pmatrix} \tilde{\mathbf{W}}^T \boldsymbol{\Psi} + \hat{\mathbf{W}}^T \dfrac{\partial \boldsymbol{\Psi}}{\partial \mathbf{W_f}} \tilde{\mathbf{W}}_\mathbf{f} \\ + \hat{\mathbf{W}}^T \dfrac{\partial \boldsymbol{\Psi}}{\partial \mathbf{W_e}} \tilde{\mathbf{W}}_\mathbf{e} \end{pmatrix} \\ + k_2 S_x^2 + \left(\tilde{\varepsilon}_{\max} + k_1\right)\left|S_x\right| \end{pmatrix} \end{aligned} \tag{42}$$

where $(1+rA_x)<0$.

Thus, differentiating the Lyapunov function V_2 yields

$$\begin{aligned} \dot{V}_2 &= S_x \dot{S}_x + \frac{1}{r_w} \tilde{\mathbf{W}}^T \dot{\tilde{\mathbf{W}}} + \frac{1}{r_{wf}} \tilde{\mathbf{W}}_f^T \dot{\tilde{\mathbf{W}}}_f + \frac{1}{r_{we}} \tilde{\mathbf{W}}_e^T \dot{\tilde{\mathbf{W}}}_e \\ &\quad + \frac{1}{r_{\varepsilon x}} \tilde{\varepsilon}_{\max} \dot{\tilde{\varepsilon}}_{\max} \\ &< -\left|(r_1 + r_2 A_x)\right| \left(k_1 \left|S_x\right| + k_2 S_x^2\right) \\ &\quad + \tilde{\varepsilon}_{\max} \left((r_1 + r_2 A_x)\left|S_x\right| - \dot{\tilde{\varepsilon}}_{\max} / r_{\varepsilon x} \right) \\ &\quad + \tilde{\mathbf{W}}^T \left((r_1 + r_2 A_x) S_x \boldsymbol{\Psi} - \dot{\tilde{\mathbf{W}}} / r_{wx} \right) \\ &\quad + \tilde{\mathbf{W}}_f^T \left((r_1 + r_2 A_x)\left(\frac{\partial \hat{f}(\mathbf{x})}{\partial W_f}\right)^T - \dot{\tilde{\mathbf{W}}}_f / r_{wf} \right) \\ &\quad + \tilde{\mathbf{W}}_e^T \left((r_1 + r_2 A_x)\left(\frac{\partial \hat{f}(\mathbf{x})}{\partial W_e}\right)^T - \dot{\tilde{\mathbf{W}}}_e / r_{\omega e} \right) \end{aligned} \tag{43}$$

Therefore, we take the following parameter adaptation rules

$$\begin{aligned} \dot{\hat{\varepsilon}}_{\max} &= r_{\varepsilon x} (r_1 + r_2 A_x) \left|S_x\right| \\ \dot{\hat{\mathbf{W}}} &= r_{wx} (r_1 + r_2 A_x) S_x \boldsymbol{\Psi} \\ \dot{\hat{\mathbf{W}}}_f &= r_{cx} (r_1 + r_2 A_x) S_x \left(\frac{\partial \hat{f}(\mathbf{x})}{\partial \mathbf{W_f}}\right)^T \\ \dot{\hat{\mathbf{W}}}_e &= r_{\omega x} (r_1 + r_2 A_x) S_x \left(\frac{\partial \hat{f}(\mathbf{x})}{\partial \mathbf{W_e}}\right)^T \end{aligned} \tag{44}$$

which leads to

$$\begin{aligned} \dot{V}_2 &\le -\left|(r_1 + r_2 A_x)\right| \left(k_1 \left|S_x\right| + k_2 S_x^2\right) \\ &\le -k_x \left(k_1 \left|S_x\right| + k_2 S_x^2\right) \le 0, \text{ if } k_x \le \left|(r_1 + r_2 A_x)\right| \end{aligned} \tag{45}$$

Since $\dot{V}_2$ are negative semi-definite, it is easy to show via Lyapunov stability theory that the second-layer sliding function S_x converges to the origin asymptotically.

In what follows, we let r be a switching gain where $r=r_0\mathrm{sgn}(S_{1x}S_{2x})$, $r_0>0$, for the IASBRR since the origin is inherently unstable, and show that the control effort u will force the system state to reach the first-layer sliding-mode surface in finite time. In doing so, we first prove $|S_{1x}| \in L_2$, $|S_{2x}| \in L_2$, i.e.

$$\int_0^\infty \left|S_{1x}\right|^2 < \infty, \quad \int_0^\infty \left|S_{2x}\right|^2 d\tau < \infty \tag{46}$$

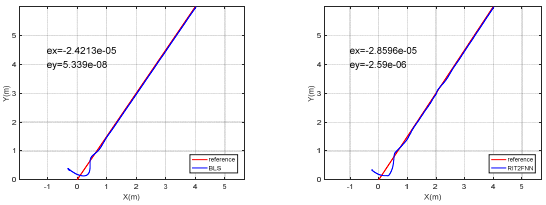
From (45), we have,

$$\begin{aligned} V_2(t) &= V_2(0) - \int_0^\infty \left(k_x k_1 \left|S_x\right| + k_x k_2 {S_x}^2\right) d\tau \\ &\le V_2(0) < \infty \Rightarrow \int_0^\infty {S_x}^2 d\tau < \infty \end{aligned} \tag{47}$$

which leads to know that

TABLE II. TOTAL SIMULATION TIME OF THE PROPOSED METHOD USING BLS AND THE CHAN'S METHOD [16].

	BLS	RIT2FNN
Averaged execution time per sampling (ms)	3.28	17.31
ISE	117.4100	180.5710
IAE	906.6102	1.3723e+03
ITAE	4.2390e+03	6.9999e+03
ITSE	122.8238	225.8240



(a) (b)

Fig. 4. Simulation results of the straight line tracking: (a) the proposed method using BLS; (b) the Chan's method using RIT2FNN [16].

$$\begin{aligned}\int_0^\infty S_x^{\,2}\,d\tau &= \int_0^\infty (S_{1x}+rS_{2x})^2\,d\tau \\ &= \int_0^\infty (S_{1x}+rS_x)(S_{1x}+rS_{2x})\,d\tau \\ &= \int_0^\infty (S_{1x}^2+rS_{1x}S_{2x}+rS_{1x}S_{2x}+r^2S_{2x}^2)\,d\tau \\ &= \int_0^\infty (S_{1x}^2+2rS_{1x}S_{2x}+r^2S_{2x}^2)\,d\tau<\infty\end{aligned} \tag{48}$$

Since $2ab \le a^2+b^2$ holds for any real numbers, it is obvious that

$$\int_0^\infty 2rS_{1x}S_{2x}\,d\tau \le \int_0^\infty (S_{1x}^2+r^2S_{2x}^2)\,d\tau \tag{49}$$

which leads to show

$$\int_0^\infty 2rS_{1x}S_{2x}\,d\tau \le \int_0^\infty S_x^2\,d\tau<\infty \Rightarrow \int_0^\infty rS_{1x}S_{2x}\,d\tau<\infty \tag{50}$$

Now, (46) and (48) and the two inequalities $S_{1x}^2 \ge 0$ and $r^2S_{2x}^2 \ge 0$ imply that

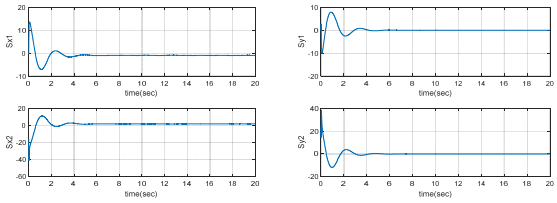
$$\int_0^\infty (S_{1x}^2+r^2S_{2x}^2)d\tau<\infty \;\Rightarrow\; \begin{cases}\int_0^\infty S_{1x}^2 d\tau<\infty \\ \int_0^\infty r^2S_{2x}^2 d\tau<\infty\end{cases} \tag{51}$$

Next, move to prove $|S_{1x}| \in L_\infty$, $|\dot{S}_{1x}| \in L_\infty$, $|S_{2x}| \in L_\infty$, and $|\dot{S}_{2x}| \in L_\infty$. From (51) one obtains $|S_x| \in L_\infty$. At the same time, from the following equation $\dot{V}_2 = S_x\dot{S}_x \le -k_xk_1|S_x| - k_xk_2S_x^{\,2} < \infty$, we can also obtain $|\dot{S}_x| \in L_\infty$. Because the choice of λ_i guarantees $\lambda_i(s_{1i}\cdot s_{2i}) \ge 0$, we obtain $|S_{1x}| \in L_\infty, |S_{2x}| \in L_\infty$, i.e.

$$\sup_{t\ge0}|S_{1x}| = |S_{1x}| < \infty,\ \sup_{t\ge0}|S_{2x}| = |S_{2x}| < \infty \tag{52}$$

From the equation $S_{1x} = \zeta_1$, it follows that

$$\dot{S}_{1x} = u_x + k_{p1}\dot{x}_{1e} - \ddot{x}_d \tag{53}$$



(a) (b)

Fig. 5. Simulated responses of the proposed method using BLS for the ball-riding robot tracking the straight line: (a) S_{x1} and S_{x2} converge to zero; (b) S_{y1} and S_{y2} converge to zero.

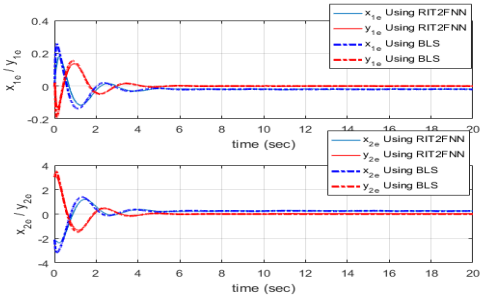


Fig. 6. Comparison of the straight line tracking errors of the proposed method using BLS and the Chan's method using RIT2FNN.

Since all variables in the right-hand side of (53) are bounded. Hence, $|\dot{S}_{1x}|$ is bounded, i.e., $|\dot{S}_{1x}| \in L_\infty$. From $S_{2x} = \zeta_2$, we can obtain $|\dot{S}_{2x}| \in L_\infty$. Because we have proved $|S_{1x}| \in L_\infty$, $|S_{2x}| \in L_\infty$. According to Barbalat's lemma, $\lim_{t\to\infty}|S_{1x}| = 0$ and $\lim_{t\to\infty}|S_{2x}| = 0$, i.e., the first-level sliding surfaces, S_{1x} and S_{2x}, converge asymptotically to zero. These indicate that both variables x and θ_x are stabilized at x_d and θ_{xd}, respectively. The main result is summarized as below.

Theorem 1: *Consider the ball-riding robot dynamic model* (8) *with the proposed control laws* (54)*. Then the second-layer sliding function* $S_x \to 0$ *and the first-layer functions,* S_{1x} *and* S_{2x}*, tend to zero as* $t\to\infty$*. Moreover,* $\theta_x \to \theta_{xd}$*,* $x \to x_d$*, for* $t\to\infty$*.*

Remark 1: Similar results can be applied for the control of the IABRR in the coronal plane. To avoid chattering, we implement the control law (41) by

$$u_x = -\hat{f}_x + \left(k_1 + c_1\left|r_1\ddot{\theta}_{xd} + r_2\ddot{\phi}_{xd}\right| + g\right)sat(S_x) + k_2S_x \tag{54}$$

V. SIMULATION RESULTS AND DISCUSSIONS

To examine the performance and merit of the proposed controller, this section will conduct comparative simulations on straight-line tracking. All computer simulations adopt the parameters listed in Table 1. At first, we select 1 input to use the BLS to estimate the uncertain vector function and increase 1.5 times mass, inertia and frictions at the 10th seconds. The numbers of mapped feature and enhancement nodes are 75, respectively. The simulations are executed using Matlab /Simulink, and the proposed controller is with the following parameter settings; k_{p1}=17.8, k_{p2}=23.1, γ=1, k=0.1. The simulation is conducted to steer the ball-riding robot to move along a straight line starting from the initial position (-0.5m,

0.5m), and the velocity are v_x=0.2m/sec and v_y=0.2m/sec. Fig. 4 shows the simulation results for the proposed method using BLS and the Chan's method with RIT2FNN [16] where the simulation time is 20 seconds. Fig. 5(b) and Fig. 5(c) depict that all the first-layer sliding functions S_{x1}, S_{x2}, S_{y1} and S_{y2} almost approach zero. Fig. 6 illustrates the straight line tracking errors of the proposed method with BLS in comparison with the Chan's method with RIT2FNN, showing that the proposed controller can converge to desired values. The simulation shows that the IABRR can be controlled to track the straight line successfully. Table 2 compares the average execution time and performance indices of the proposed controller and the Chan's controller [16]. The proposed controller using BLS is more efficient than the Chan's controller, and it outperforms the Chan's controller in terms of ISE, IAE, ITAE and ITSE. Obviously, the results in Table 2 indicate that the proposed controller using BLS is superior to the Chan's controller using RIT2FNN.

VI. Conclusions and Future Work

This paper has presented a novel motion control method using BLS for the IABRR with uncertain parameters. This type of controller has been constructed using the Lyapunov stability theory, in order to accomplish motion control trajectory tracking of the robot in presence of parameter variations, exogenous disturbances and terrain-dependent frictions. The performance and merits of the proposed control method has been shown by conducting computer simulations on the IABRR. An interesting topic for future work would be to derive a consensus-based formation control method using BLS for multiple IABRRs.

References

[1] R. Hollis, "Ballbots," *Scientific American Magazine*, pp. 72-77, Oct. 2006.
[2] T. B. Lauwers, G. A. Kantor, and R. L. Hollis, "A Dynamical Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive," *Proceedings of the IEEE Internat. Conf. on Robotics and Automation,* Orlando, USA, pp. 2884-2889, 2006.
[3] U. Nagarajan, A. Mampetta, G. A. Kantor, and R. L. Hollis, "State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile robot" in *Proc.IEEE Int. Conf. Robot. and Autom.*, pp. 998-1003.,2009.
[4] U. Nagarajan, G. A. Kantor, and R. L. Hollis, "Trajectory planning and control of an underactuated dynamically stable single spherical wheeled mobile robot" in *Proc. IEEE Int. Conf. Robot. And Autom.*, pp. 3743-3748, 2009
[5] M. Kumagai, T. Ochiai, "Development of a Robot Balancing on a Ball" in *Proc. IEEE Int. Conf. Contr. Autom. And Systems*, pp. 433-438, 2008
[6] M. Kumagai and T. Ochiai, "Development of a Robot Balancing on a Ball- Application of passive motion transportation," in *Proc. IEEE Int. Conf. Robot. And Autom.*, pp. 4106-4111, 2009.
[7] http://rezero.ethz.ch/project_en.html [online] 2011-07.
[8] C. C. Tsai, C. K. Chan, and L. C. Kuo, "LQR Motion Control of a Ball-Riding Robot," *Proc. of 2012 IEEE International Conference on Advanced Intelligent Mechatronics*, Kaohsiung, Taiwan, July 11-14, 2012.
[9] J.C. Lo and Y. H. Kuo, "Decoupled fuzzy sliding-mode control," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 3, pp. 426-435. Aug. 1998.
[10] C. M. Lin and Y.J. Mon, "Decoupling Control by hierarchical fuzzy sliding-mode controller," *IEEE Transactions on Control System Technology*, vol. 13, no. 4, pp. 593-598. July 2005.
[11] W. Wang, X.D. Liu, and J.Q. Yi, "Structure design of two types of sliding-mode controllers for a class of under-actuated mechanical systems," *IET Proceeding of Control Theory and Applications*, vol.1, no.1, pp. 163-172, Jan. 2007.
[12] H. K. Khalil, *Nonlinear systems,* 2nd ed., Prentice Hall, 1995.
[13] Y.-H. Pao and Y. Takefuji, "Functional-link net computing: theory, system architecture, and functionalities," *Computer*, vol. 25, no. 5, pp. 76–79, 1992.
[14] C. L. P. Chen and Z. L. Liu, "Broad learning system: an effective and efficient incremental learning system without the need for deep architecture," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no.1, pp. 10-24, Jan. 2018.
[15] C. L. P. Chen, Z. L. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," To appear in *IEEE Transactions on Neural Networks and Learning Systems*, 2018.
[16] C. K. Chan and C. C. Tsai, "Intelligent backstepping sliding-mode control using recurrent interval type 2 fuzzy neural networks for a ball-riding robot," *2012 International conference on Fuzzy Theory and Its Applications (iFUZZY2012)*, Taichung, 2012, pp. 169-174

Bing-Yang Chen is currently pursuing his Master's degree in Department of Electrical Engineering from National Chung Hsing University, Taichung, Taiwan, ROC. His current research interests include ball-riding robot, nonlinear control, formation control and their applications to multiple ball-riding robot handling.

Ching-Chih Tsai received the Diplomat in Electrical Engineering from National Taipei Institute of Technology, Taipei, Taiwan, ROC, the MS degree in Control Engineering from National Chiao Tung University, Hsinchu, Taiwan, ROC and the Ph.D degree in Electrical Engineering from Northwestern University, Evanston, IL, USA, in 1981, 1986 and 1991, respectively. Currently, he is currently a Distinguished Professor in the Department of Electrical Engineering, National Chung-Hsing University, Taichung, Taiwan, where he served the Chairman in the Department of Electrical Engineering from 2012 to 2014. He is a Fellow of IEEE, IET and CACS.

Dr. Tsai served as the Chair, Taipei Chapter, IEEE Control Systems Society, from 2000 to 2003, and the Chair, Taipei Chapter, IEEE Robotics and Automation Society from 2005 to 2006. In 2007, he was the program chair of 2007 CACS international automatic conference sponsored by Taipei chapter, IEEE control systems society. In 2010, he served as the program co-chair of SICE 2010 annual conference in Taiwan, which was technically sponsored by IEEE CSS; in 2011, he served as the General Chair, 2011 International conference on service and interactive robotics; in 2012, he has served as the General Chair, 2012 International conference on Fuzzy Theory and Its Applications, the General Chair, 2012-2015 CACS International Automatic Control Conferences, and the General Chair, 2016-2017 International Conference on Advanced Robotics and Intelligent Systems. Dr. Tsai served the two-term President, Chinese Institute of Engineers in Central Taiwan, Taiwan from 2007 to 2011, and two-term President of Chinese Automatic Control Society from 2012 to 2015. Since 2008, he has been the Executive Directors in Boards of Government of thee professional associations, including Robotic Society of Taiwan, Taiwan Fuzzy Systems Association, and Taiwan Systems Association. He has served as the Chair, Taichung Chapter, IEEE Systems, Man, and Cybernetics Society since 2009, the Chair of IEEE SMC Technical Committee on intelligent learning in control systems since 2009, the President of Robotics Society of Taiwan since 2016, the steering committee of Asian Control Association since 2014, a BOG member of IEEE Nanotechnology council since 2012, the Vice President of International Fuzzy Systems Association since 2015, and a BOG member of the IEEE SMCS since 2017.

Dr. Tsai has published more than 500 technical papers, and seven patents in the fields of control theory, systems technology and applications. Web of Science has indexed his paper entitled "Adaptive Neural Network Control of

a Self-Balancing Two-Wheeled Scooter" in the category Automation Control Systems, where the paper was ranked 408 out of 37607 articles (published between 2010 to 2014). Dr. Tsai is respectively the recipients of the Third Society Prize Paper Award from IEEE Industry Application Society in 1998, the Outstanding Automatic Control Engineering Award in 2008 from Chinese Automatic Control Society (CACS), and the Outstanding Engineering Professor Award in 2009 from the Chinese Institute of Engineers in 2009, the IEEE Most Active SMC Technical Committee (TC) Award in 2012 from IEEE SMC Society, the Outstanding Electrical Engineering Professor Award from the Chinese Institute of Electrical Engineering in 2014, Outstanding Industry Contribution Award from Taiwan Systems Association in 2016, the best paper award in the International Journal of Fuzzy Systems in 2017, and many best paper awards from many international conferences technically supported by IEEE. He is the advisor, IEEE SMC student branch chapter at National Chung Hsing University; this chapter was the recipient of certificate of appreciation from IEEE SMCS in 2009. He has served as the associate editors of International Journal of Fuzzy Systems, and IEEE Transactions on Systems, Man and Cybernetics: Systems, IEEE Transactions on Industry Informatics, and International Journal of Electrical Engineering. Recently, he has served as the Editor-in-Chief of a new international robotics journal called “iRobotics”. His current interests include advanced nonlinear control methods, deep model predictive control, fuzzy control, neural-network control, advanced mobile robotics, intelligent service robotics, intelligent mechatronics, intelligent learning control methods with their applications to industrial processes and intelligent machinery.

Feng-Chun Tai received the B.S., M.S. and Ph.D. degrees in Department of Electrical Engineering from National Chung Hsing University, Taichung, Taiwan, ROC. in 2007, 2010 and 2018, respectively. His current research interests include mobile robots, intelligent control, navigation system and their applications to industrial processes and machines.

Adaptive Simultaneous Tracking and Stabilization Using DNA Algorithm for Uncertain Nonholonomic Mobile Robots

Chien-Cheng Yu, Shih-Min Hsieh, and Feng-Chun Tai

***Abstract*—This paper presents an adaptive motion control approach using deoxyribonucleic-acid (DNA) algorithm for simultaneous tracking and stabilization (STS) of uncertain nonholonomic mobile robots with parameter variations. A kinematic STS controller is proposed, and then a DNA computing method is employed to search for its optimal controller parameters in the sense of minimum of integration of squared errors (ISE) or control efforts. An adaptive dynamic controller is synthesized using backstepping approach. Such a dynamic controller is implemented into a high-performance field-programmable gate array (FPGA) chip using hardware/software codesign technique and system-on-a-programmable-chip (SoPC) design concept with a reusable user intellectual property (IP) core library. In addition, a soft-core processor and a real-time operating system (RTOS) are embedded into the same chip for realizing the proposed dynamic control law to steer the mobile platform. Simulation results are conducted to show the effectiveness and merit of the proposed control method in comparison with Dixon's STS kinematic controller. Finally, the performance and applicability of the proposed embedded adaptive controller are exemplified by conducting one experiment on steering an embedded nonholonomic mobile robot to achieve circular trajectory tracking.**

***Index Terms*—Deoxyribonucleic-acid (DNA) algorithm, simultaneous tracking and stabilization (STS), Nonholonomic mobile robot, Backstepping, Regulation, Trajectory tracking.**

I. Introduction

RECENTLY, biological algorithms have gain wide applications to solve the optimization problems. For example, GAs and evolution algorithms (EAs) have been well applied to find optimal solutions in many applications, but they have a chance of converging into the local optimum rather than global optimum [1,2]. To circumvent this shortcoming, several researchers have considered DNA algorithm, firstly proposed by Aldelman [3], as another powerful optimization technique to emulate the concept of the bimolecular evolution. As a new computing paradigm, DNA computing has shown its merits in solving complex problems and attracts attentions from researchers in the control community. At present, DNA computing methods have been applied successfully to solve complex problems and show their excellent performances [3-8]. For example, Lin *et al.* [5] proposed a self-organizing PID control design based on DNA computing method (this method presented how to solve the optimal problem more effectively), Zhu *et al.* [8] presented a DNA algorithm of image recognition based on syntax and its application on isosceles triangle recognition, and Ding *et al.* [9] introduced the DNA genetic algorithm for the design of generalized membership-type Takagi-Sugeno fuzzy control system**.** In [10, 11], DNA algorithms were applied to solve the path planning problem of a mobile manipulator with an omnidirectional mobile platform. However, as our best understanding, DNA computing algorithms in [1-12] have not been yet employed to find optimal parameters of unified motion controllers for simultaneous tracking and stabilization of nonholonomic mobile robots with differential driving.

The control problem of nonholonomic mobile robots has attracted considerable attention in the control community over past and present decades. Since Brockett [13] showed that pure time-invariant state feedback laws do not hold for the nonholonomic mobile robots, many advanced approaches, such as nonlinear control [14-18], sliding-mode control [1, 19-20], fuzzy control [21, 22] and neural control [23, 24], have been presented to solve for the set-point control (regulation) and tracking control problems, respectively. However, these methods could not directly address the STS problem in one control framework, namely that a single controller can be used to solve simultaneously the regulation and tracking problems for nonholonomic mobile robots. To overcome the shortcoming, several researchers have proposed different kinds of unified control methods to achieve simultaneous tracking and stabilization of the nonholonmic mobile robots. For example, Dixon *et al.* [12, 25] proposed a unified kinematic STS controller with exponential stability, Do *et al.* [26] adopted adaptive backstepping technique to construct two STS approaches for a class of nonholonomic mobile robots with differential driving, and Morin and Samson [27] used the transverse function method to establish a unified kinematic STS controller for a more general class of nonholonomic mobile robots. Despite the advent of the aforementioned methods, there remain some improvements on the STS control schemes for nonholonomic mobile robots incorporated with dynamic effects.

Chien-Cheng Yu and Feng-Chun Tai are with the Department of Electrical Engineering, National Chung Hsing University, Taichung City, Taiwan.

Chien-Cheng Yu is also with the Department of Electronic Engineering, Hsiuping University of Science and Technology, Taichung, Taiwan.

Shih-Min Hsieh is with the Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung City, Taiwan.

(Corresponding author Chien-Cheng Yu, email: ccyu@hust.edu.tw)

The authors gratefully acknowledge financial support from the Ministry of Science and Technology, Taiwan, the R.O.C., under contract MOST 104-2221-E-005 -054 -MY2.

The system-on-a-programmable-chip (SoPC) technology has been bringing a major revolution in the design of integrated circuits [28-31]. Compared with the fixed-processor DSP [32], the SoPC technology is capable of not only achieving the same software functions running in its embedded processor but also providing additional hardware IP implementation and an embedded real-time operating system (RTOS) for further purposes. Furthermore, the SoPC technology may offer identical functions to hardware-oriented FPGAs [33-35] because the SoPC evolved from the hardware-oriented FPGA. With the benefits of low cost, low power consumption, small circuit size, IP reusability, and reprogrammable hardware/software codesign, the SoPC technology has been shown as a powerful way to combine flexible software modules and high-performance hardware units for realizing sophisticated but complicated signal processing algorithms, and high-performance but computation-intensive control laws [28-30]. As the authors' best understanding, the SoPC technology has not been applied to design any adaptive trajectory-tracking controllers for nonholonomic mobile platforms and robots yet.

The objectives of this paper are to improve the STS method proposed by Dixon *et al.* [12, 25], to use the DNA computing method to tune the optimal parameters of the proposed controller, to propose an adaptive dynamic controller for the STS problem of the nonholonomic mobile robots, and to implement such a controller in real time by employing the SoPC technology. In comparison with the control laws presented by Dixon *et al.* [12], this paper is written in the following three technical contributions.

1. A general invertible transformation and a more general kinematic STS control method are derived, and a DNA computing algorithm is applied to find the optimal parameters of the proposed kinematic STS controller.
2. Backstepping techniques are used to synthesize the adaptive stable dynamic controller for the robot incorporated with its dynamic effect.
3. An FPGA-based adaptive motion controller is implemented by an embedded processor, an embedded operating system and the SoPC technology. This embedded adaptive motion controller combining the hardware/software co-design and IP re-use concept takes the advantages of efficient implementation, excellent flexibility and satisfactory performance.

The rest of this paper is organized as follows. Section 2 introduces the general invertible transformation and the open-loop error system, develops the proposed kinematic STS controller with globally exponential stability, and finds its optimal control parameters using a DNA computing method. Section 3 derives a stable dynamic controller via backstepping technique together with the Lyapunov stability theory. Section 4 briefly describes the FPGA-based implementation of the adaptive controller for a nonholonomic mobile robot. Simulation results and experiment are presented and discussed in Section 4. Section 5 concludes this paper.

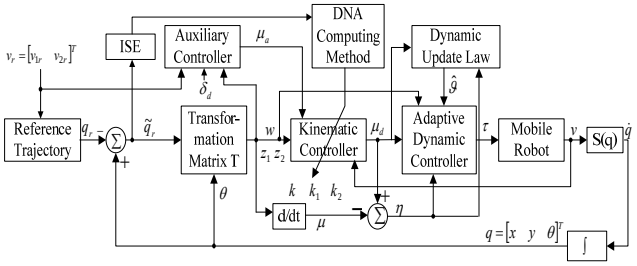


Fig. 1. Block diagram of the proposed adaptive dynamic motion controller for an uncertain nonholonomic mobile robot.

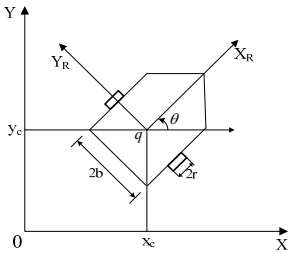


Fig.2. Position and orientation of a nonholonomic mobile robot in the Cartesian coordinate.

II. Modeling and Kinematic Controller Design

In this section, a kinematic controller is first designed to achieve STS, and a DNA computing method is then employed to off-line search for the optimal parameters of the controller in the sense of minimum of integrated squared error (ISE) or control effort. Moreover, the adaptive dynamic controller using backstepping approach will be synthesized to achieve STS in presence of dynamic effects, parameter variations and uncertainties. Figure 1 depicts the block diagram of the proposed adaptive dynamic controller for an uncertain nonholonomic mobile robot.

A. Kinematic Model and Problem Formulation

To steer the nonholonomic mobile robot effectively, it is necessary to have the kinematic and dynamic models of the robot in order to design controllers to satisfactorily accomplish out desired control objectives. In this subsection, the kinematic model of the nonholonomic mobile robot is briefly described and the design goal of the DNA-based kinematic controller is then stated.

1) Kinematic Model

Under the assumption of pure rolling, the kinematic model of the nonholonomic mobile robot, shown in Fig. 2, can be given as follows:

$$\dot{q} = S(q)v \tag{1}$$

where $q(t), \dot{q}(t) \in R^3$ are defined as $q = [x_c, y_c, \theta]^T$ and $\dot{q} = [\dot{x}_c, \dot{y}_c, \theta]^T$; $x_c(t)$ and $y_c(t)$ denote the reference position of the nonholonomic mobile robot with respect to the Cartesian coordinates, and $\theta(t) \in R^1$ is the orientation of the nonholonomic mobile robot. $\dot{x}_c(t)$ and $\dot{y}_c(t)$ denote the corresponding linear velocity. The matrix $S(q) \in R^{3\times 2}$ is given

by

$$S(q)=\begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix} \tag{2}$$

Furthermore, the velocity vector $v(t)\in R^2$ is defined as $v=\begin{bmatrix}v_1 & v_2\end{bmatrix}^T$, $v_1(t)\in R^1$ represents the linear speed, $v_2(t)\in R^1$ denotes the angular velocity.

2) Problem Formulation

The design goal of the proposed DNA-based kinematic STS control for the nonholonomic mobile robot is to let the trajectories of the nonholonomic mobile robot exponentially follow the desired positions and orientations of a fixed reference configuration and a time-varying reference trajectory in one unified control framework. To formulate the problem, let $\tilde{q}=\left[\tilde{x}(t),\ \tilde{y}(t),\ \tilde{\theta}(t)\right]^T$ be the differences between the real position $x_c(t)$, $y_c(t)$ and the angle $\theta(t)$ of the nonholonomic mobile robot with the desired reference trajectory, $q_r(t)=\left[x_{rc}(t), y_{rc}(t), \theta_r(t)\right]^T\in R^3$, in the Cartesian coordinate, i.e.,

$$\begin{aligned}\tilde{x}(t)&=x_c(t)-x_{rc}(t)\\ \tilde{y}(t)&=y_c(t)-y_{rc}(t)\\ \tilde{\theta}(t)&=\theta(t)-\theta_r(t)\end{aligned} \tag{3}$$

Moreover, the desired reference trajectory $q_r(t)$ satisfies the following equation:

$$\dot{q}_r(t)=S(q_r)v_r \tag{4}$$

where $S(\cdot)$ is also defined in (2), $v_r(t)=\left[v_{1r}(t)\ \ v_{2r}(t)\right]^T\in R^2$ represents the desired time-varying linear and angular velocities.

The aim of the proposed STS method is to design the control $v(t)$ such that $\left|\tilde{x}(t)\right|,\ \left|\tilde{y}(t)\right|,\ \left|\tilde{\theta}(t)\right|\le\alpha\ e^{-\beta}$, where α and β are positive constants. Worthy of mention is that the desired reference trajectory $q_r(t)$ is a function of time (or a fixed reference configuration), then the simultaneous regulation and trajectory tracking control method is reduced to the so-called trajectory tracking method (or the regulation method).

B. STS Kinematic Controller design

This subsection is devoted to synthesizing a STS kinematic motion controller whose parameters are searched by a DNA computing method. In Fig. 1, the proposed kinematic controller is composed of a globally invertible transformation and a STS control law with the property of exponential stability. Below is the detailed description of the proposed kinematic controller design.

1) Open-Loop Error Control System and Model Transformation

To achieve simultaneous stabilization and exponential stability analysis of both the regulator and tracking controller, we define a new globally invertible transformation as follows:

$$\begin{bmatrix}w\\ z_1\\ z_2\end{bmatrix}=\begin{bmatrix}-k\tilde{\theta}\cos\theta+2k\sin\theta & -k\tilde{\theta}\sin\theta-2k\cos\theta & 0\\ \cos\theta & \sin\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\tilde{x}\\ \tilde{y}\\ \tilde{\theta}\end{bmatrix} \tag{5}$$

where $k>0$, $w(t)\in R^1$ and $z(t)=\left[z_1(t)\ \ z_2(t)\right]^T\in R^2$ are the auxiliary tracking error vectors, and $\tilde{x}(t)$, $\tilde{y}(t)$, $\tilde{\theta}(t)\in R^1$ are defined in (3). From (5), we have the inverse transformation

$$\begin{bmatrix}\tilde{x}\\ \tilde{y}\\ \tilde{\theta}\end{bmatrix}=\begin{bmatrix}\frac{1}{2k}\sin\theta & \frac{1}{2k}(z_2\sin\theta+2k\cos\theta) & 0\\ \frac{-1}{2k}\cos\theta & \frac{-1}{2k}(z_2\cos\theta+\sin\theta) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}w\\ z_1\\ z_2\end{bmatrix} \tag{6}$$

Note that, from (6), if $w(t)$, $z_1(t)$, $z_2(t)\in L_\infty$, then $\tilde{x}(t)$, $\tilde{y}(t)$, $\tilde{\theta}(t)\in L_\infty$. Moreover, it is easy to claim that $\lim_{t\to\infty}w(t)=0$, $\lim_{t\to\infty}z_1(t)=0$, $\lim_{t\to\infty}z_2(t)=0$, then

$$\lim_{t\to\infty}\tilde{x}(t)=0,\ \lim_{t\to\infty}\tilde{y}(t)=0,\ \lim_{t\to\infty}\tilde{\theta}(t)=0 \tag{7}$$

and $\left|w(t)\right|\le\alpha_0\ e^{-\alpha_1 t}$, $\left|z_1(t)\right|\le\alpha_0\ e^{-\alpha_1 t}$, $\left|z_2(t)\right|\le\alpha_0\ e^{-\alpha_1 t}$ implies

$$\left|\tilde{x}(t)\right|\le\alpha_2\ e^{-\alpha_3 t},\ \left|\tilde{y}(t)\right|\le\alpha_2\ e^{-\alpha_3 t},\ \left|\tilde{\theta}(t)\right|\le\alpha_2\ e^{-\alpha_3 t} \tag{8}$$

where α_0, α_1, α_2, α_3 are positive constants. Furthermore, we take the time derivative of (5) and then use (1-4), to obtain the following expressions:

$$\begin{aligned}\dot{w}=&\left(\begin{bmatrix}k & k\left(-\tilde{x}\sin\theta+\tilde{y}\cos\theta\right)\\ 0 & k\end{bmatrix}\begin{bmatrix}v_1\\ v_2\end{bmatrix}\right)^T\\ &-\begin{bmatrix}kv_{1r}\cos z_2\\ kv_{2r}\end{bmatrix}\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix}+2k\left(z_1v_{2r}-v_{1r}\sin z_2\right)\end{aligned} \tag{9}$$

$$\dot{z}=\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\end{bmatrix}=\begin{bmatrix}1 & -\left(\tilde{x}\sin\theta-\tilde{y}\cos\theta\right)\\ 0 & 1\end{bmatrix}\begin{bmatrix}v_1\\ v_2\end{bmatrix}-\begin{bmatrix}v_{1r}\cos z_2\\ v_{2r}\end{bmatrix} \tag{10}$$

Combining (9) and (10), the dynamics of the tracking error becomes

$$\begin{aligned}\dot{w}&=k\mu^T J^T Z+kf\\ \dot{z}&=\mu\end{aligned} \tag{11}$$

where $J^T\in R^{2\times 2}$ is the skew-symmetry constant matrix defined by

$$J^T = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{12}$$

and $f \in R^1$ is the bounded auxiliary signal [3] given by

$$f = 2(z_1 v_{2r} - v_{1r} \sin z_2) \tag{13}$$

Further, $\mu(t) = \left[\mu_1(t) \;\; \mu_2(t)\right]^T \in R^2$ is called the auxiliary control input having the following form

$$\mu = T^{-1} v - \begin{bmatrix} v_{1r} \cos z_2 \\ v_{2r} \end{bmatrix} \tag{14}$$

where $T(t) \in R^{2\times 2}$ is expressed by

$$T = \begin{bmatrix} 1 & \left(\tilde{x}\sin\theta - \tilde{y}\cos\theta\right) \\ 0 & 1 \end{bmatrix} \tag{15}$$

Note that, from (14), the actual control vector becomes

$$v = T\left(\mu + \begin{bmatrix} v_{1r} \cos z_2 \\ v_{2r} \end{bmatrix}\right) \tag{16}$$

2) Kinematic Controller Synthesis

The primary study of the STS problem encountered for the nonholonomic mobile robot is to design a STS kinematic controller via backstepping approach based on the kinematic model (11). To facilitate the design process, one defines an auxiliary error signal $\tilde{z}(t) \in R^2$, where $\tilde{z}(t)$ is the difference between the auxiliary signal $z_d(t) \in R^2$ and the auxiliary tracking error vector $z(t)$, i.e.,

$$\tilde{z}(t) = z_d - z \tag{17}$$

To stabilize the open-loop error system of (11), one further defines another auxiliary control signal $\mu(t)$ in (14) by

$$\mu = \mu_a - k_2 z \tag{18}$$

where the control term $\mu_a(t) \in R^2$ is given by

$$\mu_a = \left(\frac{k_1 w + f}{\delta_d^2}\right) J z_d + \Omega_1 z_d \tag{19}$$

Hence, the auxiliary signal $z_d(t)$ in (17) can be given by

$$\dot{z}_d = \left(\frac{k_1 w + f}{\delta_d^2} + kw\Omega_1\right) J z_d + \frac{\dot{\delta}_d}{\delta_d}\, \mathrm{z_d} \;,\; z_d^T z_d(0) = \delta_d^2(0) \tag{20}$$

where the auxiliary control terms $\Omega_1(t) \in R^1$ and $\delta_d(t) \in R^1$ are expressed by

$$\Omega_1 = k_2 + \frac{\dot{\delta}_d}{\delta_d} + kw\left(\frac{k_1 w + f}{\delta_d^2}\right) \tag{21}$$

$$\delta_d = \alpha_0 e^{-\alpha_1 t} \tag{22}$$

where k, k_1, k_2, α_0, $\alpha_1 \in R^1$ are positive constant control gains.

Differentiating $z_d^T(t) z_d(t)$ yields

$$\frac{d}{dt}(z_d^T z_d) = 2 z_d^T \left[\left(\frac{k_1 w + f}{\delta_d^2} + kw\Omega_1\right) J z_d + \frac{\dot{\delta}_d}{\delta_d} z_d\right] \tag{23}$$

From (12) and with the properties of the skew-symmetry constant matrix *J*, (23) turns out

$$\frac{d}{dt}(z_d^T z_d) = 2\frac{\dot{\delta}_d}{\delta_d} z_d^T z_d \tag{24}$$

which leads to obtain the unique solution of (24) as follows:

$$z_d^T(t) z_d(t) = \left\|z_d(t)\right\|^2 = \delta_d^2(t) \tag{25}$$

where $\|\bullet\|$ denotes the standard Euclidean norm.

To develop the closed-loop error control system of (11), one substitutes (18) into (11) and then uses (17) to obtain

$$\dot{w} = k\mu_a^T J\tilde{z} + k\mu_a^T J^T z_d + kf \tag{26}$$

Furthermore, by substituting (19) into (26) and using (25), the closed-loop error system for $w(t)$ is obtained from

$$\dot{w} = k\mu_a^T J\tilde{z} - kk_1 w \tag{27}$$

Next, differentiating $\tilde{z}(t)$ with respect to time, and substituting $\mu_a(t)$ in (19) and $\dot{z}$ in (11) into $\dot{\tilde{z}}(t)$ and utilizing (19-21), one obtains the dynamics of the closed-loop error system for $\tilde{z}(t)$ governed by

$$\dot{\tilde{z}} = kwJ\mu_a - k_2\tilde{z} \tag{28}$$

The globally exponential stability of the aforementioned closed-loop error system can be easily proven by defining a radial, unbounded and quadratic Lyapunov function denoted by $V(\mathrm{t}) \in R^1$ as follows:

$$V(t) = \frac{1}{2}w^2 + \frac{1}{2}\tilde{z}^T\tilde{z} \tag{29}$$

This main result is summarized as below.

Theorem 1 *Let* k, k_1, k_2, α_1 *be real positive constants satisfying* $\min(kk_1, k_2) > \alpha_1$. *Then the* control *laws (18-22) make the closed-loop error system globally exponentially stable, i.e.,* $\left|\tilde{x}(t)\right|, \left|\tilde{y}(t)\right|, \left|\tilde{\theta}(t)\right| \le \alpha_2 e^{-\alpha_3 t}$, *where* $\alpha_2 \in R^1$ *and* $\alpha_3 \in R^1$ *are positive constants.*

C. Parameters Selection Using DNA Computing Method

This subsection aims to use a DNA computing method to select the optimal parameters of the proposed kinematic controller. In what follows, a DNA computing method with its coding scheme and operators is first delineated, and the proposed DNA computing algorithm for finding the optimal control parameters is then developed.

1) DNA Computing

DNA computing is a parallel, global optimization method with the search strategy similar to GAs. DNA computing consists of four bases of molecule structure, named Adenine (A), Guanine (G), Cytosine (C) and Thymine (T). In particular, DNA computing provides two new operators, enzyme and virus, which are very useful to enhance the effect of mutation. Moreover, the coding schemes of DNA computing algorithms are quite different from those of GAs. In what follows the core operators of DNA algorithms are recalled in some detail.

2) Coding Scheme

DNA algorithms use A, G, T and C to stand for their chromosome; for example, one can define A=0, G=1, T=2, and C=3. Moreover, (30) can easily be applied to define the range and precision of a parameter in the DNA algorithm [5, 11].

$$\pi = \frac{U_{\max} - U_{\min}}{4^l - 1} \tag{30}$$

where π stands for the precision, l stands for how many bits will be used, $U_{\max}$ is the maximum of the parameter, and $U_{\min}$ denotes the minimum of the parameter. Worthy of mention is that (30) hinges on the fact that DNA algorithms use four bits for the coding scheme, but GAs only use two bits.

3) Selection

The main task of selection module is to select individuals from the populations so that these individuals can be sent to the crossover and mutation module in order to attain new offspring. There are several selection methods with different characteristics, such as roulette selection, rank selection and tournament selection. Selection is one of the key operators that ensure survival of the fitness. It is necessary to get the fitness value of each chromosome prior to selection process.

4) Crossover

Crossover is the fundamental mechanism of genetic rearrangement in DNA algorithm. This is done by the exchange of strings between the two parent chromosomes from the selection module. Crossover occurs when two chromosomes break and then reconnect but to the different end piece. There are various crossover schemes such as one-point crossover, two-point crossover, uniform crossover and arithmetic crossover.

5) Mutation (Enzyme and Virus Operation)

Mutation is the process which consists of making small alterations to the bits of the chromosomes by applying some kind of randomized changes, such as single-point or multi-point mutation process. In DNA algorithms, there are two special mutation operators, enzyme and virus, which are more effective than GAs. The enzyme operator refers to deletion, in which one or more base pairs are removed and the virus operator refers to insertion, in which one or more base pairs are inserted into sequence. These two operators are used to reduce or enlarge the chromosome sequences in order to change the step length of searching [11]; thereby diverse the searching space in DNA, however, the step length should be small.

6) Fitness function

The fitness function reported here is based on the integrated squared error (ISE) regarding the three variables $\tilde{x}$, $\tilde{y}$ and $\tilde{\theta}$, or the control effort about the linear and angular velocity commands, i.e.,

$$FISE = CISE - ISE, \quad ISE = \int_0^T \left(\tilde{x}^2(t) + \tilde{y}^2(t) + \tilde{\theta}^2(t) \right) dt \tag{31}$$

$$FCE = CCE - CE, \quad CE = \int_0^T \left(v_1^2(t) + v_2^2(t) \right) dt \tag{32}$$

where T is fixed and selected by designers, and the two positive and real constants, C_{ISE} and C_{CE}, are chosen such that F_{ISE} and F_{CE} are always positive. The goal of the controllers' parameters search is to optimally select the three best parameters k, k_1, k_2 using the DNA computing method so as to minimize the fitness function in (31) or (32).

7) Proposed DNA Computing Algorithm

The DNA algorithm for finding the optimal controller parameters problem with the fitness function mentioned in (31) or (32) is described by the following steps.

Step 1: Randomly generate the parameters k, k_1, k_2 by chromosomes and make sure that all of them fit with the basic requirements. Notice that A, T, G, and C respectively represent the chromosomes in the DNA algorithm.

Step 2: Set the number for the first generation (parents).

Step 3: Calculate the fitness of the chromosomes, while the objective is not achieved, the DNA computing process cannot stop, and the aforementioned operations of selection, crossover and mutations mentioned are implemented.

Step 4: Execute the procedure of crossover and also check whether the new chromosome is acceptable. If the new chromosome does not satisfy the requirement, this procedure must be repeated until an acceptable chromosome is obtained.

Step 5: Perform mutation process (enzyme and virus operation) with low mutation rate, and ensure that new chromosomes are reasonable such that the fitness function mentioned in (31) or (32) is minimized.

Step 6: Redo these three steps from Step 3 to Step 5 until the specified condition is reached.

III. Dynamic Controller Design and Implementation

Once the controller with the optimal parameters found by the DNA computing algorithm has been developed, the dynamic controller will be synthesized to achieve STS in presence of parameter variations and uncertainties. By Newtonian mechanics, equations of motion of the nonholonomic mobile robot with static frictions can be easily obtained. Afterwards, the well-known backstepping technique together with the Lyapunov stability theory is employed to derive a stable dynamic controller. Detailed closed-loop stability analysis is also investigated as well.

A. Dynamic Model

The dynamic model of the nonholonomic mobile robot navigating in a flat terrain can be simply modified from [1] and expressed in the following form

$$M\dot{v} + C(q,\dot{q})v + F(\dot{q}) + G(q) = B\tau \tag{33}$$

where

$$M = \begin{bmatrix} m_c + 2m_w + 2r^{-2}I_w & 0 \\ 0 & 2m_w b^2 + 2I_m + 2b^2 r^{-2} I_w \end{bmatrix}$$

$$C(q,\dot{q}) = 0$$

$$F(\dot{q}) = \begin{bmatrix} F_{s1} & 0 \\ 0 & F_{s2} \end{bmatrix} \begin{bmatrix} \mathrm{sgn}(v_1) \\ \mathrm{sgn}(v_2) \end{bmatrix} \quad B = \begin{bmatrix} r^{-1} & r^{-1} \\ r^{-1}b & -r^{-1}b \end{bmatrix}$$

$\dot{v}(t) \in R^2$ denotes the time derivative of v defined in (1), $M \in R^{2\times 2}$ represents the symmetric and positive-definite inertia matrix, $C(\dot{q}) \in R^{2\times 2}$ represents the centripetal and Coriolis matrix, $F(\dot{q}) \in R^{2\times 1}$ denotes the surface friction, $B \in R^{2\times 2}$ represents the input transform matrix, and $\tau(t) \in R^2$ represents the torque input vector. Assume that the mobile robot moves in the horizontal plane; thus, the gravitational vector $G(q)$ will be zero. F_{s1} and F_{s2} are static friction elements, b is half of the width of the mobile robot, r is the radius of the wheel, m_c and m_w are the mass of the body and wheel with a motor. I_w and I_m are the wheel with a motor about the wheel axis, and the wheel with a motor about the wheel diameter, respectively.

Pre-multiplying (33) by $T^T(t)$ defined in (15) and using(1) to obtain

$$T^T M(\dot{T}\mu + T\dot{\mu} + \dot{\Pi}) + T^T F(T\mu + \Pi) = T^T B\tau \tag{34}$$

which leads to the following model:

$$\bar{M}\dot{\mu} + \bar{V}_m \mu + \bar{N} = \bar{B}\tau \tag{35}$$

where

$$\begin{aligned} &\bar{M} = T^T M T, \ \ \bar{B} = T^T B, \ \ \bar{V}_m = T^T M \dot{T}, \\ &\bar{N} = T^T M \dot{\Pi} + T^T F(T\mu + \Pi) \end{aligned} \tag{36}$$

and

$$\Pi = \begin{bmatrix} v_{1r}\cos z_2 + v_{2r}(\tilde{x}\sin\theta - \tilde{y}\cos\theta) \\ v_{2r} \end{bmatrix} \tag{37}$$

Note that $\bar{M}$ is a symmetric and positive-definite matrix, and the matrix $\frac{1}{2}\dot{\bar{M}} - \bar{V}_m$ is skew-symmetric [25].

B. Dynamic Controller Synthesis

In this subsection, one new control objective is to design an adaptive tracking controller for the transformed nonholonomic mobile robot model. Before doing so, the previous kinematic control signal is used as the desired control signal, i.e.,

$$\mu_d = \mu_a - k_2 z \tag{38}$$

For the convenience of designing the controller and theoretical analysis of the nonholonomic mobile robot, it is necessary to define an auxiliary backstepping error signal $\eta(t) \in R^2$ as follows:

$$\eta = \mu_d - \mu \tag{39}$$

To achieve globally asymptotically stable control for the robot, we propose the input torque $\tau(t) \in R$ given by

$$\tau = \bar{B}^{-1}(Y\hat{\vartheta} + k_3\eta + kJzw + \tilde{z}) \tag{40}$$

where $k_3 \in R$ is a positive definite diagonal gain matrix; $\hat{\vartheta}(t) \in R^p$ denotes the parameter estimate of ϑ, and it is calculated on-line via the following update law:

$$\dot{\hat{\vartheta}} = \Gamma Y^T \eta \tag{41}$$

where Γ is a diagonal matrix, $Y(\dot{\mu}_d, \mu_d, \mu, t)$ is the regression matrix defined as follows:

$$Y\vartheta = \bar{M}\dot{\mu}_d + \bar{V}_m \mu_d + \bar{N} \tag{42}$$

To quantify the performance of the adaptation algorithm, we define the parameter estimation error signal $\tilde{\vartheta}(t) \in R^p$, defined as follows

$$\tilde{\vartheta} = \vartheta - \hat{\vartheta} \tag{43}$$

C. Closed-Loop Error Dynamics

In order to develop the closed-loop error system for $w(t)$, $\tilde{z}(t)$, $\eta(t)$ and $\tilde{\vartheta}(t)$, it is necessary to substitute (38) and (39) into (11) and then have

$$\dot{w} = k\mu^T J^T z + kf = -kk_1 w + k\mu_d^T J\tilde{z} + k\eta^T Jz \tag{44}$$

$$\dot{\tilde{z}} = kwJ\mu_a - k_2\tilde{z} + \eta \tag{45}$$

By taking the time derivative of (39) and using (34) and (42), one obtains the following closed-loop error dynamics for

$\eta(t)$

$$\begin{aligned}\dot{\eta} &= \dot{\mu}_d - \dot{\mu} = \bar{M}^{-1}(-V_m\eta + Y\tilde{\vartheta} + Y\hat{\vartheta} - \bar{B}\tau) \\ &= \bar{M}^{-1}(-V_m\eta + Y\tilde{\vartheta} + Y\hat{\vartheta} - (Y\hat{\vartheta} + k_3\eta + kJzw + \tilde{z})) \\ &= \bar{M}^{-1}(-V_m\eta + Y\tilde{\vartheta} - k_3\eta - kJzw - \tilde{z})\end{aligned} \tag{46}$$

D. Stability Analysis

From (44-46), the aforementioned adaptive control laws (40-41) are shown globally asymptotically stable by using the following theorem.

Theorem 2 *Given the robot's dynamic model (33), the control law (40) together with the parameter updating laws (41-42) will make the closed-loop error system globally asymptotically stable,*

$$\lim_{t\to\infty} \tilde{x}(t) = 0,\ \lim_{t\to\infty} \tilde{y}(t) = 0,\ \lim_{t\to\infty} \tilde{\theta}(t) = 0 \tag{47}$$

namely that the origin is globally asymptotically stable for the closed-loop error control system of the robot with dynamic effect.

Proof: First we select a Lyapunov function $V(t) \in R^1$ as follows:

$$V = \frac{1}{2}w^2 + \frac{1}{2}\tilde{z}^T\tilde{z} + \frac{1}{2}\eta^T\bar{M}\eta + \frac{1}{2}\tilde{\vartheta}^T\Gamma^T\tilde{\vartheta} \tag{48}$$

Differentiating (48) with respect to time, taking (44-46) into $V(t)$, and using the fact that $\dot{\tilde{\vartheta}} = -\dot{\hat{\vartheta}}$ yield the following expression.

$$\begin{aligned}\dot{V} &= w(-kk_1w + k\mu_a{}^T J\tilde{z} + k\eta^T Jz) + \tilde{z}(kwJ\mu_a - k_2\tilde{z} + \eta) \\ &\quad + \eta^T(-\bar{V}_m\eta + Y\tilde{\vartheta} + Y\hat{\vartheta} - \bar{B}\tau) + \frac{1}{2}\bar{\eta}^T\dot{\bar{M}}\eta - \tilde{\vartheta}^T\Gamma^{-1}\dot{\hat{\vartheta}} \\ &= -kk_1w^2 - k_2\tilde{z}^T\tilde{z} - k_3\bar{\eta}^T\eta + \eta^T(\frac{1}{2}\dot{\bar{M}} - \bar{V}_m)\eta \\ &\quad + \tilde{\vartheta}^T(Y^T\eta - \Gamma^{-1}\dot{\hat{\vartheta}})\end{aligned} \tag{49}$$

With (41) and the skew symmetry of $\frac{1}{2}\left(\dot{\bar{M}} - \bar{V}_m\right)$, (49) turns out

$$\dot{V} = -kk_1w^2 - k_2\tilde{z}^T\tilde{z} - k_3\eta^T\eta \le 0 \tag{50}$$

which implies that the variables w, $\tilde{z}$ and η are shown uniformly bounded by Lyapunov stability theory. Moreover, w, $\tilde{z}$ and η approach zero as time tends to infinity.

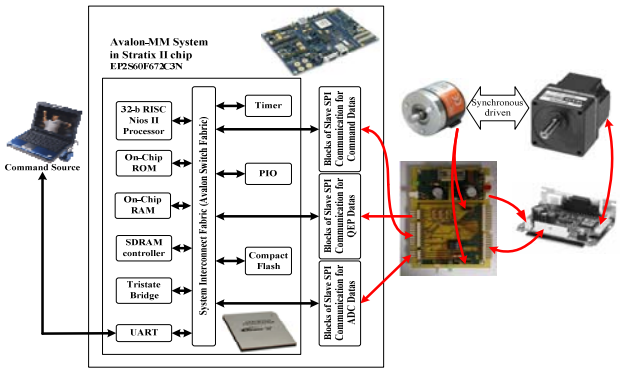


Fig. 3 FPGA implementation of the proposed adaptive motion controller

IV. FPGA-Based Implementation of the Proposed Adaptive Controller

This section is devoted to using the SoPC technology to implement the proposed adaptive motion control law of the nonholonomic mobile robots. Figure 3 shows the architecture of the FPGA implementation for the proposed adaptive motion controller. As shown in Fig. 3, the Avalon Memory-Mapped (Avalon-MM) interface is an interface protocol for use in connecting master and slave components in an SOPC Builder system. The protocol connects address-based read/write interfaces typical of an Avalon Memory-Mapped master that usually controls a number of Avalon Memory-Mapped slave peripherals. In general, the Avalon-MM master is a microprocessor (Nios II), and slaves include memories, UARTs and timers, etc. Furthermore, Avalon-MM master and slave ports do not connect together directly. Instead, Avalon-MM ports connect to system interconnect fabric (Avalon switch fabric) and the system interconnect fabric translates signals between master ports and slave ports, i.e., the Avalon switch fabric is the collection of signals and on-chip interconnect logic, including address decoding, data-path multiplexing, wait-state generation, arbitration, interrupt controller, and data-width matching, etc.

In Fig. 3, the adaptive control law for the mobile platform has been implemented into the 32-b Nios II processor whose numerical precision and computation speed are high enough to realize the adaptive control law. The Nios II processor accesses the control and status registers of on-chip components using an Avalon-MM interface. The software-based adaptive controller and hardware-based custom logic are connected to the system interconnect fabric via Avalon-MM for achieving the adaptive controller in the Altera FPGA chip. The parallel input/output (PIO) core provides a memory-mapped interface between an Avalon Memory-Mapped (Avalon-MM) slave port and general-purpose I/O ports. The I/O ports connect either to on-chip user logic, or to I/O pins that connect to devices external to the FPGA. The user IP cores (custom logic) for this robotic application have been developed by VHDL (VHSIC Hardware Description Language).

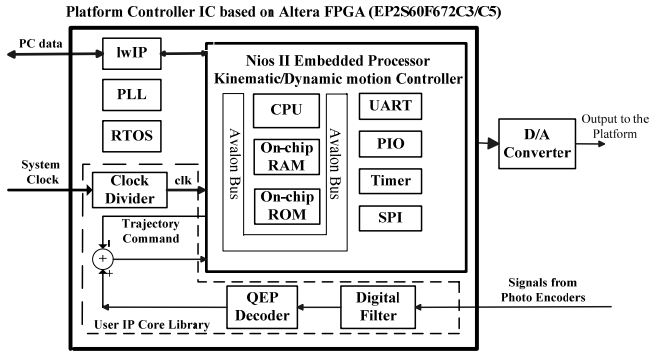


Fig. 4 Embedded adaptive controller of the nonholonomic mobile platform in Altera FPGA

The adaptive motion control system with a FPGA-based controller using the SoPC technology has been efficiently implemented in an Altera Stratix II FPGA chip using a 32-b Nios II processor. Figure 4 shows the flowchart of the proposed controller and the peripheral hardware. As shown in Fig. 4, the 12-b D/A converter, MCP4822, is employed to convert the output commands into analog voltage signals for driving two dc brushless motors mounted on the two nonholonomic wheels. The user IP core library (custom logic) connected to the Altera system interconnect fabric via Avalon-MM for the mobile platform controller in the FPGA chip. The clock divider module is mainly employed to generate the desired 50/2N-MHz clock frequencies to other clock domain modules in the controller, where N is an arbitrary integer. Photo encoders are often integrated in motor assembly to determine the current position of the motors by quadrature-encoder-pulse (QEP) processing. Because the photo encoders are very sensitive to their working environments, the measurement noise must be filtered out by a digital filter prior to the QEP in order to obtain exact readings.

The QEP module is used for direct interface with a linear or rotary incremental encoder to obtain its position, direction, and speed information for use in a high-performance motion and position control system. The two QEP signals generated from the two motors are fed back to the embedded controller. With the QEP signals, the real position and orientation of the mobile platform can be estimated directly by the embedded Nios II processor. The QEP decoder circuit was implemented by VHDL, and the real-time OS MicroC/OS-II was ported into the FPGA chip to handle the data communication with a personal computer (PC) via TCP/IP protocol. Moreover, the embedded soft-core Nios II processor works with the lwIP (lightweight IP) for the Ethernet connectivity, thereby significantly reducing resource usage. The FPGA chip is Altera Stratix EP2S60F672C3N with 60 440 logic elements (LEs), 718 user I/O pins, 36 DSP blocks, 2 544 192 RAM bit memory, 12 phase-locked loops (PLLs), and an embedded Nios II 32-b RISC (Reduced Instruction Set Computer) processor. Furthermore, the resource usage of the proposed adaptive controller IC is 6242 LEs, 3164 registers, 225 user I/O pins, 8 DSP blocks, 1 phase-locked loops (PLLs), and 56 192 memory bits. With the hardware/software codesign and SoPC technology, the adaptive controller takes the advantage of software flexibility for complicated algorithm with low sampling frequency in motion control (less than 1 kHz) and high sampling frequency required in hardware IP (greater than 1 MHz) [36].

V. Simulations, Experimental Results and Discussion

In this section, four simulations and one experiment are conducted to illustrate the feasibility, performance and merit of the embedded adaptive dynamic motion controller. The experiment is used for illustration of effectiveness of the embedded adaptive dynamic motion controller.

A. Simulation Results and Discussion

This subsection will present Matlab/Simulink-based simulation results of an adaptive dynamic motion controller using the DNA computing method for STS of nonholonomic mobile robots with differential-driving. The performance and merit of the proposed control method are exemplified by conducting four simulations on regulation and trajectory tracking. In both simulations, the fitness function is evaluated based on the ISE in (31), and the dynamical behavior of the nonholonomic mobile robot is governed as follows:

$$\begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix}+\begin{bmatrix} F_{s1} & 0 \\ 0 & F_{s2} \end{bmatrix}\begin{bmatrix} \mathrm{sgn}(v_1) \\ \mathrm{sgn}(v_2) \end{bmatrix}=\begin{bmatrix} 1/r & 1/r \\ b/r & -b/r \end{bmatrix}\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (51)$$

where $\mathrm{M} = m_c + 2\mathrm{m_w} + 2\mathrm{r}^{-2}\mathrm{I_w} = 165\mathrm{kg}$ is the mass of the robot; $\mathrm{I} = 2\,\mathrm{m_w b}^2 + 2\,\mathrm{I_m} + 2\,\mathrm{b}^2\mathrm{r}^{-2}\mathrm{I_w} = 4.643\,\mathrm{Kg\text{-}m}^2$ is the inertia of the robot; r =0.075m is the radius of the wheels; b =0.667m is the length of the axis between the wheels; $F_{S1} = 10\mathrm{N}$ and $F_{S2} = 10\mathrm{N}$ are static friction elements; τ_1 and τ_2 respectively represent the two torque inputs for both wheels.

In the simulations, the mobile robot is assumed to get started at the initial posture $(0\mathrm{m},\ 0\mathrm{m},\ \pi/6\ \mathrm{rad})$; the reference trajectory for trajectory tracking is assumed to start at $(2.0\mathrm{m},\ 3.0\mathrm{m},\ \pi/3\ \mathrm{rad})$ and the desired line trajectory is given by $(x(t),\ y(t),\ \theta(t)) = (x_0 + v_r t,\ y_0 + v_r t,\ \theta_0 + \omega_r t)$ where $v_r = 0.6$ m/s, $\omega_r = 0$ for $t \le 5$ s, and the desired circular trajectory is obtained from (4) with $v_r = 0.6$ m/s, and $\omega_r = 0.2$ rad/s for t >5s such that the initial values of the auxiliary signals are $w(0) = 1.5348$, $\tilde{z}_1(0) = 0.5$, $\tilde{z}_2(0) = 0.9912$. Therefore, the controller's parameters for the virtual motion command generator are set by $\alpha_0 = \sqrt{\left((0-2)\cos(\pi/6)+(0-2)\sin(\pi/6)^2+(\pi/6-\pi/3)^2\right)}$, $\alpha_1 = 0.2$ and $z_d = \left[(0-2)\cos(\pi/6)+(0-3)\sin(\pi/6)\quad(\pi/6-\pi/3)\right]^T$

Furthermore, in the simulations, the parameters of DNA computing method are set as follows: the low mutation rate is 0.05; the precision of DNA computing is 10000; the operator points of crossover, mutation, enzyme, and virus are set by 2, 3,

3, and 3, respectively. The use of the proposed DNA computing method obtains the optimal values of k, k_1, k_2; the DNA computing method was terminated at 60 seconds, thus giving that the optimal controller parameters $k = 0.8053$, $k_1 = 1.3732$, $k_2 = 9.5769$, which satisfy $\min(kk_1, k_2) > \alpha_1$. Theorem 2 infers that the closed-loop error control system is globally exponentially stable. Figure 5 depicts the behavior of the ISE-based fitness values of the proposed STS method with the DNA computing method in ten generations, thereby showing that the optimal fitness value is 22.5411. Worthy of mention is that the simulation result in Fig. 5 is based on the line-circular trajectory tracking, as shown in the following second simulations, and the three optimal controller parameters, $k = 0.8053$, $k_1 = 1.3732$, and $k_2 = 9.5769$, are applied to the following four simulations.

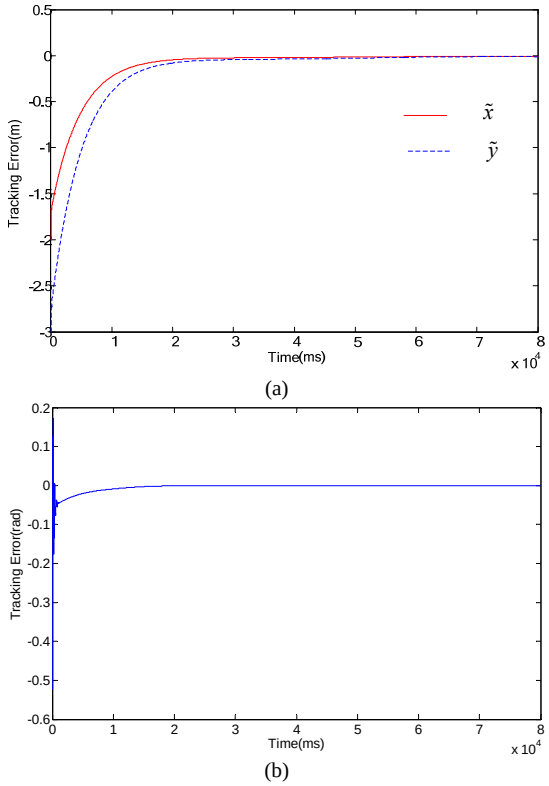


Fig. 7. Errors of the simulated regulation: (a) x-axis and y-axis position errors; (b) orientation errors.

The first simulation is conducted to show the stabilization performance of the proposed STS method with the optimal controller parameters searched via the DNA computing method. Figure 6 depicts the simulation result of the proposed controller for stabilization steering the robot from the initial pose $(0\text{m}, 0\text{m}, \pi/6 \text{ rad})$ to the destination pose $(2\text{m}, 3\text{m}, \pi/3 \text{ rad})$. Figure 7 respectively presents the simulation results of both two-axial position and orientation errors. The results in Fig. 7 clearly indicate that both position and orientation errors go to zero, thus show the effectiveness of the proposed controller.

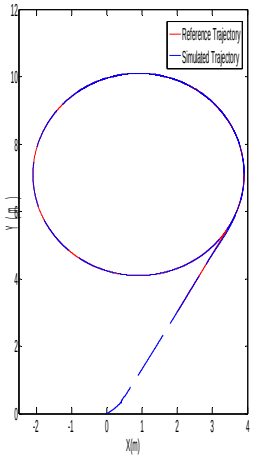


Fig. 8. Simulation result of the proposed controller for the line-circular trajectory tracking.

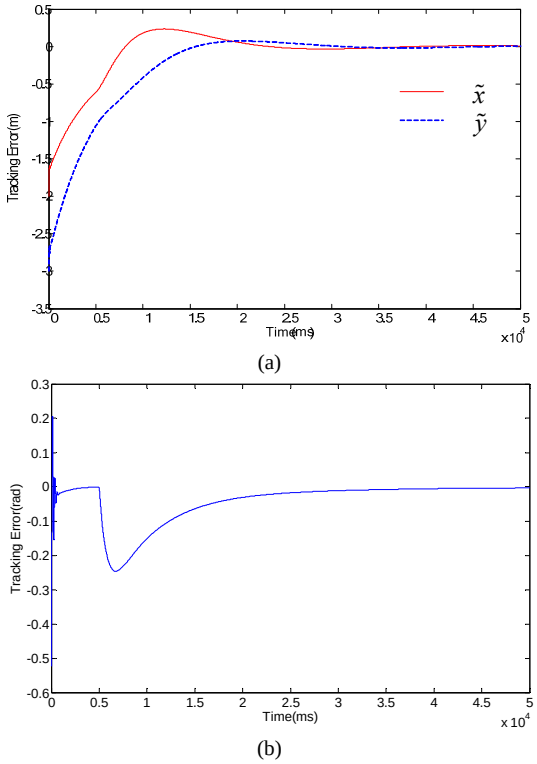


Fig. 9. Simulation results of the proposed controller for the line-circular trajectory tracking: (a) x-axis and y-axis position errors; (b) orientation errors.

The second simulation is performed to illustrate the trajectory tracking performance of the proposed controller. Figures. 8 and 9 respectively depict the simulation results of the proposed control method and both position and orientation errors of the line-circular trajectory tracking. The results in Figs. 6 and 8 clearly indicate that the nonholonomic mobile robot with the proposed DNA-based STS controller can simultaneously achieve both regulation and desired trajectory tracking, thereby illustrating the feasibility and effectiveness of the proposed STS method.

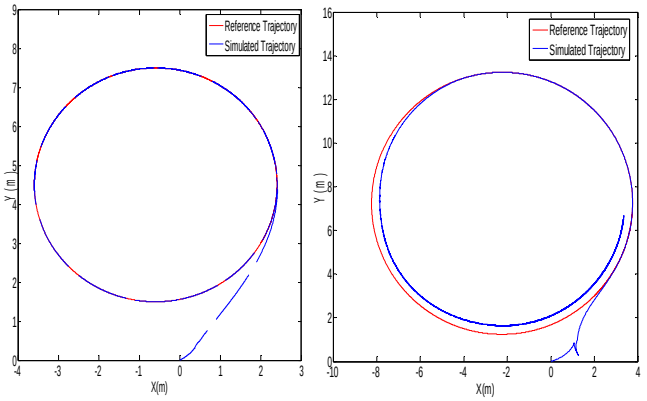


Fig. 10. Performance comparison of the simulated circular trajectory tracking for (a) the proposed controller with the proposed new STS kinematics approach; (b) the proposed controller with Dixon's STS kinematics approach.

The third simulation is executed to compare the proposed control method with the method proposed by Dixon et al. [12]. The performance comparison is done by setting the controller parameter k in (5) as unity, namely that the proposed STS kinematic controller is reduced to the approach presented by Dixon et al. [12]. Figure 10 respectively shows the simulated results of circular following by employing both control methods. The results in Figs. 10(a) and (b) clearly reveal that the parameter k indeed significantly affects the tracking performance, and the DNA-based tuning of these parameters can give better trajectory tracking performance.

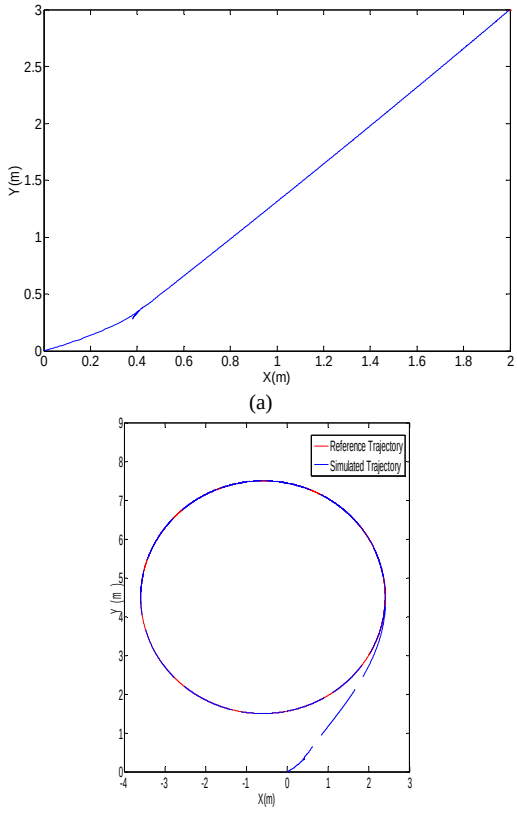


Fig. 11. Simulation results of the proposed controller for (a) regulation and (b) the line-circular trajectory tracking with M=120Kg and I=3.377Kg-m^3.

The fourth simulation is employed to show the robustness of the proposed control method in presence of mass and inertia variations. In the simulation, the platform mass and inertia are respectively changed from 165Kgw to 120Kgw and from $4.643\,\text{Kg-m}^2$ to $3.377\,\text{Kg-m}^2$, but other parameters remain unchanged. Figure 11 depicts the simulation results of the regulation and the line-circular trajectory tracking using the proposed adaptive dynamic controller. Through these simulation results, the proposed adaptive dynamic STS control method is proven capable of giving robust tracking performance against mass and inertia variations.

B. Experimental Results and Discussion

The aim of the following experiments is to examine the effectiveness and performance of the proposed embedded adaptive motion control method by constructing an experimental mobile robot incorporating with the nonholonomic mobile robot. All the hardware and software design of the nonholonomic mobile robot are integrated into an Altera FPGA chip.

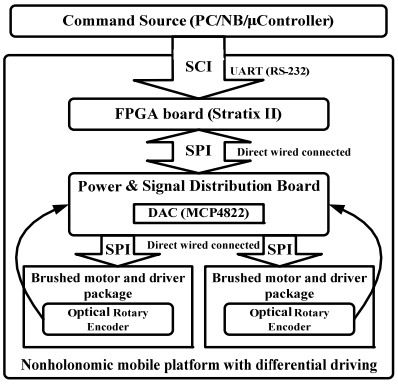


Fig. 12 Block diagram of the experimental nonholonomic mobile robot

1) Experimental Nonholonomic Mobile Robot

This subsection is aimed to describe the system structure and core components of the experimental nonholonomic mobile robot as Fig. 12 shows. In Fig. 12, the command source, such as PC or Notebook (NB), will give the motion commands to the controller (FPGA board). By using the Recommended-Standard-232 (RS-232) interface, one of the Universal-Asynchronous-Receiver/Transmitter (UART) interface, the mobile platform with differential-driving can receive the motion control commands and execute them. When the controller receives the commands, the control system will calculate and generate the digital speed control signals. Through the power and signal distribution board and the on-board Digital-to-Analog-Converter (DAC) chip MCP4822, the output commands are converted into analog voltage signals for driving two dc brushless motors mounted on the two wheels via Serial-Peripheral-Interface (SPI). After the control signal conversion, the motor drive will activate the motor and drive the pulley to generate the QEP feedback signals for dead-reckoning. The purpose of the dead-reckoning of the robot is, given a correct initial pose, to continuously keep tracking of its correct poses with respect to the reference frame.

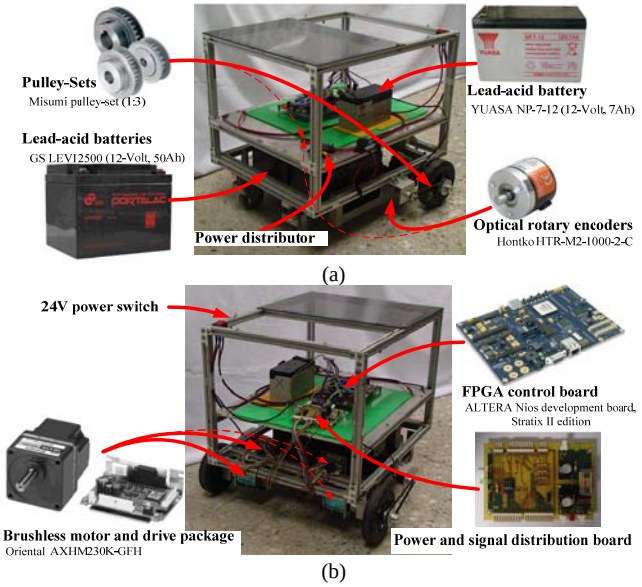


Fig. 13 Physical structure of the platform: (a) front-view (b) back-view

Figure 13 shows the physical structure of the experimental FPGA-based nonholonomic mobile robot with differential-driving, which is equipped with a pair of DC 24-Volt brushless motor and gearbox kit with their own drives, three power supply batteries including a pair of 12-Volt, 50Ah lead-acid batteries (GS LEV12500) for motor, and one 12-Volt, 7Ah lead-acid battery (YUASA NP-7-12) for FPGA board, a pair of external optical encoder with 1000 pulse/round for accuracy, a self-construction power and signal distribution board, and an Altera Stratix II FPGA board using Nios II development board as main controller hardware.

The proposed unified motion control law was implemented using C/C++ code and standard programming techniques in the Altera Nios II embedded processor. The FPGA chip integrated the embedded processor, RTOS, lwIP, and VHDL-based IP circuits to perform the adaptive motion control law of the mobile robot. In the experiments, the two encoders were employed to measure the angular velocities of the two DC brushless motors in order to achieve the dead-reckoning of the robot.

2) Circular Trajectory Tracking

The following experiment is to study circular trajectory tracking performance of the proposed adaptive motion controllers. The platform got started at $(x_0,\ y_0,\ \theta_0) = (0m,\ 0m,\ 0\ rad)$ and then was steered to move along the circular trajectory path described by $(x_r,\ y_r,\ \theta_r) = (v\cos\omega t,\ v\sin\omega t,\ \omega t)$ with $v = 0.3\ m/s$ and $\omega = \pi/12\ rad/s$. Figure 14 shows the sequential still images of the proposed controller for steering the robot from the initial pose to the destination pose. Figure 15 shows the trajectory comparison between experimental data and simulation result of circular trajectory tracking under the same conditions.

Fig. 14. Sequential still images of circular trajectory tracking.

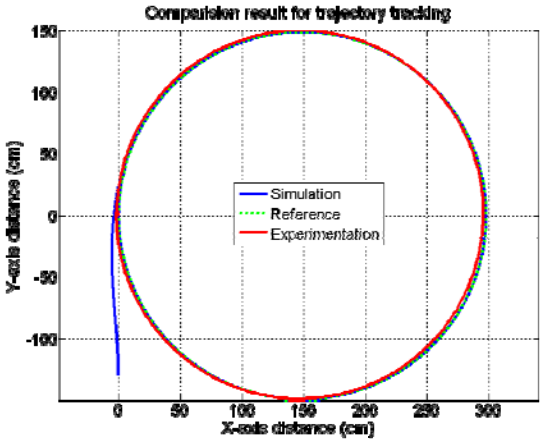


Fig. 15. Performance comparison of the experimental and simulation results for circular trajectory tracking.

Figure 15 shows that the experimental and simulation trajectories are very similar, and the trajectory is close to the reference result. This experimental result reveals that the designed controller using SoPC technology is effective, that is, the implemented controller is capable of giving satisfactory experimental results.

VI. Conclusions

This paper has presented an adaptive dynamic motion controller using DNA computing method to address the STS problem of an uncertain nonholonomic mobile robot. The proposed controller has been constructed first by establishing a kinematic STS controller and then by synthesizing an adaptive dynamic controller via backstepping approach. The kinematic controller has been proven globally exponentially stable by Lyapunov stability theory, and its three optimal parameters have been searched by the DNA computing method. Once the kinematic controller with the optimal control parameters has been constructed, the proposed dynamic controller is then constructed by Lyapunov stability theory and linearized parameterization of the dynamic robot model. Furthermore, the proposed dynamic controller has been shown to be superior to the adaptive dynamic control method with Dixon's STS

kinematic controller. The adaptive dynamic controller, which has been implemented into an FPGA chip with the SoPC technology, has been shown effective and useful by conducting several simulations/experiments on stabilization, trajectory tracking, and robustness via experimentation. An interesting research topic for future work would be to extend the proposed method to investigate adaptive dynamic STS problem for uncertain nonholonomic robots whose centers-of-gravity do not coincide with their geometry centers.

REFERENCES

[1] C. Y. Chen, T. H. S. Li, and Y. C. Yeh, " EP-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots," *Inform. Sci.*, vol. 179, no. 1-2, pp.180-195, 2009.

[2] Q. Yuan, F. Qian, and W. Du, "A hybrid genetic algorithm with the Baldwin effect," *Inform. Sci.*, vol. 180, no. 5, pp. 40-652, 2010.

[3] L. M. Adeleman, "Molecular computing of solutions to combinational problems," *Sci.*, vol. 266, no.5187, pp.1021-1024, 1994.

[4] M. H. Garzo and R. J. Deaton, "Biomolecular computing and programming," *IEEE Trans. Evol. Comput.*, vol.3, no.3, pp. 236-250, 1999.

[5] C. L. Lin, H. Y. Jan, and T. H. Huang, "Self-Organizing PID control design based on DNA computing method," In *Proc. IEEE Conf. Control Appl.*, 2004, pp. 568-573.

[6] X. Liu and Y. Li, "Efficient DNA algorithms for chromatic number of graph problems," In *Proc. IEEE Int. Conf. Automat. Logistics*, 2007, pp. 450-454.

[7] N. Y. Shi and C. P. Chu, "A molecular solution to the hitting-set problem in DNA-based supercomputing," *Inform. Sci.*, vol.180, no.6, pp.1010-1019, 2010.

[8] Y. Zhu, Y. Ding, W. Li, and L.A. Zadeh, "DNA algorithm of image recognition and its application," In *Proc. IEEE Int. Conf. Inform. Reuse and Integration*, 2006, pp. 375-379.

[9] Y. Ding and L. Ren, "DNA genetic algorithm for design of the generalized membership-type Takagi-Sugeno fuzzy control system," In *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2000, pp. 3862-3867.

[10] H. C. Huang, C. C. Tsai, and T. S. Wang, "Kinematics motion planning of an omnidirectional mobile manipulator using DNA algorithm," In *Proc. IEEE Int. Symp. Ind. Electron.*, 2007, pp. 2706-2711.

[11] C. C. Tsai, H. C. Huang, S. C. Lin, "FPGA-Based Parallel DNA Algorithm for Optimal Configurations of an Omnidirectional Mobile Service Robot Performing Fire Extinguishment," *IEEE Trans. Ind. Electron.*, vol. 58, no.3, pp.1016-1026, 2011.

[12] W. E. Dixon, D. M. Dawson, E. Zergeroglu, and A. Behal, *Nonlinear control of wheeled mobile robots*, Springer, 2001.

[13] R. W. Brockett, *Asymptotic stability and feedback stabilization*, in: R.W. Brockett et al. (Ed.), *Differential Geometric Control Theory*, Birkhauser, Boston, 1983.

[14] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive tracking control of a nonholonomic mobile Robot," *IEEE Trans. Robot. Automat.*, vol. 16, no.5, pp.609-615, 2000.

[15] Z. P. Jiang and H. Nijmeijer, "Tracking control of mobile robots: A case study in backstepping," *Automatica,* vol. 33, no.7, pp.1393-1399, 1997.

[16] Y. J. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control scheme for an autonomous mobile robot," In *Proc. IEEE Int. Conf. Robot. Autom.*, 1990, pp. 384-389.

[17] W. Oelen and J. Amerongen Van, "Robust tracking control of two-degree-freedom mobile robots," *Control Engineering Practice*, vol.2, no.2, pp.333-340, 1994.

[18] P. Soueres, J. P. Laumond, "Shortest path synthesis for a car-like robot," *IEEE Trans. Automat. Contr.,* vol.41, no.5, pp.672-688, 1996.

[19] A. M. Bloch and S. Drakunov, "Stabilization of a nonholonomic system via sliding modes," In *Proc. IEEE Conf. Dec. Control*, 1994, pp. 2961-2963.

[20] T. H. S. Li and Y. C. Huang, "MIMO adaptive fuzzy terminal sliding-mode controller for robotic manipulators," *Inform. Sci.*, vol. 180, no. 23, pp. 4641-4660, 2010.

[21] R. K. Barai and K. Nonami, "Optimal two-degree-of-freedom fuzzy control for locomotion control of a hydraulically actuated hexapod robot," *Inform. Sci.*, vol. 177, no. 8, pp.1892-1915, 2007.

[22] S. Bentalba, A. E. Hajjaji, and A. Rachis, "Fuzzy control of a mobile robot: A new approach," In *Proc. IEEE Conf. Dec. Control*, 1997, pp. 69-72.

[23] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Trans. Neural Networks*, vol.9, no.4, pp.589-600, 1998.

[24] C. C. Tsai, H. C. Huang, and S. C. Lin, "Adaptive neural network control of a self-balancing two-wheeled scooter," *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1420-1428, 2010.

[25] W. E. Dixon, D. M. Dawson, F. Zhang, and E. Zergeroglu, "Global exponential tracking control of a mobile robot system via a PE condition," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, no.2, pp.129-142, 2000.

[26] K. D. Do, Z.-P. Jiang, and J. Pan, "A global output-feedback controller for simultaneous tracking and stabilization of unicycle-type mobile robots," IEEE *Trans. Robot. Automat.*, vol. 20, no. 3, pp. 589-594, 2004.

[27] P. Morin and C. Samson, "Control of nonholonomic mobile robots based on the transverse function approach," *IEEE Trans. Robotics*, vol. 25, no.5, pp.1058-1073, 2009.

[28] I. Campo, J. Echanobe, G. Bosque, and J. M. Tarela, "Efficient hardware/software implementation of an adaptive neuro-fuzzy system," *IEEE Trans. Fuzzy Syst.*, vol.16, no. 3, pp. 761-778, 2008.

[29] Y. S. Kung and M. H. Tsai, "FPGA-based speed control IC for PMSM drive with adaptive fuzzy control," *IEEE Trans. Power Electron*., vol. 22, no. 6, pp.2476-2486, 2007.

[30] S. S. Solano, A. J. Cabrera, I. Baturone, F. J. Moreno-Velo, and M. Brox, "FPGA implementation of embedded fuzzy controllers for robotic applications," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp.1937-1945, 2007.

[31] S. J. Huang and S. S, Wu, "Vision-based robotic motion control for non-autonomous environment," *J. Intell. Robot. Syst.*, vol. 54, no.5, pp. 733-754, 2009.

[32] S. H. Han, M.H. Lee, and R. R. Mohler, "Real-time implementation of a robust adaptive controller for a robotic manipulator based on digital signal processors," *IEEE Trans. Syst. Man Cybern., Syst. Hum.*, vol.29, no. 2, pp.194-204, 1999.

[33] T.H. Li, S. J. Chang, and Y. X. Chen, "Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot," *IEEE Trans. Ind. Electron.*, vol.50, no. 5, pp.867-880, 2003.

[34] K. M. Deliparaschos, F. I. Nenedakis, and S. G. Tzafestas, "Design and implementation of a fast digital fuzzy logic controller using FPGA technology," *J. Intell. Robot. Syst.*, vol. 45, no. 1, pp. 77-96, 2006.

[35] C. F. Juang and J. S. Chen, "Water bath temperature control by a recurrent fuzzy controller and its FPGA implementation," *IEEE Trans. Ind. Electron.*, vol. 53, no.3, pp.941-949, 2006.

[36] Y. S. Kung and G. S. Shu, "Design and implementation of a control IC for vertical articulated robot arm using SOPC technology," In *Proc. IEEE Int. Conf. Mechatronics*, 2005, pp. 532-536.

Chien-Cheng Yu received the B.S. and M.S. degrees from Feng Chia University and National Taiwan Normal University, respectively. Now, he works as an Associate Professor with the Department of Electronic Engineering at Hsiuping University of Science and Technology, Taichung City, Taiwan. He is currently working toward the Ph.D. degree at National Chung Hsing University, Taichung City, Taiwan. His current research interests include memory architecture and circuit design, nonlinear control, and intelligent control.

Shih-Min Hsieh received the B.S. degree in electrical engineering from Feng Chia University, and the M.S. and Ph.D. degrees from the Department of Electrical Engineering, National Chung Hsing University, Taichung City, Taiwan. Now, he works as an Assistant Professor with the Department of Electronic Engineering at National Chin-Yi University of Technology, Taichung City, Taiwan.

His current research interests include robotics, nonlinear control, and intelligent control.

Feng-Chun Tai received the B.S., M.S. and Ph.D. degrees in Department of Electrical Engineering from National Chung Hsing University, Taichung, Taiwan, ROC. in 2007, 2010 and 2018, respectively. His current research interests include mobile robots, intelligent control, navigation system and their applications to industrial processes and machines.

Three-Dimensional Global Trajectory Planning Using RGA for Contour Flight of Autonomous Helicopters

Zen-Chung Wang and Ching-Fu Hsu

***Abstract*—This paper presents a global trajectory planning method using real-coded genetic algorithm (RGA) and elastic band technique for contour flight of autonomous helicopters. The RGA is used to find global, optimal navigation points or via-points for the terrain with static or dynamic threats, while the elastic band technique is employed to obtain smooth and deformed local paths when the static or dynamic obstacles are encountered. The integrated path planner by combining the RGA and elastic band technique is proposed to search for the near-optimum, smooth and feasible flight routes over a wide variety of terrain. By using an existing trajectory tracking controller augmented with fuzzy basis function networks (FBFN), the effectiveness and merit of the proposed three-dimensional global trajectory planning algorithm are well exemplified by conducting several simulations on an autonomous small-scale helicopter.**

Index Terms—Autonomous helicopter, elastic band, real-coded genetic algorithm (RGA), global trajectory planning, obstacle avoidance.

I. Introduction

CURRENTLY, the unmanned or autonomous helicopters have been widely used in the civilian and military applications due to their abilities to hover and low-flying. Such helicopters can also be used in harsh landing environments and implementation of emergency in inaccessible locations in order to avoid pilots casualties. In particular, unmanned helicopter used on the battlefield are shown capable of providing real-time surveillance information and performing high risk reconnaissance and jamming support [1]. Moreover, these autonomous helicopters are demanded to guide themselves through cluttered and complicated environments and arrive safely at a desired far location. To handle with these tough flights, the integrated and hierarchical approach to vehicle instrumentation, computing, modeling and control would provide a possible solution [2-5]. Fig. 1 depicts a three-layered control system which consists of a global path planner, a local trajectory re-planner, and a low-level trajectory tracking controller for an autonomous helicopter system. Upon reception of start and destination points from a supervisory controller, the path or trajectory planner generates an optimal and collision-free route for autonomous helicopters. The global path planning problem refers to the finding of connected way-points with given terrain or map and known obstacles, while the local one can be attained by taking into account helicopter dynamics and unexpected and/or moving obstacles.

Zen-Chung Wang is with the National Chung-Shan Institute of Science and Technology, and Ching-Fu Hsu is with the Department of Electrical Engineering, National Chung Hsing University, Taichung, 40227, Taiwan. (Corresponding author Zen-Chung Wang, email: zenchung1125@gmail.com)

The authors gratefully acknowledge financial support from the National Chung-Shan Institute of Science and Technology, Taiwan, R.O.C.

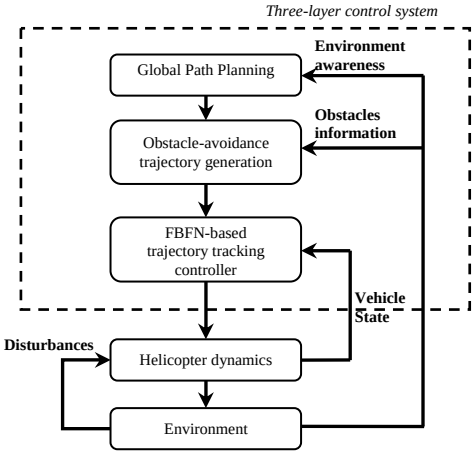


Fig. 1. Illustration of a three-layered hierarchical control system for SSUHs architecture.

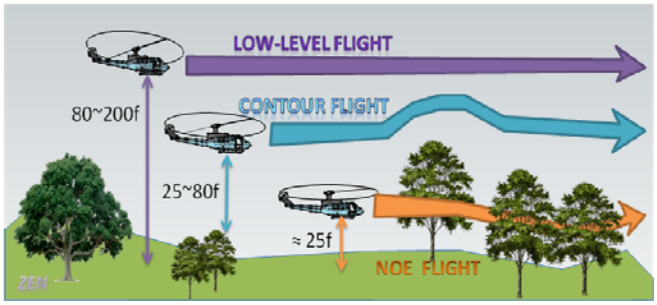


Fig. 2. Three flight modes of autonomous tactical helicopters over flight environments.

The obstacle avoidance is the most important issue for autonomous helicopters since they usually have to fly close to ground or in low altitude in order to enhance terrain masking effect. The flight height is changeable according to the perceiving threats [1]. There are three modes of terrain flight recognized in the U.S. Army doctrine. Fig. 2 illustrates the low level, contour, and nap-of-the-earth (NOE) [5-8].

Genetic algorithm (GA) [9] is one of many popular methods, including Ant Colony Optimization (ACO)[10], Particle Swarm Optimization (PSO) [11] and A-Star algorithm[5], in path planning for unmanned vehicles [12-13] or robot navigation [14-15]. Many GAs have been developed from mimicking the natural biological evolution principle, and considered as global search and optimization methods. Real-coded genetic algorithms (RGA) are different from binary genetic algorithms (BGA) due to no coding and decoding computational requirements for all chromosomes. Since RGA and BGA are regarded as adaptive, heuristic and biological search methods, they have similar four basic operators: selection, reproduction, crossover and mutation. The key idea

behind RGA is to construct the fitness function to evaluate all chromosomes in a population, and to choose the best chromosomes with the best fitness in order to reproduce new chromosomes via the selection operator. Generally speaking, the chromosomes with bigger fitness values have higher probabilities to be selected in the next generation. In addition, crossover and mutation are often utilized to create new chromosomes or off-springs. Compared with the BGA, the RGA has been shown to offer higher precision with more consistent results across replications [16-17]. Moreover, RGA together with PSO was used to find the optimal control law for addressing the multi-UAV formation reconfiguration problem [18]. The BGA was used by Lee [19] to find a set of best waypoints for autonomous helicopter, and the author conjectured that the RGA would give more precise solutions. On the other hand, the elastic band technique based on the potential field approach has been shown to provide a simple and ease computing method to plan the best trajectory to reach a goal location while unknown obstacles are avoiding. Furthermore, the bubble concept offers an efficient method of implementing the elastic band as a collision-free region [3]. The elastic-band-based path is absolutely safe and feasible only if it remains inside the coverage of bubbles. The elastic band concept was originally introduced by Quinlan and Khatib [20]-[21] for robotic path planning and recently applied to the automotive assistant systems including vehicle following [22], lane-keeping, lane-changing [23], and collision avoidance [24]. The elastic band technique has been applied to find an admissible real-time trajectory for an autonomous helicopter maneuvering in complex 3D environments with static and dynamic obstacles [2]. Lee [19] applied a BGA method along with the elastic band approach to search for a set of optimal global waypoints and obtain a smooth trajectory for autonomous helicopters. However, the resultant trajectory has not been shown plausible or feasible by using a low-level trajectory tracking controller yet.

The objective of this paper is to propose a global trajectory planning method using real-coded genetic algorithm (RGA) and elastic band technique for contour flight of autonomous helicopters. By comparing to the existing methods [25-28], the proposed method is particularly useful for contour flight of autonomous helicopters. Moreover, the proposed method may provide useful references for professionals working in this area.

The rest of this paper is organized as follows. Section II delineates the real-coded genetic algorithm (RGA), and Section III states the elastic band technique and its algorithm. In Section IV, the integrated global trajectory planner is proposed by fusing RGA and elastic band technique for contour flight of autonomous helicopter. Section V conducts one numerical simulation to demonstrate the effectiveness and merits of the proposed integrated global trajectory planning algorithm, where all the planned trajectories are tested by an existing FBFN-based trajectory tracking controller. Section VI concludes the paper.

II. Global Way-Point Planning Based On Real-Coded Genetic Algorithm

In this section, an RGA method is adopted to implement the autonomous helicopter for global path planning. All components of a real-coded genetic algorithm are described as follows.

A. Real-Coded Chromosome

The real-value type representation of chromosome is used in this paper. For the genetic algorithm, a chromosome corresponds to a flight route which consists of a sequence of nodes from the start to the goal waypoints. Each node of the same chromosome is specified by the coordinates (x_i , y_i , z_i) . The start node (x_o , y_o , z_o) and end node (x_n , y_n , z_n) are specified as the same for all the chromosomes. The length of a chromosome, i.e., the number of nodes, is determined in the beginning and kept constant during the evolution process. For example, one of chromosome can be expressed as $[w_0\ w_1\ ...\ w_n]$, where each node of $w_i = (x_i\ y_i\ z_i)$ is the waypoint coordinate, the first node (w_o) and the last one (w_n) are specified as the same with other chromosomes.

B. Evolutionary Operators

The operators of crossover and mutation are used to evolve their offsprings. The single-point crossover is applied to recombine two parent chromosomes into two offspring ones. Two parent routes are divided randomly into two parts respectively and recombined with exchanged parts from each other. This evolution process is expressed as follows;

$$\begin{aligned}&[w_0\ w_{p1}\ ...\ w_{pm}\ ...\ w_n]\otimes[w_0\ w_{q1}\ ...\ w_{qm}\ ...\ w_n]\\&=[w_0\ w_{p1}\ ...\ w_{p(m-1)}\ \mid\ w_{qm}\ ...\ w_{q(n-1)}\ w_n]\\&+[w_0\ w_{q1}\ ...\ w_{q(m-1)}\ \mid\ w_{pm}\ ...\ w_{p(n-1)}\ w_n]\end{aligned}\qquad(1)$$

where the symbol ($\otimes$) means the crossover operation, the single-point is randomly selected as m^{th} node. After the crossover operation, two offspring routes are produced to be similar to, but not the same with, their parent routes. The perturbation mutation is used to impose a random change to the coordinates of the intermediate nodes. The change of coordinates should be within the local feasible area. During this evolution process, the new coordinates of the intermediate nodes can be expressed as follows.

$$w_i = w_{i-1} + (r_1 D_x\ r_2 D_y\ r_3 D_z) \qquad (2)$$

where $i = 1,\ ...,\ (n-1)$, r_j, $j = 1,...,3$ are random real numbers between 0 and 1. D_x, D_y and D_z are reasonable distances along the three axes respectively.

C. Evolution Fitness Function

The fitness function of flight route is defined as follows;

$$V = \sum_{i=1}^{n} (w_1 L_i^{\ 2} + w_2 H_i^{\ 2} + w_3 T_i + w_4 A_i^{turn}) \qquad (3)$$

where, w_i, $i = 1,...,4$, are weighting gains for each score parameter of fitness function. Those weighting gains (w_i) can set by positive real values, the larger one stands for much importance for routing planning. L_i is the length of the i^{th} route segment, H_i is the average altitude of the i^{th} route segment, A_i^{turn} is the helicopter turning angles of elevation and azimuth at the i^{th} way-point, they are limited by its flight maximum turning angles of vehicle capability and T_i represents the danger degree exposed to the known static threats. It is expressed as follows;

$$T_i = \sum_{j=1}^{N_{threat}} \frac{k_j}{S_j^{\,4}} \tag{4}$$

where k_j represents the intensity of the j^{th} threat, S_j is the slant range to the j^{th} threat, and N_{threat} is the number of known static threats.

D. Global Way-Point Planning Algorithm

Given the start and goal waypoints, the evolutionary path planning algorithm is described in the following steps.

Step 1: Total l initial populations of chromosomes are randomly generated as the parent generation.

Step 2: Total m populations of chromosomes are generated by the crossover operation.

Step 3: Total l populations of chromosomes are duplicated from the parent generation.

Step 4: All chromosomes including the parent and the offspring populations are evaluated with the fitness function (3). Select the best l chromosomes as the parent generation.

Step 5: Go back to Step 2 if it does not reach the specified evolution generations.

III. Obstacle-Avoidance Trajectory Generation Using Elastic Band Technique

Considering the real-time applications of the autonomous helicopter, the proposed local trajectory generation adopts the elastic band technique with the bubble concept. Before the detailed algorithm is presented, some given conditions are described as follows. The start and goal positions are two navigation waypoints given by the global trajectory planning algorithm. The minimum and maximum radii of bubbles are limited by the priori vehicle dimension and performance of unmanned helicopters, respectively. The unmanned helicopter vehicle is assumed to be capable of detecting the obstacles in the long distance. The distance and direction of obstacles can be measured by the unmanned helicopter on the fly for every 1 second.

The proposed trajectory generation method can be divided into four phases: initial path build-up, elastic band deformation, bubble reorganization and trajectory transformation. First, the initial path between two long-range navigation n waypoints given by the global trajectory planning algorithm is determined by connecting them as the shortest path. Second, this path is constantly modified due to both static and moving obstacles by means of a dynamic path adaptation. Third, the path completeness is verified through the bubble reorganization process. Finally, the collision-free path is smoothed and associated with time to be a feasible trajectory. The detailed procedures and methods are explained in the following paragraphs.

A. Initial Path Build-up

The initial path is the shortest path between two given navigation waypoints by connecting them directly. The round bubbles of the same size are inserted and distributed evenly between the specified start and goal navigation waypoints. The center points of bubbles are viewed as the bubble nodes and labeled with ascending numbers from 0 (start) to $N_{bubbles}$ (goal). The total number of bubble nodes is calculated by $N_{bubbles}$ = ceil(D_{min} / R_{max}^{bub}) , where D_{min} is the shortest distance from the start position to the goal position and R_{max}^{bub} is the maximum bubble radius. The bubble size gives an indication of how far the helicopter is safe from collisions. The reasonable bubble size, R^{bub} , is determined by the following equation,

$$2 \times D^{hel} = R_{min}^{bub} \le \; R^{bub} \; \le R_{max}^{bub} = V_{max}^{hel} \times \Delta T \tag{5}$$

where D^{hel} is the diameter of the main rotor, R_{min}^{bub} is the minimum bubble radius, ΔT is the time interval of local path planning and V_{max}^{hel} is the maximum speed of the unmanned helicopter. Therefore, the initial path is determined by connecting all the bubble nodes without considering any obstacles.

B. Elastic Band Deformation

The initial path is continuously modified in real-time by using the latest measurement of the environmental static and dynamic obstacles by means of a dynamic path deformation mechanism. Two virtual forces are introduced to describe the interaction between bubbles or with external obstacles. Each node (b_i) is attracted by two internal forces from its preceding node (b_{i-1}) and following node (b_{i+1}) respectively. The internal force for a bubble b_i is computed using the following equation

$$\vec{F}_{\text{int}}^{b_i} = k_{\text{int}} \left[\frac{\vec{b}_{i+1} - \vec{b}_i}{\left\| \vec{b}_{i+1} - \vec{b}_i \right\|} \left(\left\| \vec{b}_{i+1} - \vec{b}_i \right\| - R_{min}^{bub} \right) + \frac{\vec{b}_{i-1} - \vec{b}_i}{\left\| \vec{b}_{i-1} - \vec{b}_i \right\|} \left(\left\| \vec{b}_{i-1} - \vec{b}_i \right\| - R_{min}^{bub} \right) \right] \tag{6}$$

where $\|\bullet\|$ denotes the Euclidean norm, k_{int} means the contraction gain. Normally the directions of both internal contraction forces are departing from this node (b_i) to its neighbor bubbles respectively. However, the attractive internal forces will alter to be repulsive forces with the opposite action directions if the node comes too close its neighbor bubbles.

The external forces are repulsive forces exerted by the obstacles of the environment to deform the bubble band

adequately and then keep the path collision-free. Each node (b_i) is repelled by all the nearby obstacles only if they are close enough. Therefore, the directions of all the external forces are aiming at this node from its nearby obstacles. The single repulsive force from the position vector of obstacle (O_j) is calculated as follows,

$$\vec{f}_{ext}^{\,j}(b_i) = k_{ext} e^{-(D_{aff})} \left(\frac{\vec{b}_i - \vec{O}_j}{\left\| \vec{b}_i - \vec{O}_j \right\|} \right) \tag{7}$$

where k_{ext} means the repulsive gain. The fading function depends on the affected distance D_{aff} which is calculated via

$$D_{aff} = \begin{cases} \left\| \vec{b}_i - \vec{O}_j \right\| - D_{safe} & \text{, if } \left\| \vec{b}_i - \vec{O}_j \right\| > D_{safe} \\ 0 & \text{, if } \left\| \vec{b}_i - \vec{O}_j \right\| \le D_{safe} \end{cases} \tag{8}$$

and the safe distance is defined by

$$D_{safe} = R_{\min}^{bub} + V_{bub}^{obs} \times T_s \tag{9}$$

where V_{bub}^{obs} is the relative velocity from the obstacle to this bubble. Finally, the resultant external force acting on this bubble node is calculated by

$$\vec{F}_{ext}^{\,b_i} = \sum_{j=1}^{N} \vec{f}_{ext}^{\,j}(b_i) \tag{10}$$

where *N* is the total number of nearby obstacles. Once both internal forces and external forces have been computed for the node b_i , the net applied force is summarized by

$$\vec{F}_{net}^{\,b_i} = \alpha \cdot \vec{F}_{\text{int}}^{\,b_i} + \beta \cdot \vec{F}_{ext}^{\,b_i} \tag{11}$$

where α and β are respectively the weighting factors for the internal forces and external forces. The new elastic band configuration for each bubble node is

$$\vec{P}_{new}^{\,b_i} = \vec{P}_{old}^{\,b_i} + \gamma \cdot \vec{F}_{net}^{\,b_i} \tag{12}$$

where γ is the step size for updating the bubble band deformation. The path deformation process continues until the deformation amount is less than the specified tolerance. Accompanied with the path deformation, the bubble size (radius) is also updated according to how close the bubble is to the nearest obstacle.

C. Bubble Reorganization

To maintain the elastic band as a continuous and feasible path, the constraint that each bubble overlaps with its two neighbors may require new bubbles to be inserted as the elastic band deforms. In addition, it is desirable to remove redundant bubbles to improve its efficiency. Therefore, two following properties of the elastic band must be checked properly. There is bubble redundancy when a small bubble's coverage is totally within its neighboring big bubble's coverage. In this case, it is desirable to remove redundant bubbles from elastic band to improve its efficiency. If the criterion for bubble redundancy defined by (13) is true, the k^{th} bubble is removed from the elastic band.

$$\left| R_k - R_{k-1} \right| \ge \left\| \vec{b}_k - \vec{b}_{k-1} \right\| \text{ or } R_{k+1} + R_{k-1} > \left\| \vec{b}_k - \vec{b}_{k-1} \right\| + \left\| \vec{b}_{k+1} - \vec{b}_k \right\| \tag{13}$$

where $|\bullet|$ means the absolute value and $\|\bullet\|$ is the 2-norm to measure the distance. On the other hand, there is bubble insufficiency when two neighboring bubbles do not overlap each other due to the bubble moving too far during the path deformation. If the criterion for bubble insufficiency defined in (14) is true, then an extra bubble is inserted at the middle location between b_{k-1} and b_k bubbles.

$$R_k + R_{k-1} - d_{ol} < \left\| \vec{b}_k - \vec{b}_{k-1} \right\| \tag{14}$$

where d_{ol} is the desired overlap distance between two neighboring bubbles.

D. Trajectory Transformation

Finally, this feasible path generated from the elastic-band planner is smoothed by using the cubic B-spline technique. The nodes of the elastic band will be interpolated by a cubic spline curve function (cscvn) using the Curve Fitting Toolbox of Matlab, where Γ_{EB} = cscvn (nodes) is a parametric cubic spline curve through given points (nodes). The vehicle's velocity is controlled by passing through those nodes at specified time interval (T_c) as follows,

$$\vec{V}_i = (\vec{b}_{i+1} - \vec{b}_i) / T_c \tag{15}$$

The closer the bubble nodes allocation, the slower the helicopter flight speeds. A trajectory is essentially a geometric path parameterized by time. We desire to generate a real-time trajectory at 1 Hertz which is only one order of magnitude slower than the control system. For a small-scale helicopter system, the motion controller is usually operating at 50 Hertz, that is, the time interval for trajectory generation is chosen as one second and total 50 points are retrieved evenly from the path Γ_{EB} between two bubble nodes.

E. Algorithm for an Elastic Band Trajectory Generation

On basis of the aforementioned descriptions, a real-time algorithm for elastic band trajectory generation is proposed in the following steps. This 16-step procedure is executed iteratively from the current position to the goal waypoint.

1) Construct the initial path Γ_{ini} consisting of $N_{bubbles}$ bubbles allocated evenly between the start waypoint and the goal waypoint.
2) Perform the elastic band deformation process from the current bubble node through all the uncompleted bubble nodes.
3) Compute the internal forces acting on the i^{th} bubble node due to its neighbor nodes of index *i*-1 and *i*+1 respectively. Refer to (6).
4) Calculate the external forces acting on the i^{th} bubble

node due to all the static and dynamic obstacles. Refer to (7) - (10).

5) Execute the path deformation for the i^{th} bubble node applying the sum of the virtual forces given in Steps 3 and 4. Refer to (11) - (12).
6) Repeat Steps 3 to 5 until the amount of the path deformation is less than the tolerance.
7) Decide the bubble radius on the smallest clearance distance to all the obstacles.
8) Shift the bubble index to the next and repeat Steps 3 to 7 until the end of the elastic band, $i = N_{bubbles}$.
9) Do the bubble reorganization from the current bubble node through all the uncompleted bubble nodes.
10) Check if the i^{th} bubble is redundant and deletes it if (13) is true.
11) Check if the i^{th} bubble has a broken connection with its previous one and inserts an extra bubble at the middle of two bubbles if (14) is true.
12) Do the elastic band deformation if an extra bubble is generated. Refer to Steps 3 to 6.
13) Shift the bubble index to the next and repeat Steps 10 to 12 until the end of the elastic band $= N_{bubbles} - 1$.
14) Generate the real-time collision-free trajectory from the current position to the goal.
15) Smooth the deformed elastic band to get the path Γ_{EB} by using a cubic spline function.
16) Generate a real-time trajectory by correlating the path Γ_{EB} with time based on (15).

Repeat Steps 2 to 16 until the goal is achieved ($i = N_{bubbles}$).

IV. Integrated Global Trajectory Planning Algorithm

Given the start and goal waypoints and three-dimensional terrain, the integrated global trajectory planning algorithm is described by the following steps.

1) Total l initial populations of chromosomes are randomly generated as the parent generation.
2) Total m populations of chromosomes are generated by the crossover operation.
3) Total l populations of chromosomes are duplicated from the parent generation.
4) All chromosomes including the parent and the offspring populations are evaluated with the fitness function (3). Select the best l chromosomes as the parent generation.
5) Go back to Step 2 if it does not reach the specified evolution generations.
6) After the above evolution process, the best route is chosen to perform the following elastic band deformation process, Steps 7-12.
7) Each node of the chosen route is assigned with a bubble of a variable radius size which is determined by the nearest distance with its neighboring terrain.
8) Compute the internal forces acting on the i^{th} bubble node due to its neighbor nodes of index $i-1$ and $i+1$ respectively. Refer to (5).

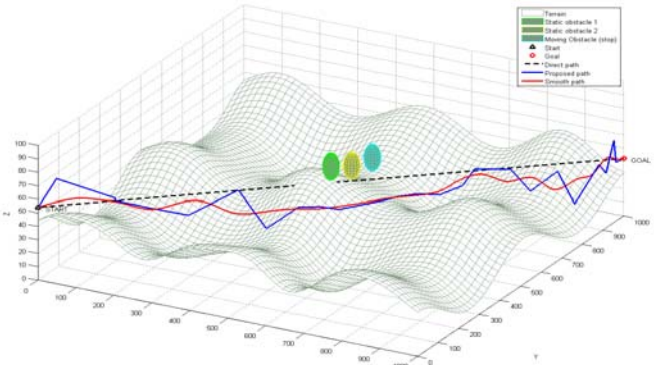


Fig. 3. Planned global trajectories: the naïve straight line by connecting the starting point to the final goal (dot black line); the planned 3D path by the RGA only (blue line); the smooth 3D global trajectory via the proposed method (red line).

9) Calculate the external forces acting on the i^{th} bubble node due to all the static and dynamic obstacles. Refer to (6) - (9).
10) Execute the path deformation for the i^{th} bubble node applying the sum of the virtual forces given in Steps 8 and 9. Refer to (10) - (11).
11) Repeat Steps 8-10 until the amount of the path deformation is less than the tolerance.
12) Decide the bubble radius on the smallest clearance distance to all the obstacles.

V. Simulation Results And Discussion

In this section, one numerical simulation is conducted to investigate and examine the effectiveness and performance of the proposed integrated global trajectory planning algorithm for contour flight of autonomous helicopters. The low-level flight trajectory tracking control is done by a flight controller which adopts an existing intelligent trajectory tracking method augmented with FBFN approximation to the coupling forces between the helicopter actuators [25]. The missions of the autonomous helicopter are to follow the desired trajectories, given by the proposed integrated global trajectory planning algorithm, so as to fly through the 3D complex environment and arrive at the specified destination.

Given start and goal waypoints, the proposed algorithm is offline used to find automatically a feasible trajectory over a rough 3D terrain of 1000 m square space with scattered static and dynamic ground threat targets. The simulated rough terrain with mountains and valleys is represented by a meshed 3D surface which is produced by the following mathematical function.

$$z = a + \sin(y+b) + c \times \sin(x) + d \times \cos(y) + e \times \cos\left(d \times \sqrt{x^2+y^2}\right) + f \times \sin\left(f \times \sqrt{x^2+y^2}\right) \quad (16)$$

where a, b, c, d, e *and* f are constants. The start and the goal locations are specified as [0, 0, 53.91] and [1000, 1000, 42.07] (unit: m), where both waypoints have clear 10 meters of

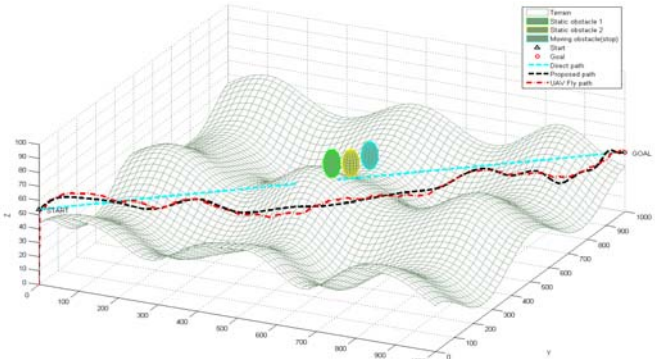


Fig. 4. The refined 3D global path by the proposed integrated global trajectory planning algorithm.

height above the terrain. Three simulations use the identical parameters of the used RGA algorithm as follows; the total initial populations of chromosomes are l=20, which are randomly generated from the parent generation; the total populations of chromosomes are $m = 20$, which are generated by the crossover process among the best 5 parent chromosomes; the total populations of chromosomes are $l = 20$, duplicating from the parent generation; after the evaluation of fitness values, the best chromosomes, $l = 20$, are selected as in the parent generation for evolution of the next generation. Moreover, the three simulations are terminated after 150 generations.

This used scenario considers three static ground threats scattered in the terrain environment. As shown in Fig.3, three static threats are respectively marked by blue, green and yellow colors, mesh balls that are put nearby the moving forward path of the helicopter by directly connecting the starting position to the destination. The coordinates of the three colorful static threats are set by (500m, 700m, 35.79m) for the blue threat, (500m, 600m, 60.23m) for the yellow one, and (500m, 500m, 70.09m) for the green one, respectively. With both given starting and goal waypoints, the proposed integrated global trajectory planning algorithm is applied to find an optimal and safe flight trajectory over the simulated terrain. During the planning process, the RGA-based path planning algorithm is initially employed to search for a set of the best waypoints in the sense of minimum fitness; in particularly, this searching process has a rapid convergence speed. Moreover, at Step 6 in Section 3, the best set of waypoints after processing total 150 populations of chromosomes is selected and shown as the blue solid line in Fig.3. Worthy of mention is that the planned path by connecting these best waypoints is not smooth and it needs further processing by the elastic band deformation algorithm.

After performing the elastic band deformation process from Step 7 to Step 11, the resultant trajectory becomes collision-free and safe over the simulated terrain, shown as the red solid line in Fig.3. It is obvious that this smooth trajectory in red solid line is smoother than the path in the blue solid line. Fig.4 shows the simulated flight trajectory of the autonomous helicopter using the FBFN-based trajectory tracking controller [25]. Through Figs.3-4, the results indicate that the flight position deviations of the autonomous helicopter don't exceed 5 meters in three axes such that the simulated flight trajectory lies within the desired safe working space, showing that the proposed integrated global trajectory planning algorithm is effective.

VI. CONCLUSIONS

This paper has presented an offline global trajectory planning method by combining the real-coded genetic algorithm and elastic band technique, in order to find a collision-free and safe globally flight trajectory for contour flight of autonomous helicopters. The proposed integrated global trajectory planning algorithm has been synthesized as a 12-step design procedure by taking the advantages of the real-coded genetic algorithm and elastic band technique for the contour flight. The effectiveness and merit of the proposed method have been well exemplified by performing three simulations on a class of autonomous helicopters with three scenarios of three static or moving threats. Through three simulations, the planned trajectories have been successfully tested by the used helicopter with an existing FBFN-based trajectory tracking controller in [25]. An interesting topic for future research would be to apply the proposed method to address the trajectory planning for formation of a class of multiple helicopters.

REFERENCES

[1] J. Heikell, *Electronic warfare self-protection of battlefield helicopters: a holistic View*, Doctoral Dissertation, Helsinki University of Technology. Espoo Finland, 2005. (http://lib.tkk.fi/Diss/2005/ isbn9512275465/ isbn9512275465.pdf)

[2] C.T. Lee and C.C.Tsai, "3D collision-free trajectory generation using elastic band technique for an autonomous helicopter," in *Proc. of 14th FIRA RoboWorld Congress*. Kaohsiung Taiwan, pp. 34–41, 2011.

[3] C.T. Lee, *Trajectory planning and adaptive trajectory tracking control for a small scale autonomous helicopter*, Doctoral Dissertation, National Chung-Hsing University, 2010.

[4] H. Yang and Y. Zhao, "Trajectory planning for autonomous aerospace vehicles amid known obstacles and conflicts," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 6, pp. 997-1008, 2004.

[5] S.S. Jan and Y.H. Lin, "Integrated flight path planning system and flight control system for unmanned helicopters," *Sensors*, vol. 11, no. 8, pp. 7502-29, 2011.

[6] V.H.L. Cheng, "Concept development of automatic guidance for rotorcraft obstacle avoidance," *IEEE Trans. on Robotics and Automation*, vol. 6, no. 2, pp.252-257, 1990.

[7] B. Bhanu, S. Das, B. Roberts, and D. Duncan, "A system for obstacle detection during rotorcraft low altitude flight," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 32, no. 3, pp.875-97, 1996.

[8] 1-212th Aviation Regiment, *Tactical flight mission planning and map preparation guide*, U.S. Army, 2009.

[9] Y. Wang, "Path planning and obstacle avoidance of unmanned aerial vehicle ased on improved genetic algorithms," in *Proc. of the 33rd Chinese Control Conference*, Nanjing China, pp. 8612-6, 2014.

[10] H. Duan, Y. Yu, X. Zhang, and S. Shao, "Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm," *Simulation Modelling Practice and Theory*, vol.18, no.8, pp. 1104-1115, 2010.

[11] Y. Fu, M. Ding, and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV," *IEEE Transactions on Systems and Humans*, vol. 42, no. 2, pp.511-526, 2011.

[12] D.G. Macharet, A.A. Neto, and M.F.M. Campos, "Feasible UAV Path planning using genetic algorithms and B´ezier curves," *Advances in Artificial Intelligence–SBIA*, vol. 6404, pp. 223-232, 2010.

[13] Y.V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerospace Science and Technology*, vol. 16, no.1, pp.47-55, 2012.

[14] A.T. Ismail, S. Alaa, and A.W. Mohammed, "A mobile robot path planning using genetic algorithm in static Environment," *Journal of Computer Science*, vol. 4, no.4, pp.341-348, 2008.

[15] T. Arora, Y. Gigras, and V. Arora, "Robotic path planning using genetic

algorithm in dynamic environment," International Journal of Computer Applications, vol. 89, no. 11, pp.8-12, 2014.
[16] Z. Michalewicz, *Genetic algorithms + data structure = evolution programs. 3rd ed*, Berlin Germany, Springer-Verlag, 1996.
[17] C. Zheng, L. Li, F. Xu, and F. Sun, "Evolutionary route planner for unmanned air vehicles," *IEEE Trans. on Robotics*, vol. 21, no. 4, pp.609-620, 2005.
[18] H. Duan, Q. Luo, G. Ma, and Y. Shi, "Hybrid particle swarm optimization and genetic algorithm for multi-UAV formation reconfiguration," *IEEE Computational Intelligence Magazine*, vol. 8, no.3, pp. 16-27, 2013.
[19] C.T. Lee, "3D Global path planner based on genetic algorithm and elastic band technique for autonomous helicopters," in *Proc. of 2011 International Conference on Service and Interactive Robots*, National Chung Hsing University. Taichung Taiwan. pp. 527-531, 2011.
[20] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," *IEEE Computer Society Press*, vol 2, pp. 802-807, 1993.
[21] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vo. 5, no. 1, pp. p.90-98, 1995.
[22] J. Hilgert, K. Hirsch, T. Bertram and M. Hiller, "Emergency path planning for autonomous vehicles using elastic band theory," in *Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Kobe Japan, vol. 2, pp.1390-1395, 2003.
[23] S.K. Gehring and F.J. Stein, "Elastic bands to enhance vehicle following," in *Proc of 2001 IEEE International Conference on Intelligent Transportation Systems*. Oakland CA USA, 2001, pp. 597-602.
[24] T. Hesse and T. Sattel, "An approach to integrate vehicle dynamics in motion planning for advanced driver assistant systems," in *Proc. of the 2007 IEEE Intelligent Vehicle Symposium*, Istanbul Turkey, pp. 912-917, 2007.
[25] C.C. Tsai, Z.C. Wang, C.T. Lee and Y.Y. Li, "Intelligent adaptive trajectory tracking control for an autonomous small-scale helicopter using fuzzy basis function networks," *Asian Journal of Control*. Article first published online: 13 APR 2014, DOI: 10.1002/asjc.881.
[26] C.T. Lee and C.C. Tsai, "Improved nonlinear trajectory tracking using RBFNN for a robotic helicopter," *Int. J of Robust and Nonlinear Control*, vol. 20 no.10, pp.1079-1096, 2010.
[27] C.T. Lee and C.C. Tsai, "Nonlinear adaptive aggressive control using recurrent neural networks for a small scale helicopter," *Mechatronics*, vol. 20, no. 4, pp.474-84, 2010.
[28] Z.C. Wang, *Intelligent distributed formation control of small-scale unmanned helicopters*, Ph. D dissertation, Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, R.O.C., Jan. 2016.

Zen-Chung Wang received his M.S. degree from Chung Cheng Institute of Technology, Taoyuan, Taiwan. Currently, he is a Ph.D. in the Department of Electrical Engineering, National Chung-Hsing University, Taichung, Taiwan. Mr. Wang also works as an engineer for Chung-Shan Institute of Science & Technology, Taiwan. His research interest is in electronic control system design, especially in training simulators.

Ching-Fu Hsu received the B.S. degree in Electronic Engineering from Chienkuo Technology University in 2001 and M.S. degree in Department of Computer Science and Information Engineering from Chaoyang University of Technology in 2003, respectively. From 2003 to 2015, he served a project leader in Taiwan Wintek Corp. Since 2015, he has been with Taiwan ASUS Computer Corp. Currently, he is working toward his Ph.D degree at Department of Electrical Engineering, National Chung Hsing University. His current interests include image processing and driving technology and their applications to touch/display modules.

A Laboratory Course on Mobile Robotics Education

Chien-Cheng Yu, Ching-Fu Hsu, and Feng-Chun Tai

Abstract — **Mobile robotics education often needs to develop fundamental but interesting hands-on exercises that keep students interested in the mobile robotics theory presented in lectures. In this paper, we put emphasis on the construction of a laboratory course with a pedagogical differential-driving mobile robot for the teaching of mobile robotics concepts in senior undergraduate or first-year graduate engineering environments. Such a pedagogical mobile robot can be easily and inexpensively constructed using low-tech commercial components and feedback control approaches. An education process together with a pedagogical method is presented to show how the experimental mobile robot can be incorporated into the laboratory course. To increase students' hands-on experience and keep them interested in learning mobile robotics, particular efforts are paid to investigate how the enrolled students responded to this pedagogical tool. This education method along with the mobile robot is shown significantly effective in helping students understand mobile robotics theory and practices, and also resulting in more motivated and active learning.**

***Index Terms*—Differential-driving, locomotion and motion control, mapping and localization, mobile robot, path planning and navigation, sensing and perception.**

I. Introduction

MOBILE robotics has been extensively instructed in many senior undergraduate or first-year graduate engineering curriculums, such as electrical engineering, mechanical engineering, mechatronic engineering and so on [1]-[3]. Mobile robotics education often faces the challenging problem of how to give enrolled students interesting and pragmatic pedagogical tools by providing hands-on experiments and design problems that complement the theory presented in lectures of mobile robotics. In particular, students can gain significant benefits if such pedagogical tools are simultaneously applied to main core technologies in mobile robotics. Furthermore, the students are significantly impressed on mobile robotics concepts presented in class if they can easily implement and evaluate the performance of their developed methods on these teaching tools.

Generally speaking, the general control structure of mobile robots includes four main technologies: locomotion and motion control, sensing and perception, localization and mapping, cognition and path planning, as proposed in [4]. As our best understanding, most of mobile robotics courses cover the main theories of these four main technologies in lectures, but they lack of laboratory exercises to complement those theories. Hence, this paper aim to develop four interesting and pragmatic laboratory exercises for the course lectures.

To date, many pedagogical tools have been presented to give senior undergraduate and/or graduate students hands-on experience on learning basic concepts of mobile robotics [5]-[6]. Wheeled mobile robots have been viewed as one of the most widely used pedagogical tools in teaching mobile robots among many universities [1]-[3]. From the viewpoint of mobile robot design and implementation, simple differential-driving wheeled mobile robots are much cheaper to construct by using inexpensive commercial components, and easier to control. With the differential-driving platforms, a completely mobile robotic system can be efficiently constructed by a synthesis of desired sensors with appropriate interfacing, mechatronic components, and software of localization, mapping, path planning, obstacle avoidance, navigation, and motion control. This reason motivates us to explore how to develop a differential-driving wheeled mobile robot for teaching mobile robotics.

The rest of this paper is organized as follows. The course overview and syllabus are introduced in Section II. Sections III and IV respectively detail the laboratory setup and experimental exercises setup. Section V conducts four experiments with students to exemplify the merit of the proposed four exercises mentioned in the previous sections. The course evaluation from the enrolled students at the end of this course is briefly presented and discussed in Section VI. Section VII concludes the paper.

II. Course Description

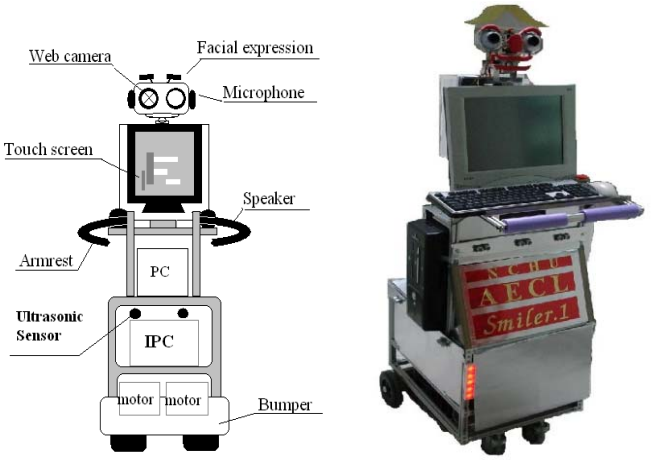
A. Course Overview

The one-semester course, called autonomous mobile robotics, was taught for the first-year graduate students at the National Chung Hsing University, Taichung, Taiwan. In this course, all students should take 3-hour lecture each week. Except the introduction, this course is composed of four parts: sensing and perception, mapping and localization, motion control, and path planning. For each part, one experiment is accompanied by the course lectures, and the enrolled students have to attend a 3-hour laboratory after finishing lectures regarding each part. The laboratory was taught by three teaching assistants under supervision of the teaching professor. Below is the syllabus of this course.

Chien-Cheng Yu, Ching-Fu Hsu and Feng-Chun Tai are with the Department of Electrical Engineering, National Chung Hsing University, Taichung, 40227, Taiwan.

Chien-Cheng Yu is also with the Department of Electronic Engineering, Hsiuping University of Science and Technology, Taichung, Taiwan.
(Corresponding author Chien-Cheng Yu, email: ccyu@hust.edu.tw)
(Email: hchingfu@gmail.com, fctai@nchu.edu.tw)

The authors gratefully acknowledge financial support from the Ministry of Science and Technology, Taiwan, the R.O.C., under contract MOST 106-2221-E-005-003.



(a) (b)

Fig. 1 Experimental mobile robot: (a) system structure; (b) physical robot.

B. Syllabus

This course was offered in an 18-week semester. The class met once a week and each class session persisted three hours. The weekly topics taught in the course are outlined as follows.

- Week 1 (lecture): Introduction; basic concepts, categories and applications of mobile robots are learned.
- Week 2 (lecture): Locomotion; describe how mobile robots needs locomotion mechanisms that enable them to go throughout their environments.
- Week 3-5 (lecture): Sensing and Perception; some fundamental concepts and features of various kinds' sensors are discussed.
- Week 6 (experiment 1): Sensing and Perception; acquire the surrounding environment information by laser scanning sensor, ultrasonic ranging module, infrared switch module and RFID module.
- Week 7-9 (lecture): Mapping and Localization; describe how sensors and effectors uncertainty is responsible for the difficulties of localization, and two extreme approaches that avoiding localization altogether and performing explicit map-based localization to dealing with the challenge of robot localization are mentioned.
- Week 10 (experiment 2): Mapping and Localization: calibrate the RFID system; robot position estimation; robot orientation estimation; real-time pose initialization estimation.
- Week 11-12 (lecture): Mobile Robots Kinematics and Control; introduce kinematic models, controller designs and the related controllers' synthesis to achieve position control (goal-seeking) and obstacle avoidance.
- Week 13 (experiment 3): conduct experiments of position control (goal-seeking) and obstacle avoidance.
- Week 14-16 (lecture): Path Planning and Navigation; discuss path planning and obstacle avoidance, required for mobile robot navigation; demonstrates how combine motion control, sensing and perception, mapping and localization, path planning, obstacle avoidance and navigation into one complete robot system for a real-world application.
- Week 17 (experiment 4): Path Planning and Navigation; conduct experiments of combination of position control and obstacle avoidance.
- Week 18: final examination and final report.

C. Textbook, Course Materials

The textbook "Autonomous Mobile Robots" [6] is used to teach most of the core concepts and techniques of mobile robots.

III. LABORATORY SETUP

This section briefly describes the laboratory setup. In order to let all enrolled students, have hands-on experience on wheeled mobile robots, the experimental robot is a differential-driving wheeled mobility platform equipped with two personal computers, two motor drives, laser scanner, ultrasonic ranging modules, infrared sensors, radio-frequency identification (RFID) module, power supply subsystem, facial expression system, web camera, touch-screen panel, and one wireless internet card. Fig. 1 displays the physical structure of the laboratory setup for teaching mobile robots. The computing unit of the robot includes a small-scale personal computer and an industrial personal computer (IPC), one D/A interfacing card, one digital parallel 8255 input/output interfacing card, and a touch-screen panel. This computing unit is responsible for executing software of the sensing perception, localization, cognition and path planning, motion control.

To calculate the traveling distances and speeds of the experimental mobile robot, two HTR-M2-1000-2-C rotary encoders are mounted on the driving wheels. In regular operations, HTR-M2-1000-2-C will output two square waves with different phases into the QEP module of the counter board for decoding. Thus, the calculated information could be reckoned for achieving accurate motion control. With different angular resolutions and ranging distances, the laser scanner LMS 291-S05 manufactured by SICK Electro-Optics and the ultrasonic ranging modules are utilized to obtain the environmental information around the robot's surrounding. The laser system and the ultrasonic distance measurement are based on the time-of-flight (TOF) measurement principle. The infrared switch module is adopted to provide the nearest safe protection; for example, when the approaching object or visitor is detected, the robot immediately stops. The RFID module is employed to roughly find the position of the robot by installing a reader on the robot and several tags in the robot's working environment.

IV. EXPERIMENTAL EXERCISES SETUP

This section is devoted to introducing experimental exercises for teaching aforementioned four main techniques.

A. Experiment 1: Sensing and Perception

One of the most important tasks of the mobile robot is to

acquire knowledge about its environment. RFID and laser scanners have been considered as two powerful devices for mapping of the environment where the mobile robot works. The following illustrates the calibration method to convert the Received Signal Strength Indication (RSSI) values from active RFID tags into their corresponding distances. For the *n*-th tag, the relation between the RSSI and the distance is expressed by

$$\mathrm{RSSI}_i = \frac{K}{d_i^{\,m}}, \quad i = 1, \cdots, n. \tag{1}$$

where the parameters *K* and *m* are two unknown measurements, $d_1, d_2, \cdots d_n$ denote the distances from the robot to the tags. Given *m* pairs of the RSSI and distance measurement, both parameters *K* and *m* can be found by taking the logarithmic operation for (1) and then using least-squares method. The calibration procedure to find parameters *K* and *m* will be repeated for each tag until all the tags are calibrated.

B. Experiment 2: Global Localization

In this experiment, our global localization system adopts four active tags and one RFID reader mounted on the robot.

1) Robot Position Estimation

From the aforementioned calibration model, the distances between the robot and the tags can be computed. Therefore, the following subsequent equation (2) can be obtained.

$$\mathbf{AX} = \mathbf{B} \tag{2}$$

where

$$\mathbf{A} = \begin{bmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) & 2(z_1 - z_2) \\ \vdots & & \\ 2(x_1 - x_i) & 2(y_1 - y_i) & 2(z_1 - z_i) \\ \vdots & & \\ 2(x_1 - x_m) & 2(y_1 - y_m) & 2(z_1 - z_m) \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} d_2^2 - d_1^2 + (x_1^2 - x_2^2) + (y_1^2 - y_2^2) + (z_1^2 - z_2^2) \\ \vdots \\ d_i^2 - d_1^2 + (x_1^2 - x_i^2) + (y_1^2 - y_i^2) + (z_1^2 - z_i^2) \\ \vdots \\ d_m^2 - d_1^2 + (x_1^2 - x_m^2) + (y_1^2 - y_m^2) + (z_1^2 - z_m^2) \end{bmatrix}$$

The least-squares method is again used to solve the matrix equation.

$$\mathbf{X} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{B} \tag{3}$$

which gives the robot position $(x, y, z)^T$.

2) Robot Orientation Estimation

Since the robot position $(x, y, z)^T$ can be calculated via (3), the robot heading would be found by driving the robot to move in a straight line with the original heading direction, where the straight line is given as follows:

$$y_i = m\mathbf{X}_i + c \tag{4}$$

In (4), the parameter *m* is the slope of the straight line and *c* is the constant offset. Assuming two more robot positions to be obtained from (3), one re-arranges (4) in a matrix-vector form by

$$\begin{bmatrix} 1 & \mathrm{X}_1 \\ 1 & \mathrm{X}_2 \\ \vdots & \\ 1 & \mathrm{X}_n \end{bmatrix} \begin{bmatrix} \mathrm{c} \\ m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \Rightarrow \mathbf{CY} = \mathbf{D}, \ \mathbf{Y} = \begin{bmatrix} \mathrm{c} \\ m \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 1 & \mathrm{X}_1 \\ 1 & \mathrm{X}_2 \\ \vdots & \\ 1 & \mathrm{X}_n \end{bmatrix}, \ \mathbf{D} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{5}$$

Again, the least-squares method is employed to solve the matrix equation (6).

$$\begin{bmatrix} \mathrm{c} \\ m \end{bmatrix} = \mathbf{Y} = (\mathbf{C}^{\mathrm{T}}\mathbf{C})^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{D} \tag{6}$$

From (6), the slope *m* is determined, and the robot orientation θ can be calculated by solving $m = \tan\theta$.

C. Experiment 3: Goal-Seeking and Obstacle Avoidance

In this experiment, the fuzzy control strategy is used to establish two basic behaviors of the mobile robot, including goal-seeking and obstacle avoidance behaviors.

1) Kinematic Model

Under the assumption of pure rolling, the kinematic model of the mobile robot can be given as follows:

$$\dot{q}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} v_1 \cos\theta \\ v_1 \sin\theta \\ v_2 \end{bmatrix}, \ v_1(t) = r w_{ave}, \ v_2 = \frac{w_r - w_l}{b} \tag{7}$$

where $x(t)$ and $y(t)$ are the reference points of the mobile robot with respect to the Cartesian frame, $\theta(t)$ is the orientation of the mobile robot. Furthermore, $v = \begin{bmatrix} v_1 & v_2 \end{bmatrix}^T$, $v_1(t)$ represents the linear speed, $v_2(t)$ denotes the angular velocity, w_r and w_l respectively represent the angular speeds of the left and right wheels, *r* is the radius of the driving wheel, and *b* is the distance between the left and right wheels. Once both linear and angular velocities of the robot have been determined, the angular speeds of the left and right wheels can be solved.

2) Goal-seeking Behavior Design

The robot is now treated as a point within an artificial potential field $U_p(q)$. Under the influence of the field, the robot moves from high potential to low potential, The artificial force field $F(q)$ acting at the point $q = [x, y]^T \in S$, can be defined as follows:

$$F(q) = -\nabla U_p(q) = -\begin{bmatrix} \dfrac{\partial U_p}{\partial x} & \dfrac{\partial U_p}{\partial y} \end{bmatrix}^T \tag{8}$$

where $\nabla U_p(q)$ is the gradient of U_p at the position $q = [x, y]^T$. The potential field function U_p is regarded as the attractive field of the destination and the artificial force field $F(q)$ is

considered as the attractive force from the goal to the robot. The attractive potential function can be defined as a parabolic function:

$$U_p = K_p \rho_p^2(q)/2 \tag{9}$$

where K_p is a positive coefficient and $\rho_p(q)$ represents $\|q - q_{goal}\|$. With the differentiable attractive potential function, the attractive force can be described as follows:

$$F(q) = -\nabla U_p(q) = -K_p \rho_p(q) \nabla \rho_p(q) = -K_p(q - q_p) \tag{10}$$

When the robot arrives in the desire position, (10) will converge to zero. The control law is constructed according to the force components that act on the robot and the robot's speed.

3) Obstacle Avoidance Behavior Design

The obstacles are detected by a laser scanner and nine ultrasonic rangers mounted on the robot. There are three ultrasonic rangers facing forward, three on the left side, and three on the right side. By fusing these sensing data with fuzzy control algorithm, the robot can move around the obstacles without collision. The scan range of the laser scanner is 180 degrees. In order to simplify the derivation of the obstacle behavior, we divide the scanning range into five regions, each of which covers 36 degrees. From right to left, these regions are labeled by L_1, L_2, L_3, L_4 and L_5, in which L_2, L_3, L_4 belong Group I and L_1, L_5 belong Group II. The shortest distances found inside the five regions are denoted by D_{L1}, D_{L2}, D_{L3} D_{L4} and D_{L5}, and are inferred from the perceived distances obtained from the laser scanner. With the use of the fuzzy rules for laser-based obstacle avoidance, the rotation control ω_{obsL} can be obtained. The ultrasonic readings can be divided into three regions, each of which comprises three sectors. These sectors are labeled from right to left as U_1, U_2 and U_3 (Group I), U_4, U_5 and U_6 (Group II), U_7, U_8 and U_9 (Group III). The shortest distances for nine sections are named in turn by D_{U1}, D_{U2}, $\cdots$, D_{U9}, and are derived from the perceived distances obtained from the ultrasonic rangers. With the use of the fuzzy rules for laser-based obstacle avoidance, the rotation control ω_{obsU} can be obtained.

For calculating the ultimate rotation control $\omega_{obstacle}$ of the obstacle avoidance behavior, the rotation control of the obstacle avoidance behavior can be expressed as follows:

$$\omega_{obstacle} = \begin{cases} \omega_{obsL} + \lambda_3 \omega_{obsU}, \text{ when } (\omega_{obsL} \cdot \omega_{obsU}) \le 0 \\ \max(\omega_{obsL}, \omega_{obsU}), \text{ when } \omega_{obsL}, \omega_{obsU} \ge 0 \\ \min(\omega_{obsL}, \omega_{obsU}), \text{ when } \omega_{obsL}, \omega_{obsU} < 0 \end{cases} \tag{11}$$

where λ_3 is the pre-specified weighting.

D. Experiment 4: Path Planning and Navigation

To perform the local navigation, the robot must immediately react to sensory events such as passing through narrow passageway without obstacle collision. For global navigation, the robot needs to implement dynamic path planning to find an optimal path between the goal and the starting pose. In order to solve the optimal path problem, the Dijkstra's algorithm [ ] can be used to obtain the optimum route in the well-defined environment. Dijkstra's algorithm is a greedy one for solving the single-source shortest path problem. According to the working principle of the Dijkstra's algorithm, the vertices (nodes) and the path are the two essential factors. Thus, before using the Dijkstra's algorithm, the environment data has to be processed to obtain a topological map. After obtaining necessary nodes in the environment data, the starting point of the mobile robot and the chosen goal are added into the set of nodes. Then a topological map is thus constructed.

Each node format should contain four elements: (i) node ID; (ii) its geometrical coordinate; (iii) the number of nodes connected to it (the next nodes); (iv) the node ID of the nodes connected to it. Worthy of mention is that once the Dijkstra's algorithm terminates, the found global optimal path is expressed in terms of a sequence of via-nodes.

V. Experimental Results and Discussion

This section present four experimental results for sensing and perception, localization, goal-seeking and obstacle avoidance, and path planning and navigation using the experimental mobile robot.

A. Experiment 1: Sensing and Perception

In this experiment, Visual C++ was applied to program the user operation interface, shown in Fig. 2(a). Furthermore, the system utilize this interface to receive the RSSI value via RS-232 communication port for further analysis. The setting of RS-232 communication port as show in Fig. 2(b). The RFID module via the operation interface receives the tag information while setting the Tag ID in the calibration procedure, and records each RSSI signal corresponding to each meter. Every tag calibration curve is used to covert the RSSI values into their corresponding distances, as shown in Fig. 2(c). With change into the calibration curve interface, the host computer will transform the RSSI values to the corresponding distances. Once the data between the RSSI values and the distance are recorded, the values of *m* and *K* are then calculated using the least-squares method, as shown in Fig. 2(d).

B. Experiment 2: RFID Global Pose Initialization

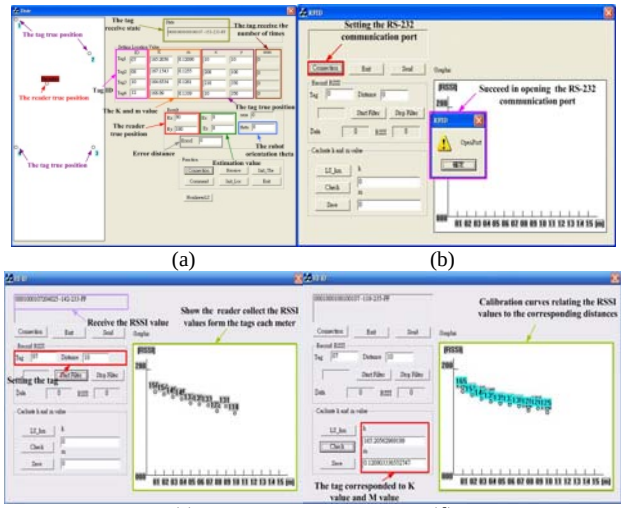
This experiment was performed to investigate the accuracy of the proposed method for RFID static pose estimate of the tour-guide robot with one reader on the head and four tags on the ceiling. The four tags were installed at the position $(x_1, y_1, z_1) = (47,\ 544,\ 300)$, $(x_2, y_2, z_2) = (400,\ 680,\ 299)$,

$(x_3, y_3, z_3) = (473, 313, 299)$ and $(x_4, y_4, z_4) = (595, 2, 300)$ (unit: cm) with respect to the world frame. The true position of the robot in both x and y coordinate frame was given by (117.5cm, 160cm) and the true heading angle θ was 90°. Before experimentation, all the four tags were calibrated using (1). Afterwards, the real-time pose initialization algorithm was applied to calculate the pose estimates of the robot $(128.43\text{ cm}, 171.06\text{ cm}, 74.36^o)$.

In Fig. 3(a), the circle represents the true position, and the cross represents the least-squares estimate. We observe that the proposed RFID global localization method is proven capable of having the position error of less than 20 cm and the heading error of less than 25^o. These experimental results indicate that the proposed pose initialization method can be effectively used to find the correct initial pose of the robot with respect to the world frame.

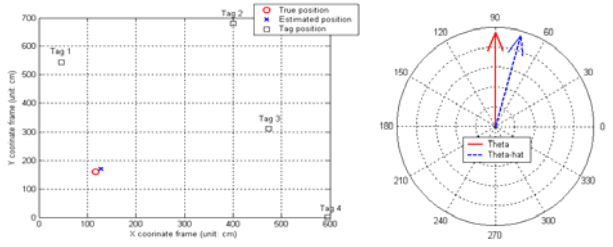
C. Experiment 3: Behavior Control

Fig. 4 depicts continuous moving pictures of the proposed goal-seeking behavior, and Fig. 5 displays the experimental pictures of the proposed obstacle-avoidance behavior. As can be seen in Figs. 4 and 5, both proposed behavior designs were shown effective in achieving required motions. These results also illustrate the applicability of these two proposed behaviors.



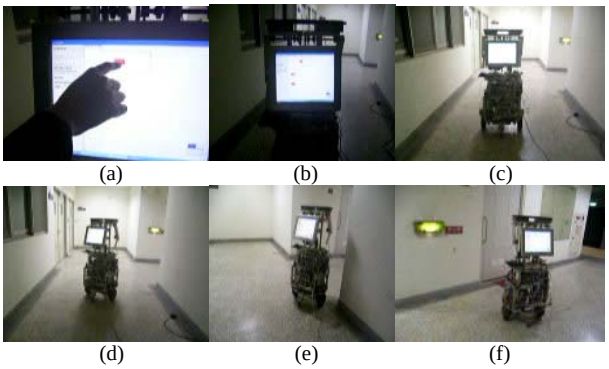
(a) (b) (c) (d)

Fig. 2. The calibration procedure of the relationship between the reader and each tag. (a)User operation interface for RFID system. (b)RS-232 communication port connects to host computer. (c)The relationship between RSSI values and distances. (d)Finding the values of the parameters *K* and *m*.



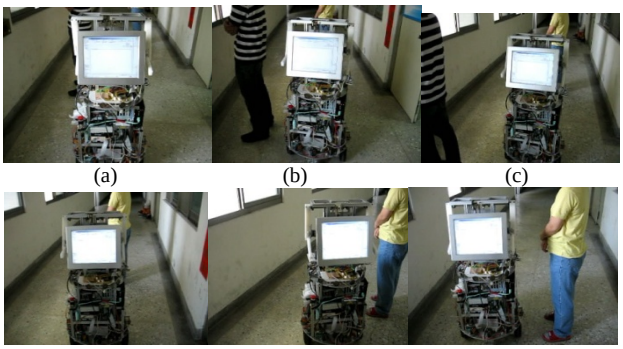
(a) (b)

Fig. 3. (a) Static robot position estimates. (b) Static robot orientation estimates using the RFID pose initialization algorithm.



(a) (b) (c) (d) (e) (f)

Fig. 4. Continuous moving pictures of the mobile robot while demonstrating its goal seeking behavior.



(a) (b) (c) (d) (e) (f)

Fig. 5. Continuous moving pictures demonstrating the performance of the proposed obstacle-avoidance behavior.

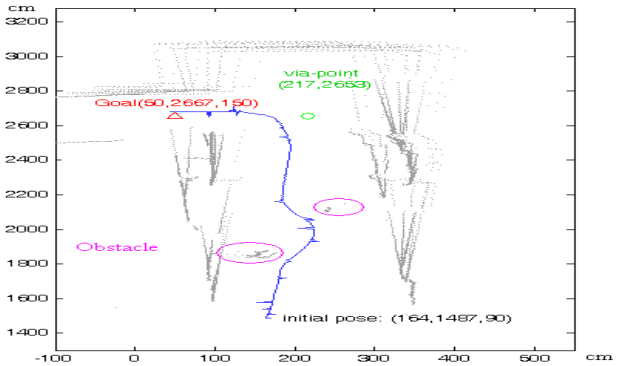


Fig. 6. Experimental results with the initial pose $(164\text{ cm}, 1847\text{ cm}, 90°)$ to the final pose.

TABLE 1. STUDENT EVALUATION RESULTS.
(5 is excellent, 4 is good, 3 is fair, 2 is poor, 1 is very p.oor)

Questions	5	4	3	2	1	Avg
The course organization and planning are adequate.	4	9	2	0	0	4.13
The level of the handouts is adequate.	5	8	2	0	0	4.2
The contents of the handouts are well-developed.	4	8	3	0	0	4.07
This lab related to the lecture courses well.	1	11	3	0	0	3.87
I gained useful and practical skills from the course of mobile robotics.	4	8	3	0	0	4.07
I gained usefully theoretical skills from the course of mobile robotics.	4	8	3	0	0	4.07
Experience gained with MATLAB/Simulink in class work for the course of mobile robotics.	3	7	5	0	0	3.87
Overall, this is a good course.	9	4	2	0	0	4.47

D. Experiment 4: Behaviors Fusion and Path Planning

The cooperation/competition coordination methods are the main technologies for the behaviors fusion of hybrid navigation method. The actual robot angular velocity ω is generated by combining the output ω_{goal} of the position control behavior control and the output $\omega_{obstacle}$ of the obstacle avoidance behavior, with the weighting constant α determined by the obstacle information. The angular output of the cooperation fuzzy weight coordination is simply expressed by

$$\omega = \alpha \cdot \omega_{goal} + (1-\alpha) \cdot \omega_{obstacle} \text{ , for } 0 \le \alpha \le 1 \qquad (12)$$

Fig. 6 shows the experimental results of the proposed path planning approach and behaviors fusion method where the initial pose and the final pose of the robot were set by $(164\text{ cm}, 1847\text{ cm}, 90°)$ and $(50\text{ cm}, 2667\text{ cm}, 150°)$, the via-point generated from the map information is $(217\text{ cm}, 2653\text{ cm})$. In order to make the robot to move smoothly, there was a buffer region with a radius of 150 cm around the via-point, when the robot moved into this region, it started to turn toward the goal direction. The experimental results clearly indicate the effectiveness of the proposed fuzzy behavior fusion strategy together with sensing and localization methods.

VI. Assessment of Student Performance and Student Feedback

This section briefly presents and discusses the course evaluation from the enrolled students at the end of this course, especially for the use of the experimental mobile robots. The course evaluation was based on a set of questionnaire shown in Table 1. Overall, the responses of the enrolled students of the proposed laboratory activities were quite positive.

From this evaluation, it was concluded that after completion of the laboratory experiments by studying the four main techniques, most of the enrolled students considered it as an effective teaching tool to help them understand the basic robotics concepts behind to learn the mobile robotics theory and practices. With the four laboratory lectures, the students had a stronger motivation to construct robotics control system, to design robotic controllers and then to evaluate their performance and effectiveness, conduct the relevant simulations and experiments, and gain usefully theoretical and practical skills.

VII. Conclusions

This paper has presented a laboratory course on mobile robotics education. Via the help of a simple differential-driving wheeled mobile robot, experiments related to the four laboratory lectures can be easily conducted for showing how the proposed methods work well. For educational purpose, the proposed with differential-driving wheeled mobile robot can be highly recommended to be introduced at a senior undergraduate or first-year graduate course in mobile robotics.

The use of the proposed differential-driving wheeled mobile robot as a concrete and interesting example of a wheeled mobile robot not only permits enrolled students to effectively improve their theoretical understanding and mastery of mobile robot technology, but also helps them have hands-on experience of working on a complete analysis and design procedure of a mobile robot system. Hence, most of the students improve their professional confidence as a result of the substantially and meaningfully educational experience.

References

[1] "Robotic courses," The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2011 [Online]. Available: http://www.ri.cmu.edu/ri_static_content.html?menu_id=276.
[2] "Department of Computing, Robotics course," Imperial College London, London, U.K., May 2011 [Online]. Available: http://www.doc.ic.ac.uk/teaching/coursedetails/333.
[3] "Master in robotics, systems and control," ETH Zurich, Zurich, Switzerland, May 2011 [Online]. Available: http://www.masterrobotics.ethz.ch.
[4] R. Siegwart and I. R. Nourbakh, *Introduction to Autonomous Mobile Robots*, the MIT Press, 2004.
[5] M. Pinto, A. P. Moreira, and A. Matos, "Localization of Mobile Robots Using an Extended Kalman Filter in a LEGO NXT," *IEEE Trans. Educ.*, vol. 55, no. 1, pp. 135-144, 2012.
[6] H. Y. Liu, W. J. Wang, and R. J. Wang, "A Course in Simulation and Demonstration of Humanoid Robot Motion Education," *IEEE Trans. Educ.*, vol. 54, no. 2, pp. 255-262, 2011.
[7] C. Buiu, "Hybrid Educational Strategy for a Laboratory Course on Cognitive Robotics," *IEEE Trans. Educ.*, vol. 51, no. 1, pp. 100-107, Feb. 2008.
[8] S. C. Lin and C. C. Tsai, "Development of a Self-balancing Human Transportation Vehicle for the Teaching of Feedback Control," *IEEE Trans. Educ.*, vol. 52, no. 1, pp. 157-168, Feb. 2009.

Chien-Cheng Yu received the B.S. and M.S. degrees from Feng Chia University and National Taiwan Normal University, respectively. Now, he works as an Associate Professor with the Department of Electronic Engineering at Hsiuping University of Science and Technology, Taichung City, Taiwan. He is currently working toward the Ph.D. degree at National Chung Hsing University, Taichung City, Taiwan. His current research interests include memory architecture and circuit design, nonlinear control, and intelligent control.

Ching-Fu Hsu received the B.S. degree in Electronic Engineering from Chienkuo Technology University in 2001 and M.S. degree in Department of Computer Science and Information Engineering from Chaoyang University of Technology in 2003, respectively. From 2003 to 2015, he served a project leader in Taiwan Wintek Corp. Since 2015, he has been with Taiwan ASUS Computer Corp. Currently, he is working toward his Ph.D degree at Department of Electrical Engineering, National Chung Hsing University. His current interests include image processing and driving technology and their applications to touch/display modules.

Feng-Chun Tai received the B.S., M.S. and Ph.D. degrees in Department of Electrical Engineering from National Chung Hsing University, Taichung, Taiwan, ROC. in 2007, 2010 and 2018, respectively. His current research interests include mobile robots, intelligent control, navigation system and their applications to industrial processes and machines.

Design and Implementation of SLAM by Piecewise Linear Feature Based RBPF for Service Robots

Chih-Yin Liu, Chung-Lin Lee, Jie-Jhong Liang, Chih-Yen Chen, and Tzuu-Hseng S. Li*

***Abstract*—This paper proposes a piecewise linear feature based Rao-Blackwellised particle filter (RBPF) simultaneous localization and mapping (SLAM) system with some applications for interacting service robots. The SLAM is a fundamental and important capability for mobile robots and it allows the robots to construct the environment map and to localize themselves. The proposed SLAM system calculates line features form the data extracted by a 2D laser range finder and adopts an RBPF structure to improve the accuracy of the system. For each particle, it estimates a trajectory of the robot and a map by matching the observation model with the built map. A fitness function is designed for evaluating the quality of the particles individually and the particles have good fitness values will be remained and others will be replaced. Besides, we also propose four necessary functions with the SLAM system for our interacting service robot, including path planning, object location establishment, human robot interaction system, and accompany walking strategy. An interacting scenario is setup to examine the efficiency of all functions. In this scenario, the robot acts as a staff in a supermarket and provides two services that are similar to human workers. The experimental results demonstrate the interacting service robot can provide attainable interaction and service.**

***Key words*: accompany walking strategy, human robot interaction, interacting robot, Rao-Blackwellised particle filter, simultaneous localization and mapping**

I. Introduction

Rnowadays have more and more chances to interact with people and provide personal services. How to interact with a person and provide suitable services quickly and suitable is not an easy problem. The robot must be equipped some level of autonomous learning abilities to interact with people and the environment. Mobility is the basis of an interacting robot, because it allows the robot to move in the surroundings. The mobility contains two technologies, simultaneous localization and mapping (SLAM) and path planning. Therefore, we propose a Rao-Blackwellised particle filter (RBPF) based SLAM system. The SLAM system allows the robot to build the environment map and localize itself. Based on the SLAM system, several application functions, including path planning, object location establishment, human robot interaction system, and accompany walking strategy, are incorporated in for providing suitable personal services. Our ideal is that the robot can interact with the environment and the person to find out what it should provide and then generate a suitable moving path and motions to accomplish it.

Therefore, constructing a SLAM system is the first step we should do. Many sensors can be used for SLAM and they can generally be divided into image sensors [1] [2] and ranger sensors [3] [4]. The SLAM used image sensors are usually called vision SLAM, and it can present more details of the environment. However, the amount of information and computation complexity limits the range it can construct. These advantages and disadvantages of image sensors are just contrary to ranger ones. The laser ranger finders have usually high angular resolution, dense scanning, and are suitable used for large environment. Therefore, we adopt a 2D laser range finder as the main instrument for measurement.

For laser based SLAM, there are two popular main structure, namely extended Kalman filter (EKF) [6]-[10] algorithm and RBPF [11]-[13]. Because the EKF methods are easily affected by the nonlinearity and cause inevitable inconsistency in solutions [14], we choose the RBPF framework. The RBPF can perform well no matter the SLAM system is a linear or nonlinear one. Another advantage of RBPF is that the Rao-Blackwellization method can reduce the sample space and improve the computation time. For further reducing the compilation complexity, we adopt an enhanced sequential segmentation algorithm [15] to extract the line features from the point data to build a compact map.

The SLAM system constructs a map and calculates the robot location, but it cannot guide the robot how to move. Therefore, a path planning function has to be integrated in. A* algorithm is one of the best known path planning algorithm and it performs well when the global map is existed [16]. The advantage of A* algorithm is simple for implement, fast, and high modifiability [17]. Therefore, we choose it for path planning algorithm and provide a modification method to smooth the robot path.

When the robot walks with a person or try to interact with a person, the moving path planning is another problem different form the basic path planning problem. The suitable relative position and velocity is the key element in this problem. It can be generated by observing human behavior, the trajectories of

This work was supported by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant MOST 105-2221-E-006 -104 -MY2 and MOST 107-2221-E-992 -301.

The authors are with the aiRobots Laboratory, Department of Electrical Engineering, National Cheng Kung University, Tainan 701, Taiwan, R.O.C. (e-mail: scold@mail.ncku.edu.tw; zllee.jason@gmail.com; km02000147@gmail.com; garycjrock83@gmail.com; thsli@mail.ncku.edu.tw;)

*: Corresponding Author.

two persons walking side by side are analyzed as the reference of the robot position [18] [19], or it can be designed by psychology researches [20]. In another situation, when the robot wants to provide service actively and approaches to a person, it has to predict the behaviors of the person and choose a gently manner to approach to [21] [22].

Besides, the above mentioned functions, some other functions are also developed and integrated in the robot system to allow the robot accomplish interacting tasks. For illustrate and demonstrate the efficiency of the propose methods, we construct an interaction scenario similar a market. We choose market scenario because some researches show a robot provides services in a market is workable and may change the shopping experience. For example, some robots play a role as a receptionist and the survey from the customers show that the robot receptionists do attract the attention of customers [23] [24]. Some robots detected the moving trajectory of a person and approach him/her to provide advertisement [25], and some robots can follow a customer and grasp the merchants from the shelf for him/her [26].

The main contributions of this paper are:

- A SLAM system which can be easily implemented on a mobile robot in an indoor environment with low computation load and high performance.
- A real application scenario which demonstrate the efficiency of the proposed SLAM system and other integrated functions.
- An intelligent and feasible interaction which provides the person who interacts with the robot feels comfortable and convenient.

The remainders of this paper are organized as follows. The concept and details of the proposed piecewise linear feature based RBPF is presented in Section II. Next, Section III describes three main relative functions of interacting with a person in a market scenario. In Section IV, the accuracy of the SLAM system and the application services are examined and the results are illustrated. The conclusion and future work are represented in Section V.

II. Piecewise Linear Feature Based RBPF SLAM

SLAM means that both the trajectory of the robot and the locations of landmarks are needed to be estimated without any priori information. The robot uses a sensor to measure the position of the landmarks, and calculates its position by means of the established map.

In general, the SLAM problem is represented as a probabilistic form as the following equation

$$p(\mathbf{x_k}, \mathbf{M} | \mathbf{Z_{0:k}}, \mathbf{U_{0:k}}, \mathbf{x_0}) \tag{1}$$

where $\mathbf{x_k}$ is the state vector which represents the robot's location and orientation at time k, $\mathbf{M}$ is the set of position of landmarks. $\mathbf{Z_{0:k}}$ is the set of landmark observations from time 0 to k, $\mathbf{U_{0:k}}$ is the history of control inputs, and $\mathbf{x_0}$ is the initial position of the robot. This probability distribution represents joint posterior density of locations of landmark and state vector of the robot's location given the set of observation of landmarks, the history of control inputs, and initial pose of the robot.

After that, we can use a recursive way to solve the SLAM problem. This algorithm contains two steps, a predict step and a correction step. In prediction step, we define a motion model of the robot to predict the next position of the robot, as the following equation

$$\begin{aligned} &p(\mathbf{x_k}, \mathbf{M} | \mathbf{Z_{0:k-1}}, \mathbf{U_{0:k}}, \mathbf{x_0}) = \\ &\int p(\mathbf{x_k} | \mathbf{x_{k-1}}, \mathbf{u_k}) p(\mathbf{x_{k-1}}, \mathbf{M} | \mathbf{Z_{0:k-1}}, \mathbf{U_{0:k-1}}, \mathbf{x_0}) d\mathbf{x_{k-1}} \end{aligned} \tag{2}$$

where $p(\mathbf{x_k} | \mathbf{x_{k-1}}, \mathbf{u_k})$ is the motion model of the robot in which the current position is predicted by the previous position, $\mathbf{x_{k-1}}$, and the motion command, $\mathbf{u_k}$.

The position of the robot, $\mathbf{x_k}$, is a three-dimensional state vector as $\mathbf{x_k} = [x_k \quad y_k \quad \theta_k]^T$. x_k and y_k are coordinates of robot pose in Cartesian coordinate, and θ_k is the orientation. $\mathbf{u_k}$ is the result of a motion command at time $k-1$, as the following equation

$$u_{\mathbf{k}} = \begin{bmatrix} Dm_k \cos(\theta_{k-1} + \theta m_k) \\ Dm_k \sin(\theta_{k-1} + \theta m_k) \\ \theta m_k \end{bmatrix} \tag{3}$$

where Dm_k is the moving distance, θm_k is the delta angle of robot's orientation, and θ_{k-1} is the robot's orientation. Therefore, we can calculate the motion model of the robot by means of summing up $\mathbf{x_{k-1}}$ and $\mathbf{u_k}$ as the following equation

$$p(\mathbf{x_k} | \mathbf{x_{k-1}}, \mathbf{u_k}) = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} Dm_k \cos(\theta_{k-1} + \theta m_k) \\ Dm_k \sin(\theta_{k-1} + \theta m_k) \\ \theta m_k \end{bmatrix} + \begin{bmatrix} \delta m_x \\ \delta m_y \\ \delta m_\theta \end{bmatrix} \tag{4}$$

where δm_x, δm_y, and δm_θ represent noises caused by uneven ground, dynamical friction, and other reasons. These noise are modeled as a Gaussian distribution with zero mean.

In correction step, we use the sensor input to construct an observation model of the landmarks to correct the prediction of robot position as the following equation

$$\begin{aligned} &p(\mathbf{x_k}, \mathbf{M} | \mathbf{Z_{0:k}}, \mathbf{U_{0:k}}, \mathbf{x_0}) = \\ &\frac{p(\mathbf{z_k} | \mathbf{x_k}, \mathbf{M}) p(\mathbf{x_k}, \mathbf{M} | \mathbf{Z_{0:k-1}}, \mathbf{U_{0:k}}, \mathbf{x_0})}{p(\mathbf{z_k} | \mathbf{Z_{0:k-1}}, \mathbf{U_{0:k}})} \end{aligned} \tag{5}$$

where $p(\mathbf{z_k} | \mathbf{x_k}, \mathbf{M})$ is the observation model which is a conditional probability distribution of the observation set of landmarks given the set of position of landmark and the pose of the robot.

In this study, we adopt a laser range finder as the sensor, and extract the 180 degrees ahead of the robot with an angular

resolution of 0.5°. Because each measurement is assumed individual, the observation model can be calculated by the product of the individual measurement likelihood [27], as the following equation

$$p(\mathbf{z_k}\,|\,\mathbf{x_k},\mathbf{M}) = \prod_{r=1}^{R} p(z_k^r\,|\,\mathbf{x_k},\mathbf{M}) \tag{6}$$

where the observation is represented as $\mathbf{z_k} = \{z_k^1, z_k^2, \cdots, z_k^R\}$, and R is the number of the observation data.

In real application, some noises occurred in the sensor measurement [27], so we modeled three kinds of noise in the observation to improve the accuracy of our system. The first noise is caused by the atmospheric effect on signal and this noise restricts the resolution of the sensor. We modeled this noise as a Gaussian distribution, as the following equation

$$p_{SMN}(z_k^r\,|\,\mathbf{x_k},\mathbf{M}) = \begin{cases} N(z_k^r\,|\,z_k^{r*},\sigma_{SMN}) & ,\text{if } 0 < z_k^{r*} < z_{Max} \\ 0 & ,\text{otherwise} \end{cases} \tag{7}$$

where z_k^{r*} is the real distance between the sensor and the object in the r_{th} angle of measuring, σ_{SMN} is the standard deviation of this Gaussian model, and z_{Max} is the maximum distance which the sensor can work effectively in. When the measured distance is within the maximum distance, an adjusted distance is calculated by

$$N(z_k^r\,|\,z_k^{r*},\sigma_{SMN}) = \frac{1}{\sigma_{SMN}\sqrt{2\pi}} e^{-\frac{(z_k^r - z_k^{r*})^2}{2\cdot\sigma_{SMN}^2}} \tag{8}$$

Otherwise, the measured distance will be replaced by zero.

The second noise is caused by the random noise of the laser ranger sensor itself. Hence, we modeled this noise as a uniform distribution as the following equation

$$p_{Rand}(z_k^r\,|\,\mathbf{x_k},\mathbf{M}) = \begin{cases} \dfrac{1}{z_{Max}} & ,\text{if } 0 < z_k^{r*} < z_{Max} \\ 0 & ,\text{otherwise} \end{cases} \tag{9}$$

The third noise is caused by the limitation of the sensor. When the robot meets a narrow gap between obstacles, or a black object, the laser sensor may measure a super big value. Therefore, we detect the measured value and when it is bigger than the maximum distance, we replace the measured distance by the maximal distance, as the following equation

$$p_{Max}(z_k^r\,|\,\mathbf{x_k},\mathbf{M}) = \begin{cases} 1 & ,\text{if } z_k^r = z_{Max} \\ 0 & ,\text{otherwise} \end{cases} \tag{10}$$

After that, we integrate these three noises to model our observation model by the following equation

$$p(z_k^r\,|\,\mathbf{x_k},\mathbf{M}) = \begin{bmatrix} w_{SMN} \\ w_{Max} \\ w_{Rand} \end{bmatrix}^T \begin{bmatrix} p_{SMN} \\ p_{Max} \\ p_{Rand} \end{bmatrix} \tag{11}$$

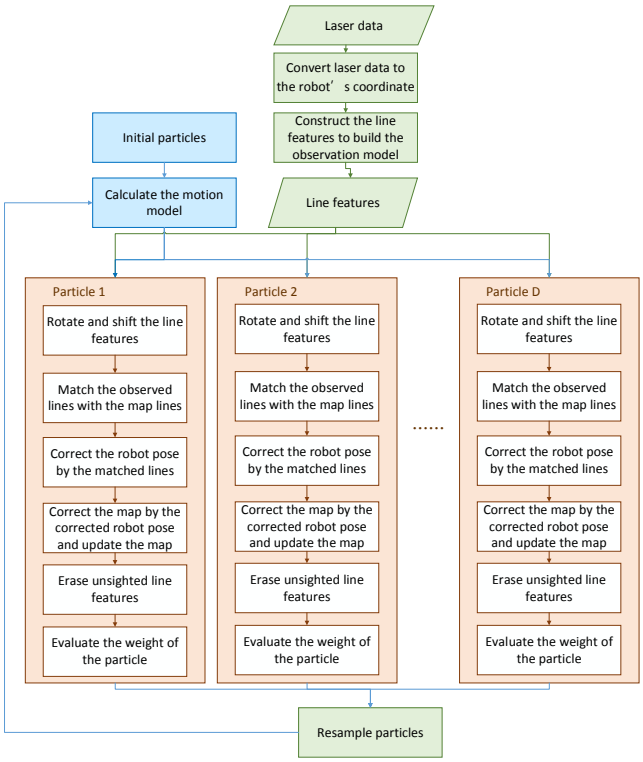


Fig. 1. The overall flow chart of the RBPF SLAM.

where w_{SMN}, w_{Max}, w_{Rand} are the weights of each probability distribution function, and their summation is equal to one.

Therefore, the probabilistic form of SLAM can be represented by multiplying the observation model and the motion model, as the following equation

$$p(\mathbf{X_{1:k}},\mathbf{M}\,|\,\mathbf{Z_{1:k}},\mathbf{u_{1:k}}) = p(\mathbf{M}\,|\,\mathbf{X_{1:k}},\mathbf{Z_{1:k}})\,p(\mathbf{x_k}\,|\,\mathbf{Z_{1:k}},\mathbf{u_{1:k\text{-}1}}) \tag{12}$$

In this paper, we apply the RBPF to our SLAM system as shown in Fig. 1, to improve the system accuracy. We first initial several particles and each particle will estimate a trajectory of the robot and a map by matching the observation model with the built map. A fitness function is designed for evaluate the quality of the particles individually and in its entirety. The particles have good fitness values will be remained and others will be replaced by new particles.

The map in SLAM system can be represented by several kinds of features, such as point features, line features, or image features. Choosing a suitable feature type is the first issue in this problem. For real time implementation on a service robot, we adopt piecewise linear features to construct a map. Line features can decrease the computation load and time of map matching and robot localization.

Before extracting the features, we have to convert raw data of the laser range finder into the robot's coordinate [28] by the following equation

$$p_{\gamma,k}^l = \begin{bmatrix} x_{\gamma,k}^l \\ y_{\gamma,k}^l \end{bmatrix} = \begin{bmatrix} x_k^r \\ y_k^r \end{bmatrix} + \begin{bmatrix} z_{\gamma,k}\cos(\theta_k^r + \theta_{\gamma,k}^l) \\ z_{\gamma,k}\sin(\theta_k^r + \theta_{\gamma,k}^l) \end{bmatrix} \tag{13}$$

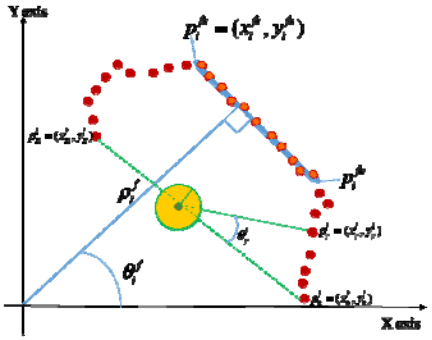


Fig. 2. An example of data conversion.

where $z_{\gamma,k}$ is the measured distance at the γ_{th} angle at time k, and $\theta^l_{\gamma,k}$ is the angle of γ. x^r_k, y^r_k, and θ^r_k represent the robot's location and orientation. Hence, the measured distance can be translate to a position in the robot coordinate as $x^l_{\gamma,k}$ and $y^l_{\gamma,k}$. Fig. 2 illustrates an example in which the yellow circle is the robot and the red dots represent the translated observed positions in the robot coordinate.

After that, we will search consecutive points to construct a line. How many points are needed to be a line is a tricky problem, in this paper, we define five consecutive points is the lowest criterion of a line. Therefore, we calculate the Euclidean distance between two adjacent points. If the distance is lower than a threshold, these two points are treated as consecutive points. The threshold is defined as the following equation

$$th^l = c_1(z_{\gamma,k} + z_{\gamma+1,k})\sin(Ra) + c_2 \tag{14}$$

where Ra is the angular resolution of the laser range finder. c_1 and c_2 are adjusting coefficients. Therefore, this threshold is variant with the measured distances, $z_{\gamma,k}$ and $z_{\gamma+1,k}$. The larger the distance is, the bigger the threshold is. When we have extracted the consecutive points, we then use the line regression skill [28] to calculate the parameters of each line. A line is represented by the following features: the positions of two end points, relation coefficient of linear regression, and center of gravity, slope, and intercept of the line. In this paper, we choose to use polar coordinate to construct a map, because the rotation and shift in polar coordinate is faster and simpler than in Cartesian coordinate.

A line in polar coordinate is represented as follows:

$$x\cos\theta + y\sin\theta = \rho \tag{15}$$

where x and y are the variables in the Cartesian coordinate. θ is the slope, and ρ is the intercept of this line. They are calculated by the following equations

$$\theta = \frac{1}{2}A\tan 2\left(\frac{-2S_{xy}}{S_{yy} - S_{xx}}\right) \tag{16}$$

$$\rho = x^g\cos\theta + y^g\sin\theta \tag{17}$$

where $g = (x^g, y^g)$ is the coordinate of the center of gravity of this line, Sxx and Syy are the variances, and Sxy is the covariance. The relation coefficient is calculated by the following equation

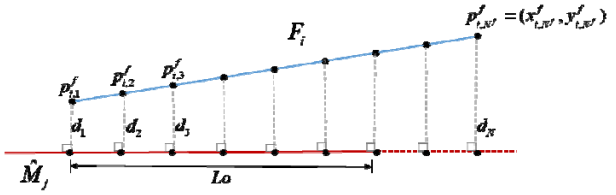


Fig. 3. Illustration of similarity comparison.

$$r = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}} \tag{18}$$

If the absolute value of the relation coefficient is smaller than the threshold, th^r, this line will be abandon. This calculation ensures these points are really on the same line. Besides, we also check the length of the line, and too short lines will be discarded. Before constructing the line into the map, we add another step to check whether the points close to the two end points belong to this line or not. If the nearby point is close enough to the end point and meets the features of the line, this point will be added into the line and the features of the line will be recalculated.

Therefore, we can get all features of an observed line F_i and represent it as

$$F_i = (p_i^{fs}, p_i^{fe}, \theta_i^f, \rho_i^f, g_i^{fg}, r_i^f, N^f) \tag{19}$$

where p_i^{fs} and p_i^{fe} are the two end points, N^f is the number of the points on this line. Fig. 2 also shows some line features.

Therefore, the observed map is a set of observed lines and can be represented as

$$F = \{F_1, F_2, ..., F_I\}, i = 1, ..., I \tag{20}$$

After the SLAM system extracting the line features from the raw data, the next step is to match these lines with the established map lines. At first, the center of gravity, slope, and intercept of the line need to be rotated and shifted by the following equations

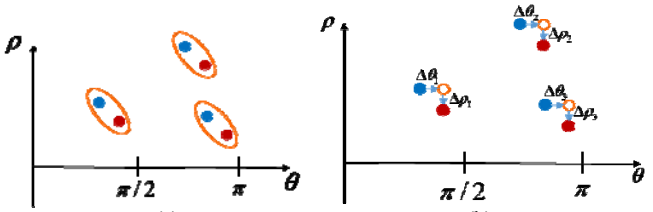
$$g^{fg,R\&S} = \begin{bmatrix} x^{fg,R\&S} \\ y^{fg,R\&S} \end{bmatrix} = \begin{bmatrix} \cos(\theta^{Ro}) & -\sin(\theta^{Ro}) \\ \sin(\theta^{Ro}) & \cos(\theta^{Ro}) \end{bmatrix} \begin{bmatrix} x^{fg} \\ y^{fg} \end{bmatrix} + \begin{bmatrix} x^{Sh} \\ y^{Sh} \end{bmatrix} \tag{21}$$

$$\theta^{f,R\&S} = \theta^f + \theta^{Ro} \tag{22}$$

$$\rho^{f,R\&S} = x^{fg,R\&S}\cos(\theta^{f,R\&S}) + y^{fg,R\&S}\sin(\theta^{f,R\&S}) \tag{23}$$

where θ^{Ro}, x^{Sh}, and y^{Sh} are the rotated angle and the shifted distances of the line which are estimated by the robot motion model. The symbols with a superscript *R&S* are the translated features of the line.

Then, we calculate the similarity of the translated line with the established line by comparing their absolute difference between the slopes, $|\Delta\theta|$, and the overlap length Lo. If both of them are smaller than the predefined thresholds, $th^{\Delta\theta}$ and th^{Lo}, we then calculate the difference between these two lines. For



(a) (b)

Fig. 4. The result of the matching pairs. (a) The matched line pairs. (b) The rotation slope and the intercept difference of the line pairs.

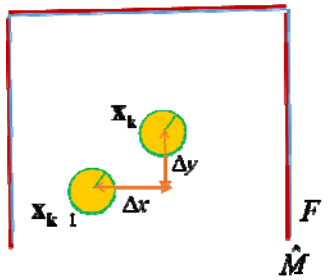


Fig. 5. An illustration of the robot pose displacement.

doing this, we sampling N^f points on the observed line with uniform distance. For each point we calculate the perpendicular distance with the map line and sum them up, as the following equation

$$d_i = \sum_{n=1}^{N^f} |d_n| = \sum_{n=1}^{N^f} \left| x_{i,n}^f \cos(\theta_j^m) + y_{i,n}^f \sin(\theta_j^m) - \rho_j^m \right| \quad (24)$$

where d_n is the perpendicular distance between the n_{th} sampling point, ($x_{i,n}^f$, $y_{i,n}^f$), on the observation line and the established line, $\hat{M}_j$. Fig. 3 is an illustration of the similarity comparison. Similar with observed lines, an established line $\hat{M}_j$ and the established map can be represented as

$$M_j = (p_j^{ms}, p_j^{me}, \theta_j^m, \rho_j^m, g_j^{mg}, r_j^m, N_j^m) \quad (25)$$

$$M = \{M_1, M_2, ..., M_J\}, j = 1, ..., J \quad (26)$$

Because the length and point number of lines are variant, using a fixed value as the threshold to distinguish the similarity difference is not appropriate. Therefore, in this paper, we calculate the threshold in accordance with the sampling points on each line as the following equation

$$th^{d_i} = c_3 + \sum_{n=1}^{N^f} \left| x_{i,n}^{f,R\&S} \cos(\theta_i^f) + y_{i,n}^{f,R\&S} \sin(\theta_i^f) - \rho_i^f \right| \quad (27)$$

$$\begin{bmatrix} x_{i,n}^{f,R\&S} \\ y_{i,n}^{f,R\&S} \end{bmatrix} = \begin{bmatrix} \cos(th^{\Delta\theta}) & -\sin(th^{\Delta\theta}) \\ \sin(th^{\Delta\theta}) & \cos(th^{\Delta\theta l}) \end{bmatrix} \begin{bmatrix} x_{i,n}^f - x_k^r \\ y_{i,n}^f - y_k^r \end{bmatrix} + \begin{bmatrix} x_k^r \\ y_k^r \end{bmatrix} \quad (28)$$

where $x_{i,n}^{f,R\&S}$ and $y_{i,n}^{f,R\&S}$ is the n_{th} sampling point which is rotated by the center of the robot pose and the slope threshold. c_3 is the constant which is mainly determined by the moving speed of the robot.

If the difference is smaller than the threshold, this line is viewed as a matched one, otherwise, this line will be treated as a new one and be added into the map. When the line is matched with more than one lines in the map, the line with smallest difference will be chosen. After that, we can get the matched pairs, as depicted in Fig. 4(a).

Because the motion model of the robot used to calculate the rotated and shifted features of the observed lines are estimated only by the control command and some noises, the precision may be affected by many reasons. For further improving the accuracy both of the map and the localization, we use the matched lines to re-estimate the robot position. This step is also called a correction step in which we use the observation model to correct the robot pose.

The weighted average rotation slope and intercept difference of all matched lines are calculated first by the following equations

$$\Delta\bar{\theta} = \sum_{q=1}^{Q} l_q \Delta\theta_q \Big/ \sum_{q=1}^{Q} l_q \quad (29)$$

$$\Delta\bar{\rho}_q = \sum_{q=1}^{Q} \Delta\rho_q / \mathrm{Q} \quad (30)$$

where l_q , $\Delta\theta_q$, and $\Delta\rho_q$ are the length, the rotation slope, and the intercept difference of the q_{th} matched line pair, respectively. Q is the number of all matching pairs.

Because we use consecutive points to construct a line and extract its features, the longer the matched line is, the trustworthy the line is. Therefore, we weight these lines by their length. Fig. 4(b) is an illustration of the rotation slope and the intercept difference.

When we calculate the average rotation slope and the intercept difference, they are then used to all of the lines. Hence, we can calculate the position of the robot and its moving distance in Cartesian coordinate by the matched map, as shown in Fig. 5. We first define the moving distance of the robot as

$$\Delta x = x_k - x_{k-1} \quad (31)$$

$$\Delta y = y_k - y_{k-1} \quad (32)$$

Next, we have to estimate the moving distance by the established line. A line in the map in time *k*-1 can be represented as

$$(x_k - \Delta x)\cos(\theta_q^m) + (y_k - \Delta y)\sin(\theta_q^m) = (\rho_q^m + \Delta\rho_q) \quad (33)$$

where θ_q^m and ρ_q^m are the slope and the intercept of the established line on the map on the q_{th} matching pair. When we subtract (33) form (15), we get the following equation

$$-\Delta x \cos(\theta_q^m) - \Delta y \sin(\theta_q^m) = \Delta\rho_q \quad (34)$$

Then, we solve Δx and Δy by a weighted least square method, therefore we can get the moving distance of the robot, as following equation

$$\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix}=\begin{bmatrix} w_1\cdot\cos(\theta_1^m) & w_1\cdot\sin(\theta_1^m)\\ \vdots & \vdots\\ w_q\cdot\cos(\theta_q^m) & w_q\cdot\cos(\theta_q^m)\\ \vdots & \vdots\\ w_Q\cdot\cos(\theta_Q^m) & w_Q\cdot\sin(\theta_Q^m)\end{bmatrix}^{+}\begin{bmatrix}-w_1\cdot\Delta\rho_1\\ \vdots\\ -w_q\cdot\Delta\rho_q\\ \vdots\\ -w_Q\cdot\Delta\rho_Q\end{bmatrix} \quad (35)$$

where w_q is the weight of the q_{th} matching pair and it is calculated by

$$w_q=(l_q\Big/\sum_{q=1}^{Q}l_q) \quad (36)$$

After these calculation, the robot pose is corrected. When the robot motion model correction is done, the observation model of the line features is then adjusted by the corrected robot pose. Therefore, the next step is to update the estimated map.

There are three cases in the map updating according to the matching relationships between the established lines and the observed lines. In the first case, an observed line is not matched with any line in the map, so it will be treated as a new one and be added to the map.

In the second case, an established line does not match with any observed line. It means the line feature should exist and be observed, but it is not. This circumstance happens when the environment features are changed, such as an object or people in the map is moved. For ensuring the features on the map are really disappear, but not caused by the sensor noise. Only when the features are not observed continuously during 30 iterations, they are erased on the map.

In the third case, an observed line is matched with an established line, so these two line have to be merged into a line and its line features will be calculated to update the map. If we re-calculate all points on the merged line by (15)-(18), it will spend lots of computation load and time, therefore, we use a recursive least squares method [29] to calculate the merged line features by the following equations

$$g_q^{mg'}=\frac{(N_q^f g_q^{fg}+N_q^m g_q^{mg})}{(N_q^f+N_q^m)} \quad (37)$$

$$\theta_q^{m'}=\frac{(N_q^f\theta_q^f+N_q^m\theta_q^m)}{(N_q^f+N_q^m)} \quad (38)$$

$$\rho_q^{m'}=x_q^{mg}\cos\theta_q^m+y_q^{mg}\sin\theta_q^m \quad (39)$$

$$N_q^{m'}=N_q^f+N_q^m \quad (40)$$

where $\theta_q^m,\rho_q^m,g_q^{mg},N_q^m$ are the slope, intercept, center of gravity, and number of points of the established line, M_q ,on the q_{th} matching pair. $\theta_q^f,\rho_q^f,g_q^{fg},N_q^f$ are these features of the observed line, F_q , in the same pair. Using the above equations, we can get four features and construct a new line, next we have to calculate the two end points of this line. For doing this, we project these two lines onto the merged line M_q' and find the

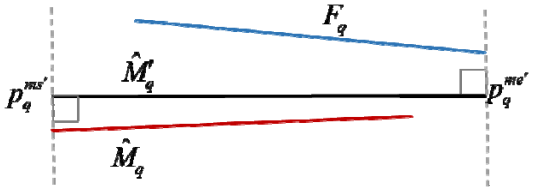


Fig. 6. The result of finding the endpoints of the merged line.

projected end points as shown in Fig 6. Hence, all features of the merged line are found, and the map will be updated by the updated lines.

In RBPF based SLAM system, all particles will update its map and robot pose by the calculations mentioned above, only some particles which have good fitness values will represent the possibility distribution. Therefore, we have to define a fitness function to evaluate these particles after them updating the features of the map.

Because we hope the observation model is as close to the built map as possible, the fitness function is defined as the following equations

$$fit=\frac{1}{R}\sum_{q=1}^{Q}w_q \quad (41)$$

$$w_q=N_q^f e^{-c4\cdot d_q} \quad (42)$$

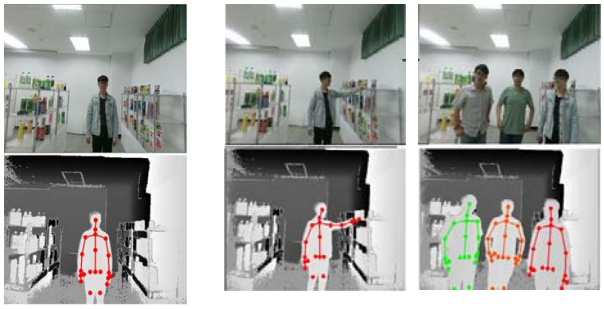
where R is the number of laser measurement points, w_q is the weight of the matching pair q. d_q is the distance between the map line and the observed line and it can be calculated by (24). $c4$ is a constant, in this paper, we set it as 0.005.

Sometimes, the performance of the particles are not good enough, and they need to be resampled. This means the particles which have low fitness values will be replaced by the particles with high fitness values. However, if we resample the particles each time, the particle depletion problem may occur which means the important particles are depleted. Therefore, we adapt a selective resampling method [30] to solve this problem.

In this method, the quantity of the current particles is calculated by the following equation

$$Qua=\frac{1}{\sum_{d=1}^{D}(fit_d)^2} \quad (43)$$

where fit_d is the fitness value of d_{th} particle and D is the number of particles. The concept of the quantity calculation is when this particle set can well represent the real environment, the map and the robot pose in all particles will similar and results in the low variance between all particles. Otherwise, when the variance become large, it means they cannot represent the real situation. Therefore, we only resample particles when the quantity value is lower than $D/2$ [31]. This selective resampling method can not only reduce the risk of the particle depletion problem and but also decrease the computation load and time.



(a) (b) (c)

Fig.7 The results of skeleton detection and recognition.

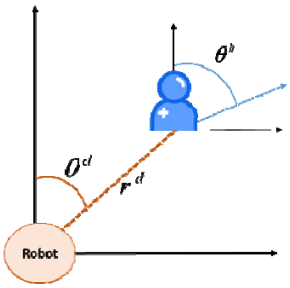


Fig. 8. Illustration of relative position and moving angle.

III. SLAM FOR INTERACTION WITH HUMAN BEINGS

SLAM is a basic function for a mobile interacting robot because it allows the robot knows where it is and where it will go to. However, for real applications, there are some other functions have to be integrated in, or else, the robot cannot provide real personal services. Therefore, we develop and integrate three useful functions with the presented RBPF SLAM system, including path planning function, human robot interaction system, and accompany walking strategy.

A. Path planning and Object location establishment

When the robot has the environment map, it also needs a path planning function to plan its route between two specific positions. We first reduce the map size from 600x600 pixels to 100x100 pixels, then inflate the map, and use the A-star algorithm to plan a path. However, this path contains too many points and it results in the robot move slowly and discontinuously. Therefore, we design a simplification method to reduce the path points.

First, we detect the points on the same line, and discard all points on the line except for the first and the last point. And then, we consider the distance between two path points. If the distance is longer than a predefined threshold, a weighted average point of these two points will be calculated to replace these two points. After these, the path will become compact which allows the robot move safely and smoothly.

Another basic function relative to SLAM is to mark the location of items. In this part, robot must integrate object recognition function to find the particular object and record its location in the map. When the robot successfully identifies the object, it will record its current location and the angle of head as the position of the object. Therefore, next time the robot needs to find or grasp the object, it will know where it should go to.

B. Human robot interaction system and GUI

For interacting robots, how to interact with people is the most important issue and it affects experience of human directly. People expect convenient, feasible, and intelligent interaction manners. Therefore, we design a graphical user interface (GUI) and integrate face recognition, skeleton recognition, and speech system to provide a suitable interaction. For the robot, variant interaction manners allow it provides more feasible and stable interaction, while for human beings, this means there are more choices they can choose.

The first function integrated in the human robot interaction system is the person recognition. The robot recognizes a person by his/her face and skeleton. The face recognition function utilizes the Haar-like feature [32] to detect the face and the Fisherface method [33] to establish the face models. Once the robot starts to provide a service to a person, it detects the face form the images obtained by its camera, and trains a face model of the person. So, it can ensure the person it service to is the correct one. The skeleton detection and recognition ability is a backup function for person recognition. When his/her face is unable to be seen, the robot recognizes by his/her skeleton information.

The skeleton detection also provides another function, avoiding a person. When the robot moving around the environment or following a person, it can avoid the other people by detecting them. Some detection and recognition results are shown in Fig. 7, in which the first detected person are marked as red color, as Fig. 7(a), and the other two persons are marked as orange and green colors, as shown in Fig. 7(c).

There are two main manners the robot interacts with the person, a GUI and a speech system. The robot will introduce itself before starting service, and recognize the command form the person. For example, when the robot accompanies the person shopping, the person asks the robot to help him/her pick a merchandise, the robot can receive the command and execute it. The GUI is mounted on a touch screen by which a person can choose the service he/she needs, and the corresponding information will be shown on the screen.

Speech is the most natural interaction manner for human being, however, the accuracy of speech recognition decreases when the words and sentences increase. On the contrary, the touch screen is the most accurate and trustworthy manner among the human robot interaction. Combining these two interaction manners provides a convenient and feasible way. At the first, the person uses the GUI to command the robot, so the robot can receive the command fast and correct. During the service, the person can command the robot by the current situation through speech. It means that the person need not stop the motion what he/she is doing and move to the back of the robot to operate the touch screen, instead, he/she can speak to the robot directly.

C. Accompanying walking strategies

Accompanying walking with a person is another important function for the robot while interacting with human beings. This function can use in many situations, such as accompanying a person to shop, following a person to a specific position, learning new information about the environment. However, maintaining a suitable position and walking speed relative to the person is not an easy problem.

In psychology researches, the suitable and comfortable distance between two persons are determined by the relationship of these two persons. Based on the classification of Hall [20], the robot should maintain a social distance (typically between 1.2 and 3.5 m) which allows the robot to receive commands directly and not to encroach on the personal area of the person.

In addition to maintain a distance, we also expect the robot can maintain a relative angle with the person, as shown in Fig 8, in which r^d and θ^d is the desired relative distance and angle between the person and the robot, respectively. For doing so, first, the robot needs to detect and to calculate the position of the person, in this paper, we do these by a RGBD camera. The person's position in vision coordinate will be transfer to the world coordinate.

Next, the robot has to predict the walking velocity of the person. We use the Taylor series expansion (TSE) velocity estimator [34] to implement, because it has a good response velocity transients and is easy to implement. Theoretically speaking, the higher order terms we use to estimate the velocity, the more precise the estimated velocity is. However, there are more noises in higher order terms in practice. Therefore, we use the 3rd order TSE to estimate the velocity of human. In discrete signal processing, we fix every time step which we sample the position, so the 3rd order TSE estimator can be written as follows:

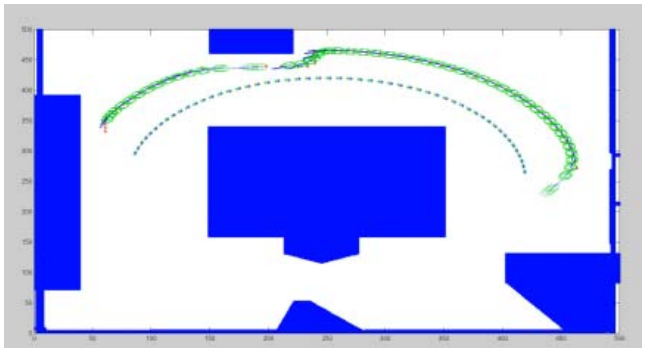
$$\hat{\mathbf{v}}_k = \Delta\mathbf{x}_k + \frac{1}{2}(\Delta\mathbf{x}_k - \Delta\mathbf{x}_{k-1}) + \frac{1}{8}(\Delta\mathbf{x}_k - 2\cdot\Delta\mathbf{x}_{k-1} + \Delta\mathbf{x}_{k-2}) \quad (44)$$

where $\Delta\mathbf{x}_k$ is the displacement in time step k and $\Delta\mathbf{x}_k = x_k - x_{k-1}$. $\hat{\mathbf{v}}_k$ is the estimated velocity in time step k and it can be represented as $\hat{\mathbf{v}}_k = [v_k^x \quad v_k^y]^T$.

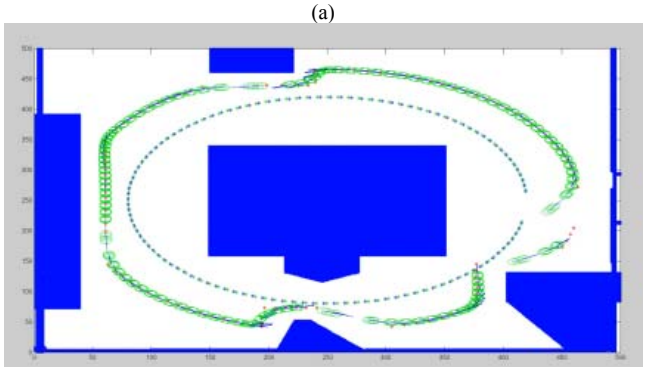
Following, the robot has to calculate a suitable position it should move to and the position can be calculated by the following equation

$$\begin{bmatrix} x_{k+1}^r \\ y_{k+1}^r \end{bmatrix} = \begin{bmatrix} x_k^h \\ y_k^h \end{bmatrix} + R\begin{bmatrix} r^{dx}\cdot v_k^x \\ r^{dy}\cdot v_k^y \end{bmatrix} \quad (45)$$

where x_k^h and y_k^h are the position of the person. $r^d = [r^{dx} \quad r^{dy}]^T$ is the desired relative distance between the person and the robot. R is a rotation matrix which transfers the vision observation coordinate to the robot control coordinate and it can be calculated by the following equation,



(a)



(b)

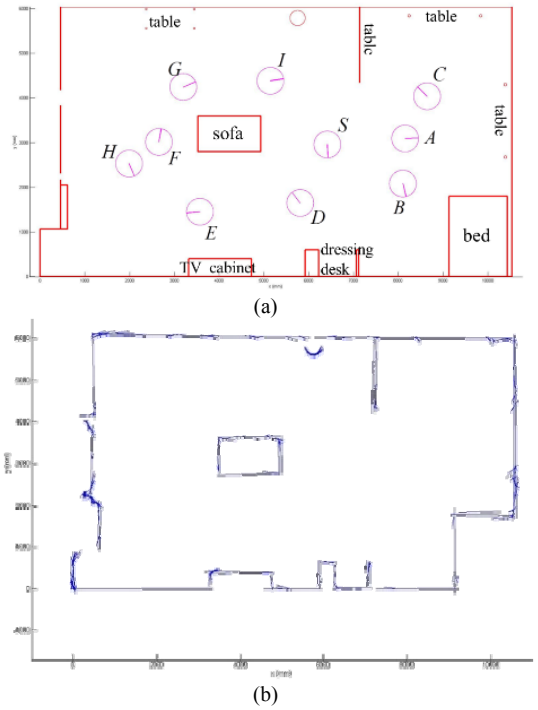
Fig. 9. simulation results of accompany walking strategy.

$$R = \begin{bmatrix} \cos(\theta^d - \theta^h) & -\sin(\theta^d - \theta^h) \\ \sin(\theta^d - \theta^h) & \cos(\theta^d - \theta^h) \end{bmatrix} \quad (46)$$

where θ^h is the moving angle of the person. Because the robot always faces the position of the person and tries to maintain the person in her vision center, θ^d can be treated as the moving angle of the robot in the world coordinate, the subtraction angel between θ^h and θ^d allows the robot to tune itself and to keep facing the person.

Based on the planned position, the robot can maintain a relative position of the person. However, if there is an obstacle on the way of the robot, obstinately maintain the relative position is not workable. Instead, the robot should adjust the relative angle to avoid the obstacle.

Fig 9 shows a simulation result of following a person. The relative distance and angle are set as $\pi/6$ and 0.9m. The positions of the person given by the simulator and estimated by the TSE estimator are shown as the green and blue points in the inner cycle in Fig 9, respectively. The calculated suitable robot positions are shown as the red points in the outer cycle, and the green cycles are the results of robot path planning. One can see, these two trajectories maintain a suitable distance and the robot maintain a position behind the person. This simulation demonstrates the efficiency even in some narrow areas.



(a)

(b)

Fig. 10. The performance of the SLAM system. (a) The graphic configuration of the experimental field and the sampling points. (b) The constructed map by SLAM system.

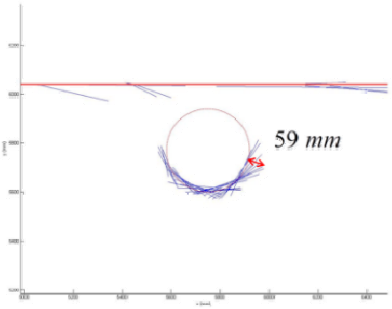


Fig. 11 The constructed map of a circle item.

IV. EXPERIMENTS

A. *Accuracy of SLAM*

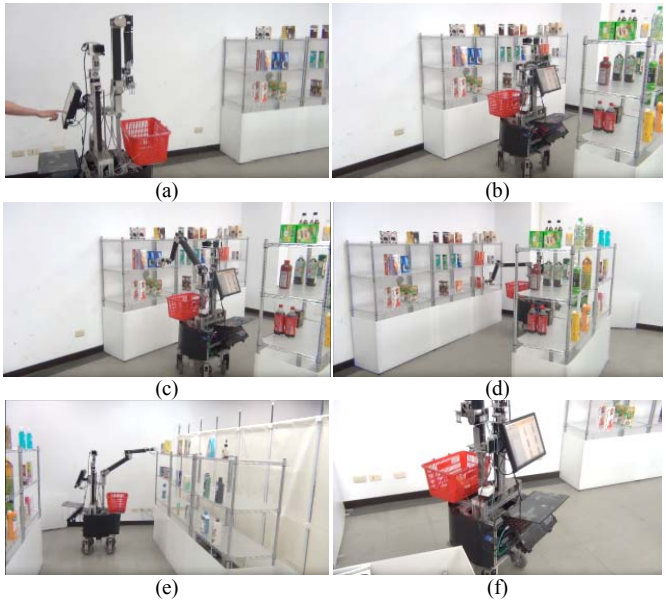
We first examine the accuracy of our RBPF SLAM system. The experiment field is about 6×10 m^2, and some furniture were placed inside. Fig. 10(a) is the configuration of the experiment field and Fig. 10(b) is the constructed map by the SLAM system. Generally speaking, the map built by the proposed SLAM system is really similar with the real environment.

In this environment, the trash can is the difficult item, because its shape. Because we choose piecewise linear feature as the basis of the SLAM system to reduce the computation complexity, circle object will be represented as many lines. Fig. 11 shows the details of the map and the difference of the trash can. One can see the proposed SLAM can also deal with the circle environment with an acceptable error.

The accuracy of localization was examined by sampling 10 points in this environment, driving the robot to each sampling point, and checking the localization results. For each location, we measured 100 times and the results are tabulated in Table 1. The average error of x-axis and y-axis are 38.4 millimeters and 101.3 millimeters, respectively, and the average error of orientation is 0.39 degree. Though it is not the most accurate SLAM method so far, it is good enough for an interacting robot.

TABLE 1 THE ACCURACY OF LOCALIZATION.

Point	Error in x-axis(mm)		Error in y-axis(mm)		Error in orientation (degree)	
Point	mean	Std.	mean	Std.	mean	Std.
S	12.3	7.57	8.09	2.37	-0.44	0.19
A	1.21	17.65	89.77	6.88	0.12	0.24
B	2.21	3.48	62.23	1.68	-0.66	0.19
C	21.9	3.06	39.44	4.68	-0.03	0.11
D	-15.4	4.62	-87.94	4.61	-0.97	0.11
E	68.1	8.19	-101.15	4.99	-0.46	0.27
F	37.5	2.79	-257.3	7.80	-0.35	0.08
G	12.6	16.97	-145.14	11.87	0.07	0.23
H	95.9	10.92	-143	2.07	-0.18	0.18
I	-116.7	3.57	-79.23	2.63	0.59	0.11
Mean	38.4	7.88	101.3	4.96	0.39	0.17



(a) (b) (c) (d) (e) (f)

Fig. 12 Snapshots of the experiment I. (a) The customer placed an order by the touch screen. (b) The robot moved to the location of the first merchandise. (c) The robot recognized and grasped the first object. (d) (e) The robot grasped the second and third objects. (f) The robot went back to the customer.

B. *Interacting application scenarios*

For demonstrating the interacting functions, we set up another experiment field and constructed two application scenarios. The experiment field simulated a market with eight shelves and more than 60 kinds of merchandises, such as foodstuff, cookies, beverages, and the articles for daily use. The first scenario is called 「Help Me Shop」, and the snapshots of the experiment are shown in Fig. 12. In this scenario, the robot will help the customer to take all merchandises he/she wants, so that the customer need not to walk around the supermarket. First, the customer placed a shopping order by the touch screen, as Fig 12 (a), the robot then received the shopping list, calculated the optimal picking sequence, and planned the route by the SLAM system and the location of the merchandises. After that, the robot picked the merchandise sequentially, as shown in Fig. 12

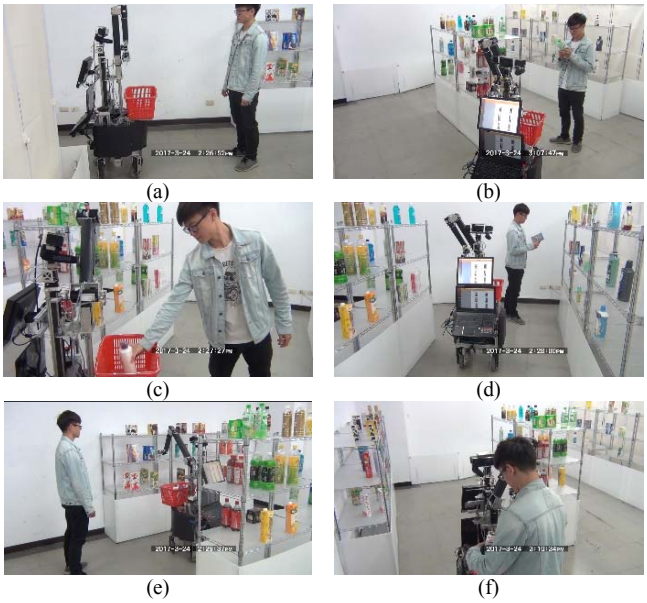


Fig. 13 Snapshots of the experiment II. (a) The robot detected the skeleton of the customer. (b) The robot followed the customer by tracking his skeleton. (c) (d) The robot continuously kept a safe distance with the customer during shopping. (e) The customer asked the robot to grasp a good and the robot recognized the command and grasped it. (f) The customer commanded the end of the shopping.

(b)-(e). Finally, the robot passed all merchandises on the shopping cart to the customer, as Fig. 12 (f).

This experiment has demonstrated the efficiency of the integration of the SLAM system, object recognition system, and the human robot interaction GUI. The robot used the SLAM system to move to the correct positions of the merchandises, and grasped the merchandises accuracy no matter where the merchandise placed.

The second experiment scenario is called 「Stand by Me」. In this scenario, the robot follows the customer with a relative distance while he/she carries the goods, so that the customer need not carry the shopping cart. The robot first detected the skeleton of the customer, and kept tracking him and maintaining a suitable distance during shopping. Some snapshots are shown in Fig. 13. The robot illustrated itself first and recorded the skeleton of the customer, as Fig 13 (a). When the customer shopped in the market, the robot followed the person and helped him carrying the goods, as Fig. 13 (b)-(d). In the meantime, the detected skeleton was used to calculate the distance between the robot and the customer to maintain a fixed distance. Besides, the customer can command the robot to help him take the good on the shelf anytime during shopping, as shown in Fig.13 (e). When the customer wanted to end the shopping, he commanded the robot by speech, and the robot stopped, as Fig. 13 (f). Different from the first scenario, this one demonstrated another two interaction matters, speech communication and skeleton recognition, and illustrated the feasibility of the interaction. The robot can recognize the customer by several ways and communicate with him intelligent.

V. Conclusion

This paper has proposed a piecewise linear feature based RBPF SLAM along with some application functions for interacting service robots. The proposed SLAM system allows the robot to construct the environment map and to localize itself before or during providing personal services. Besides, we also integrated some other functions with the SLAM system to really provide personal services, including path planning, object location establishment, human robot interaction system, and accompany walking strategy. Therefore, the robot can recognize the person it interacts with, receive commands by the human robot interaction system, establish the location of an object, plan a path, and move to the object. The effectiveness of the proposed SLAM system has been illustrated in the experiments, the accuracy of established map and robot location are both good enough for a service robot. We also constructed a market scenario with two useful applications to demonstrate what the robot can do and how it interacts with human. The experiments have demonstrated the robot integrated several functions to provide feasible and intelligent interaction, and further provided the customer convenient and comfortable shopping experiences.

References

[1] R. O. Castle, G. Klein, and D. W. Murray, "Combining Monoslam with Object Recognition for Scene Augmentation using a Wearable Camera," *Image and Vision Computing*, vol. 28, no. 11, pp. 1548-1556, 2010.

[2] L. Zhao, S. Huang, and G. Dissanayake, "Linear Monoslam: A Linear Approach to Large-Scale Monocular SLAM Problems," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1517-1523, 2014.

[3] C. Audras, A. Comport, M. Meilland, and P. Rives, "Real-Time Dense Appearance-Based SLAM for RGB-D Sensors," in *Proceedings of Australasian Conference on Robotics and Automation*, vol. 2, 2011.

[4] J. D. Fossel, K. Tuyls, and J. Sturm, "2D-SDF-SLAM: A Signed Distance Function Based SLAM Frontend for Laser Scanners," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1949-1955, 2015.

[5] J. Sola, T. Vidal-Calleja, J. Civera, and J. M. Montiel, "Impact of Landmark Parametrization on Monocular EKF-SLAM with Points and Lines," *International Journal of Computer Vision*, vol. 97, no. 3, pp. 339-368, 2012.

[6] D. Wang, H. Liang, T. Mei, H. Zhu, J. Fu, and X. Tao, "Lidar Scan Matching EKF-SLAM using the Differential Model of Vehicle Motion," in *Proceeding of IEEE Intelligent Vehicles Symposium*, pp. 908-912, 2013.

[7] S. Wen, X. Chen, C. Ma, H. K. Lam, and S. Hua, "The Q-Learning Obstacle Avoidance Algorithm Based on EKF-SLAM for NAO Autonomous Walking under Unknown Environments," *Robotics and Autonomous Systems*, vol. 72, pp. 29-36, 2015.

[8] S. Ahn, M. Choi, J. Choi, and W. K. Chung, "Data Association using Visual Object Recognition for EKF-SLAM in Home Environment," in *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2588-2594, 2006.

[9] S. Ahn, J. Choi, N. L. Doh, and W. K. Chung, "A Practical Approach for EKF-SLAM in an Indoor Environment: Fusing Ultrasonic Sensors and Stereo Camera," *Autonomous Robots*, vol. 24, no. 3, pp. 315-335, 2008.

[10] H. C. Lee, S. H. Lee, M. H. Choi, and B. H. Lee, "Probabilistic Map Merging for Multi-Robot RBPF-SLAM with Unknown Initial Poses," *Robotica*, vol. 30, no. 2, pp. 205-220, 2012.

[11] S. Nasuriwong and P. Yuvapoositanon "Posterior Elimination Fast Look-Ahead Rao-Blackwellized Particle Filtering for Simultaneous Localization and Mapping," *Procedia Computer Science*, vol. 86, pp. 261-264, 2016.

[12] B. D. Gouveia, D. Portugal, and L. Marques, "Speeding up Rao-Blackwellized Particle Filter SLAM with a Multithreaded

Architecture," in *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1583-1588, 2014.
[13] M. Li, R. Lin, H. Wang, and H. Xu, "An Efficient SLAM System Only using RGBD Sensors," in *Proceeding of IEEE International Conference on Robotics and Biomimetics*, pp. 1653-1658, 2013.
[14] X. L. Tang, L. M. Li, and B. J. Jiang, "Mobile Robot SLAM Method Based on Multi-Agent Particle Swarm Optimized Particle Filter," *The Journal of China Universities of Posts and Telecommunications*, vol. 21, no. 6, pp. 78-86, 2014.
[15] H. J. Sohn and B. K. Kim, "An Efficient Localization Algorithm Based on Vector Matching for Mobile Robots using Laser Range Finders," *Journal of Intelligent and Robotic Systems*, vol. 51, no. 4, pp. 461-488, 2008.
[16] Z. Zhang and Z. Zhao, "A multiple mobile robots path planning algorithm based on A-star and Dijkstra algorithm," *International Journal of Smart Home*, vol. 8, no. 3, pp. 75-86, 2014.
[17] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59-69, 2014.
[18] R. Murakami, L. Y. Morales Saiki, S. Satake, T. Kanda, and H. Ishiguro, "Destination unknown: walking side-by-side without knowing the goal," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction,* pp. 471-478, 2014.
[19] Y. Morales, T. Kanda, and N. Hagita, "Walking together: side by side walking model for an interacting robot," *Journal of Human-Robot Interaction*, vol. 3, no.2, pp. 51-73, 2014.
[20] E. T. Hall, *The Hidden Dimension*. Garden City, NY: Doubleday, 1966.
[21] K. Dautenhahn, M. Walters, S. Woods, K. L. Koay, C. L Nehaniv, A. Sisbot, and T. Siméon, "How may I serve you?: a robot companion approaching a seated person in a helping context," In *Proceedings of the ACM SIGCHI/SIGART conference on Human-robot interaction*, pp. 172-179, 2006.
[22] V. Rousseau, F. Ferland, D. Létourneau, and F. Michaud, "Sorry to interrupt, but may I have your attention? Preliminary design and evaluation of autonomous engagement in HRI," *Journal of Human-Robot Interaction*, vol. 2, no. 3, pp. 41-61, 2013.
[23] Y. Shimizu, et al., "An Interactive Support System for Activating Shopping Streets Using Robot Partners in Informationally Structured Space," in *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts*, Tokyo, Japan, November 7-9, 2013
[24] K. Takeuchi, et al., "Consideration and Evaluation of Benefits of Human Shaped Service Robots," in *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts*, Tokyo, Japan, Nov. 7-9, 2013.
[25] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, "A robot that approaches pedestrians," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 508-524, 2013.
[26] M. Goller, et al., "Sharing of Control between an Interactive Shopping Robot and its User in Collaborative Tasks" in *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication*, Viareggio, Italy, Sept. 12-15, 2010.
[27] K.-C. Cheng, *Design and Implementation of SLAM and Behavior Strategy for Home Service Robot*, M.S. Thesis, Department of Electrical Engineering, National Cheng Kung University, Tainan 701, Taiwan, 2010.
[28] B. W. Kuo, H. H. Chang, Y. C. Chen, and S. Y. Huang, "A Light-and-fast Slam Algorithm for Robots in Indoor Environments using Line Segment Map," *Journal of Robotics*, 2011.
[29] H. J. Sohn and B. K. Kim, "VecSLAM: An Efficient Vector-based SLAM Algorithm for Indoor Environments," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 3, pp. 301-318, 2009.
[30] G. Grisetti, C. Stachniss, and W. Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," in *Proceeding of IEEE International Conference on Robotics and Automation*, pp. 2432-2437, 2005.
[31] J. S. Liu, "Metropolized Independent Sampling with Comparisons to Rejection Sampling and Importance Sampling," *Statistics and Computing*, vol. 6, no. 2, pp. 113-119, 1996.
[32] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *Vision and Pattern Recognition*, pp. 511-518, 2001.
[33] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 19, no. 7, pp. 711-720, 1997.
[34] R. H. Brown, S. C. Schneider, and M. G. Mulligan, "Analysis of Algorithms for Velocity Estimation from Discrete Position versus Time Data," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 1, pp. 11-19, 1992.

Chih-Yin Liu received the B.S. degree in Department of Social Work from National Taiwan University, Taipei, Taiwan, R.O.C., in 2003. She received the M.S. degree in the Department of Psychology from National Chung Cheng University, Chia-Yi, Taiwan, R.O.C., in 2005. She received the Ph.D. degree in the Department of Electrical Engineering, National Cheng-Kung University, Tainan, Taiwan, R.O.C., in 2017, and keeps studying in robotics as a postdoctoral fellow until now. Her major research interests include intelligent control system, robot learning, home service robot, robot cooperation, and human-robot interaction.

Chung-Lin Lee received the B.S. and M.S. degree from the Department of Electrical Engineering from National Cheng Kung University, Tainan, Taiwan, R.O.C., in 2014 and 2016, respectively. His major research interests include home service robot, robot system design, mechanical drawing, robotic manipulator motion control, and simultaneous localization and mapping. He is currently a firmware engineer at Phison Electronics Corp., Taiwan, R.O.C.

Jie-Jhong Liang received the B.S. degree from the Department of Mechanical Engineering in 2016 and the M.S. degree in the Department of Electrical Engineering in 2018 from National Cheng Kung University (NCKU), Tainan, Taiwan, R.O.C. His major research interests include home service robot, robotic manipulator motion control, and system modeling. He is currently a firmware engineer at Delta Electronics, Inc., Taiwan, R.O.C.

Chih-Yen Chen received the B.S. degree from the Department of Mechanical Engineering in 2016 and the M.S. degree in the Department of Electrical Engineering in 2018 from National Cheng Kung University (NCKU), Tainan, Taiwan, R.O.C. His major research interests include home service robot, robotic manipulator motion control, and system modeling. He is currently an engineer at Micron Technology, Inc., Taiwan, R.O.C.

Tzuu-Hseng S. Li received the B.S. degree in electrical engineering from the Tatung Institute of Technology, Taipei, Taiwan (R.O.C.), in 1981, and the M.S. and Ph.D. degrees in electrical engineering from the National Cheng Kung University (NCKU), Tainan, Taiwan, in 1985 and 1989, respectively. Since 1985, he has been with the Department of Electrical Engineering, NCKU, where he is currently a Distinguished Professor. From 1996 to 2009, he was a Researcher with the Engineering and Technology Promotion Center, National Science Council, Tainan. From 1999 to 2002, he was the Director of the Electrical Laboratories, NCKU. From 2009 to 2012, he was the Dean of the College of Electrical Engineering and Computer Science, National United University, Miaoli City, Taiwan. He has been the Vice President of the Federation of International Robot-Soccer Association since 2009. He has been the Director of the Center for Intelligent Robotics and Automation, NCKU, since 2014. His current research interests include artificial and/or biological intelligence and applications, fuzzy system and control, home service robots, humanoid robots, mechatronics, 4WIS4WID vehicles, and singular perturbation methodology.

Dr. Li was a recipient of the Outstanding Automatic Control Award in 2006 from the Chinese Automatic Control Society (CACS) in Taiwan and received the Outstanding Research Award in 2017 from the Ministry of Science and Technology, Taiwan. He was a Technical Editor of the IEEE/ASME *Transactions on Mechatronics* and an Associate Editor of the *Asia Journal of Control*. He is currently an Associate Editor of the *International Journal of Electrical Engineering*, the *International Journal of Fuzzy Systems*, and the *IEEE Transactions on Cybernetics*. He was elected as the President of the CACS from 2008 to 2011 and the Robotics Society of Taiwan from 2012 to 2015. He was elevated to CACS Fellow and RST Fellow in 2008 and 2018, respectively.

ARIS2019 / NCAR2019

2019 International conference on Advanced Robotics and Intelligent Systems

2019 National Conference on Advanced Robotics

Taipei Nangang Exhibition Center, Taiwan, August 20-23, 2019

aris2019.nchu.edu.tw | ncar2019.nchu.edu.tw

ARIS2019

Host by

RST (Robotics Society of Taiwan)

NCHU (National Chung Hsing University) & ISU (I-Shou University)

TAIROA (Taiwan Automation Intelligence and Robotics Association)

Technically Sponsored by

IEEE SMC Society (Systems, Man and Cybernetics)

Co-Sponsored by

MOST, Taiwan, R.O.C. (Ministry of Science and Technology)

AI Robotics Hub @ CTSP(Central Taiwan Science Park)

CTSP (Central Taiwan Science Park)

IEEE RA Society, Taipei Chapter (Robotics and Automation)

CACS (Chinese Automatic Control Society)

IFSA (International Fuzzy Systems Association)

KROS (Korea Robotics Society), Korea

IEEE SMC Society, Taichung Chapter (Systems, Man and Cybernetics)

IEEE Computational Intelligence Society, Taipei Chapter

NCAR2019

主辦單位：

台灣機器人學會 (RST)

國立中興大學 (NCHU) & 義守大學(ISU)

台灣智慧自動化與機器人協會 (TAIROA)

技術協助：

IEEE SMC學會

協辦單位：

中華民國科技部

中科智慧機器人自造基地

中部科學工業園區管理局(CTSP)

IEEE SMC學會台中分會

韓國機器人學會

IEEE 機器人與自動化學會 (RA) 台北分會

國際模糊系統學會 (IFSA)

中華民國自動控制學會 (CACS)

eTop 科技部工程科技推廣平台

Call for Paper

Topics of interest include but are not limited to
論文主題希望以下列內容為主，但並未強制限定

3D Perception	Automation Intelligence	Big-Data Analytics for Industry 4.0
Brain Machine Interface System	Cloud Computing for Industry 4.0	Cognitive Robots
Cyber-Physical Systems for Industry 4.0	Education Robots	Embedded Robot and Intelligence
Entertainment and Amusement Robots	Fuzzy Control and Robotics	Home Robots and Service Robots for Elder and Children
Human-Robot Interactions	Humanoid Robot	Indoor and Outdoor Service Robots
Inspection Robots	Intelligent Automated System and Its Applications	Intelligent Control for Assistive Robots
Intelligent Control Systems	Intelligent Manufacturing Systems	Intelligent Medical Systems and Applications
Intelligent Robotics and Systems	Intelligent Systems for Smart City	Internet of Things (IoT) for Industry 4.0
Localization and SLAM	Medical Robots	Multi-Agent Systems
Multi-Robot Systems	Rehabilitation Robot & Caring Robot	Rescue Robots and Hazardous Duty Robots
Security Robots	System Integration Technology	Wheelchair Robots
Wireless and Intelligent Sensor Fusion	Other Robotic Technologies	Other Robot-related Automation .

ARIS2019 Contributed Papers:

[Regular paper]
Agree to be included in IEEE Xplore.

[Position paper]
Excluded in IEEE Xplore.

[Presentation-Only paper]
Excluded in IEEE Xplore.

Important Dates /重要日期

May 10, 2019.	**Proposals of organized/invited sessions**
May 31, 2019.	**Submission of contributed papers**
June 28, 2019.	**Notification of acceptance**
July 26, 2019.	**Final papers due**

議程提案截止	2019/05/10
論文投稿截止	2019/05/31
論文接受通知	2019/06/28
論文完稿截止	2019/07/26

Contest Information / 競賽資訊

ARIS2019

Best Conference/Presentation/Student Paper Contest:

Select your interesting paper contest(s) while uploading your manuscript electronically.

NCAR2019

最佳研討會/學生論文競賽：

大會將根據投稿論文與學生論文內容與現場發表之表現，評選優秀論文若干篇，頒發獎狀與獎金以茲鼓勵。

ARIS2019 Conference Contact Information:

Web site: http://aris2019.nchu.edu.tw
Secretariat: Advanced Electrical Control Lab. @ National Chung-Hsing University (NCHU)
Phone: +886-4-22851549 ext. 601
Fax: +886-4-22856232
Address: Department of Electrical Engineering, 145, XingDa Road, Taichung 40227, Taiwan, R.O.C.
E-mail: aris2019@nchu.edu.tw

NCAR2019 研討會聯絡資訊：

研討會網址：http://ncar2019.nchu.edu.tw
秘書處：國立中興大學 尖端電控實驗室
電話：04-2285-1549#601
傳真：04-2285-6232
地址：(40227)臺中市南區興大路145號電機大樓601室
電子郵件：ncar2019@nchu.edu.tw

Information for Authors

Aim and Scope

The ***i*Robotics** is an official journal of Robotics Society of Taiwan (RST) and is published quarterly. The ***i*Robotics** will consider high quality papers that deal with the theory, design, and application of intelligent robotic system, intelligent artificial system, and extension theory systems ranging from hardware to software. Survey and expository submissions are also welcome. Submission of a manuscript should indicate that it has been neither published, nor copyrighted, submitted, accepted for publication elsewhere (except that the short version may have been presented at the conferences). Submitted manuscripts must be typewritten in English and as concise as possible.

Process for Submission of a Manuscript

The ***i*Robotics** publishes two types of articles: regular papers and technical notes. All contributions are handled in the same procedure, where each submission is reviewed by an associate editor who makes an initial decision to send the manuscript out for peer review or to reject without external review. Articles can be rejected at this stage for a variety of reasons, such as lack of novelty, topics outside the scope of the Journal, flaws in the scientific validity, or unprofessional presentation. We are therefore not normally able to provide authors with feedback on rejected manuscripts. If the associate editor believes the article may be of interest to our readers, it is then sent out for external peer review by at least two external reviewers. According the recommendation of the associate editor, the Editor-in-Chief makes the final decision. All manuscripts should be submitted electronically in Portable Document Format (PDF) through the manuscript submission system at [http://www.rst.org.tw]. The corresponding author will be responsible for making page proof and signing off for printing on behalf of other co-authors. Upon acceptance of a paper, authors will be requested to supply their biographies (100 to 200 words) and two copies of the final version of their manuscript (in DOC format and in PDF format).

Style for Manuscript

Papers should be arranged in the following order of presentation:

1) First page must contain: a) Title of Paper (without Symbols), b) Author(s) and affiliation(s), c) Abstract (not exceeding 150 words for Papers or 75 words for Technical Note, and without equations, references, or footnotes), d) 4-6 suggested keywords, e) Complete mailing address, email address, and if available, facsimile (fax) number of each author, f) Preferred address for correspondence and return of proofs, and g) Footnotes (if desired).
2) The text: Submitted manuscripts must be typewritten double-spaced. All submitted manuscripts should be as concise as possible. Regular papers are normally limited to 26 double-spaced, typed pages, and technical notes are 12 double-spaced, typed pages. Please see the Page charge for those who want to submit long papers.
3) Acknowledgements of financial or other support (if any).
4) References: References should be numbered and appear in a separate bibliography at the end of the paper. Use numerals in square brackets to cite references, e.g., [15]. References should be complete and in the style as follows.
   [1] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller - Part I," *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 2, pp. 404-418, 1990.
   [2] C. Golaszewski and P. Ramadge, "Control of discrete event processes with forced events," in *Proc. of 26th IEEE Conf. Decision and Control*, Los Angeles, CA, pp. 247-251, Dec. 1987.
   [3] P. E. Wellstead and M. B. Zarrop, *Self-Tuning Systems*, New York: Wiley, 1991.
   [4] Project Rezero, available at http://rezero.ethz.ch/project_en.html (last visited: 2017-07).
5) Tables
6) Captions of figures (on separate sheet of paper)

Style for Illustrations

1) It is in the author's interest to submit professional quality illustrations. Drafting or art service cannot be provided by the Publisher.
2) Original drawings should be in black ink on white background. Maximum size is restricted to 17.4 by 24.7 cm. Glossy prints of illustrations are also acceptable.
3) All lettering should be large enough to permit legible reduction of the figure to column width, sometimes as small as one quarter of the original size. Typed lettering is usually not acceptable on figures.
4) Provide a separate sheet listing all figure captions, in proper style for the typesetter, e.g., "Fig. 5. The error for the proposed controller."
5) Illustrations should not be sent until requested, but authors should be ready to submit these immediately upon acceptance for publication.

Page Charges

After a manuscript has been accepted for publication, the author's company or institution will be approached with a request to pay a page charge to cover part of the cost of publication. The charges include:

1) NT$ 5000 for the 10 printed pages of a full paper or for the 6 printed pages of a short paper, and the excess page charge of NT$ 1500 per extra printed page for both full and short papers.
2) For color figures or tables, an additional cost will be charged. The cost, depending on the number of color figures/tables and the final editing result, will be given upon the acceptance of this paper for publication.

Copyright

It is the policy of the RST to own the copyright of the technical contributions. It publishes on behalf of the interests of the RST, its authors, and their employers, and to facilitate the appropriate reuse of this material by others. Authors are required to sign a RST Copyright Form before publication.

Manuscripts (in PDF Format) Submission Website: http://www.rst.org.tw

Editor-in-Chief: Ching-Chih Tsai, Department of Electrical Engineering, National Chung Hsing University, Taiwan
Email: cctsai@nchu.edu.tw
Tzuu-Hseng S. Li, Department of Electrical Engineering, National Cheng Kung University, Taiwan
Email: thsli@mail.ncku.edu.tw

Managing Editor: Dr. Feng-Chun Tai, Department of Electrical Engineering, National Chung Hsing University, Taiwan
Email: fctai@nchu.edu.tw

iRobotics

VOLUME 1, NUMBER 4 DECEMBER, 2018

CONTENTS