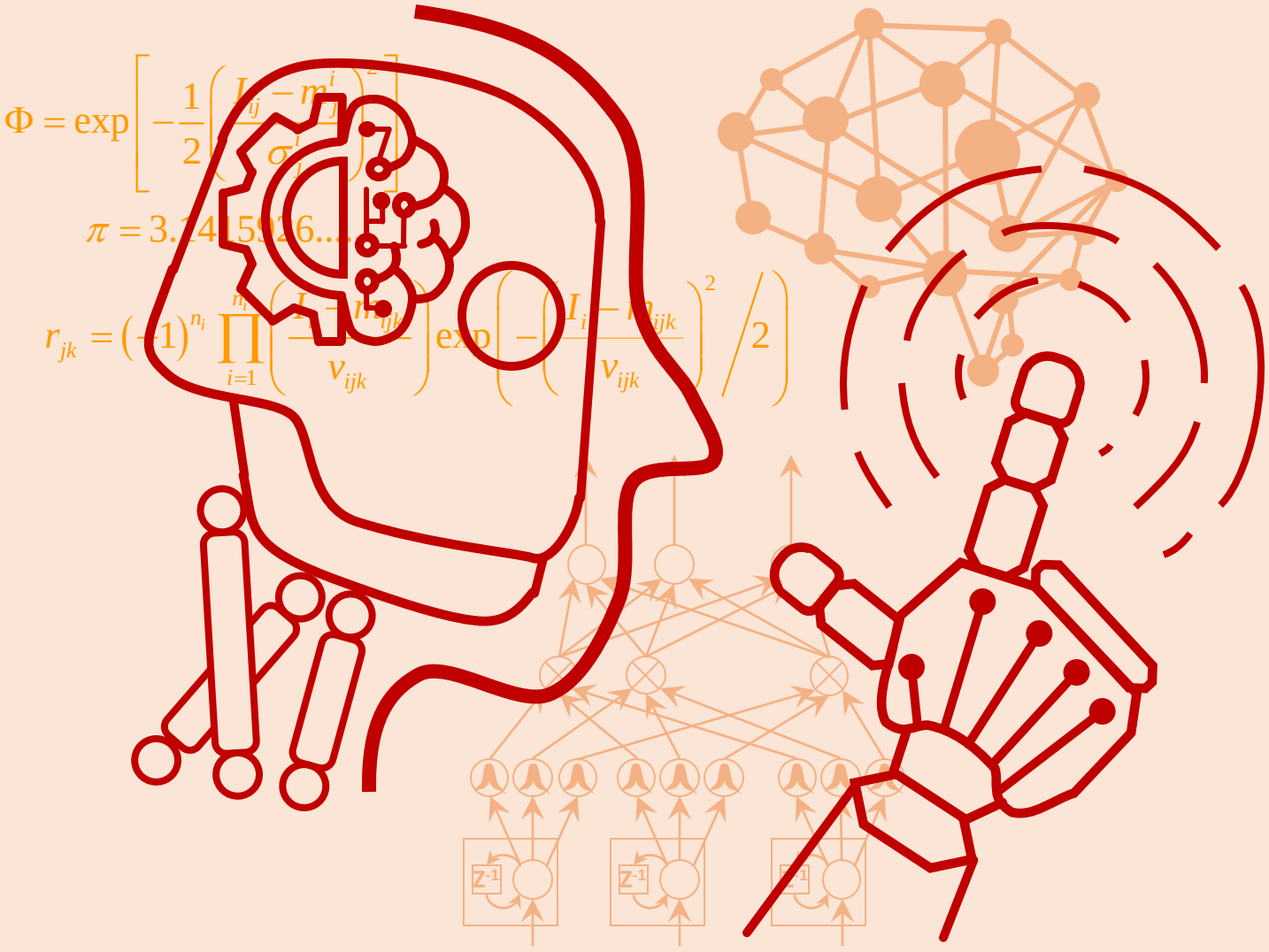
ISSN 2616-8170

Robotics Society of Taiwan

iRobotics

VOLUME 1
NUMBER 1
MARCH 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.

An Editorial Platform to Produce Screenplays for Robotic Puppet Shows

Kao-Shing Hwang, *Senior Member*, *IEEE*, Wei-Cheng Jiang, and Yu-Jen Chen

***Abstract*—In recent years, the field of robotics has seen steady growth. In addition to academic and industrial applications, robots are also being developed for entertainment, for robotic puppet shows. In these shows, robots are programmed to speak, to assume different postures, and exhibit human-like behavior. This study proposes an editorial Platform of Screenplays for Robotic Puppet Shows (EPRS). The EPRS allows users of all ages to compose screenplays and control robots quickly and intuitively. Game design concepts that allow for varied storylines are used. The use of timestamps allows the user greater control over timing. This is crucial to creating complex arrangements of motion and sound, which allows for more complex scenes. Individual scenes are then ordered to create ever more sophisticated storylines. The experiment results demonstrate that the platform is flexible and user-friendly.**

***Index Terms*—Entertainment robot; Robotic puppet shows; Timeline**

I. Introduction

Robots are rapidly replacing humans in many jobs that are too repetitive or too dangerous. The principles of robotics are being applied in academia and in industry. In addition to service-oriented robots that make life more convenient and work more efficient, entertainment-oriented robots have expanded opportunities for leisure. This study proposes an editorial platform of screenplays for robotic puppet shows (EPRS) that can be applied to multiple types of robots in entertainment-related fields, such as drama and theater. The intuitive graphic user interface can be operated by children easily. Drama and games are forms of entertainment that are different from the viewer's perspective. Drama is a passive form of entertainment but games allow players to interact directly with content. That is to say, drama is fully produced before it reaches its audience, as in the case of television programming and film. The audience can only watch as the story unfolds. However, in a game, the decisions of the user determine outcomes. The story could be said to be a product of decisions that are made by the user, real-time feedback from those decisions that are received by the user from the game and subsequent decisions that are made by the user based on that feedback. In short, playing games involves continuously making decisions. For example, in Massive Multiplayer Online Role-playing Game (MMORPG), each player's role is the same when they first start playing. As each individual player makes choices each character in the story encounters a distinctly different fate. These choices have a critical effect on the direction of the story. During the story's development there are also scripted events and triggered events [1]. Scripted events are previously arranged and cannot be changed by the choices that characters make. For example: weather conditions such as thunder, lightning or rain cannot be changed by any individual character. However, triggered events only occur if the character satisfies certain requirements and these affect the next part of the story. People make choices depending on their personality. Robotic puppet shows typically implement event-driven story arrangements [2][3] that are independent of time so only a limited level of complexity is possible [4]. This study uses time stamps to allow users to more precisely arrange performance elements. The EPRS allows users from different age groups to edit their own personal screenplays by accessing the website. By editing the screenplay, users can determine the robots' posture and speech, to make them seem more lifelike. The addition of sound effects and music also render the performance more realistic. In traditional drama, a screenplay's plot is fixed. This platform incorporates a feature called "branching", which allows the user to make real-time decisions that determine the direction of the plot. This renders the performance more intriguing and multi-dimensional.

This study comprises five parts. The first section explains the motivation for and objectives of the study. The second section describes the platform environment, the development tools and related works. The third section describes system analysis and design; this section also defines the motion generation, screenplay editing and script the performance and describes the hardware and software architectures that are used in the platform. Section 4 demonstrates the implementation of the platform and compares the results with those for previous works. Section 5 summarizes this study and makes some recommendations for future work.

Kao-Shing Hwang is with the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan. He is also an adjunct professor of the Department of Healthcare Administration and Medical Informatic, Kaohsiung Medical University, Taiwan (e-mail:hwang@ccu.edu.tw).

Wei-Cheng Jiang is with the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (e-mail:enjoysea0605@gmail.com).

Yu-Jen Chen is with the Department of Electrical Engineering, National Chung Cheng University, Chiayi 62102, Taiwan (e-mail:yujenchn@gmail.com).

II. Background

A. MVC Architecture

MVC is a software architectural pattern that divides an application into three parts: the model, the view and the controller [5][6]. This pattern is used to simplify code modification, extensibility and reutilization. The model uses business logic and interface data exchange. Of the three components, only the model has direct access to the database. The view develops data, which is displayed as visual outputs, such as webpages. The controller takes input data from the view and translates it into commands, in order to update the status of the model or to change the appearance of the associated view.

B. User Interface Framework

LAMP is a software bundle that is used to build web servers. The acronym is derived from the names, Linux, Apache, MySQL and PHP. LAMP is a common web platform because it is a free, stable and offers an open source software package. JavaScript is usually embedded with an HTML document. It is recognized as client scripts that are directly interpreted by the web browser rather than sending data back to the web server. It allows user interaction, such as reaction to an event, data validation and the reading and writing of HTML elements. JQuery is a JavaScript library that selects all DOM elements or the partial parts that use a designated ID or CSS, in order to bind various operations together, such as event triggers, animations, effects, AJAX and extensibility through plug-ins. In this platform the Kendo UI, a jQuery-based framework includes many UI widgets and data visuals that render the web interface more attractive and user-friendly.

C. Motion Capture

Developers aim to make the robotic motion in robotic puppet shows more lifelike. Motion programming is usually accomplished by adjusting the parameters for the robot's actuators individually, which is time-consuming and labor-intensive. Motion sensing devices that use optics such as Microsoft Kinect [7] provide developers with an efficient, cost-effective method to portray humanlike motion.

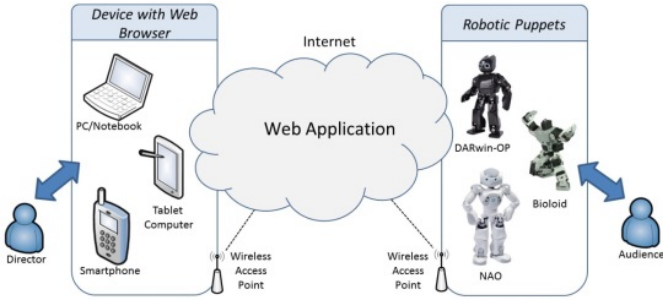


Fig. 1. The concept of the EPRS architecture.

III. System Analysis And Design

A. System Introduction

In general, a controller controls a single type of robot. Therefore, controlling different types of robots at the same time is a complicated and high-cost task. This study implements a platform that addresses this problem and uses the platform to develop a robotic puppet show. The core design principle allows users of different ages to make different types of robots assume different postures and perform a broader range of actions by editing a screenplay.

Web services have become integral to human interaction with computing systems. Developing a web component reduces the need for storage space on the client, so the system is freer from restrictions on time and distance. Wi-Fi allows high transport speeds and is in widespread use. A robotic puppet can feature a built-in Wi-Fi function or can have a Wi-Fi module installed. Therefore, to allow users to access the website and control robotic puppets, Wi-Fi is the main communication type for this system. The system design is shown in Fig. 1.

The process of performing a screenplay includes three methods to generate motions for controlling robots: actuator parameter adjustment preprogramed robotic motions and Kinect-based motions. Actuator parameter adjustment [8][9] is commonly used to control a robot's joints. In this study, preprogramed robotic motions and Kinect-based motions are used for robotic puppet shows. The preprogramed robotic motions are developed and stored within the robotic puppets, for future use. The platform also allows users to generate new motions by using Kinect sensing devices. These two motions

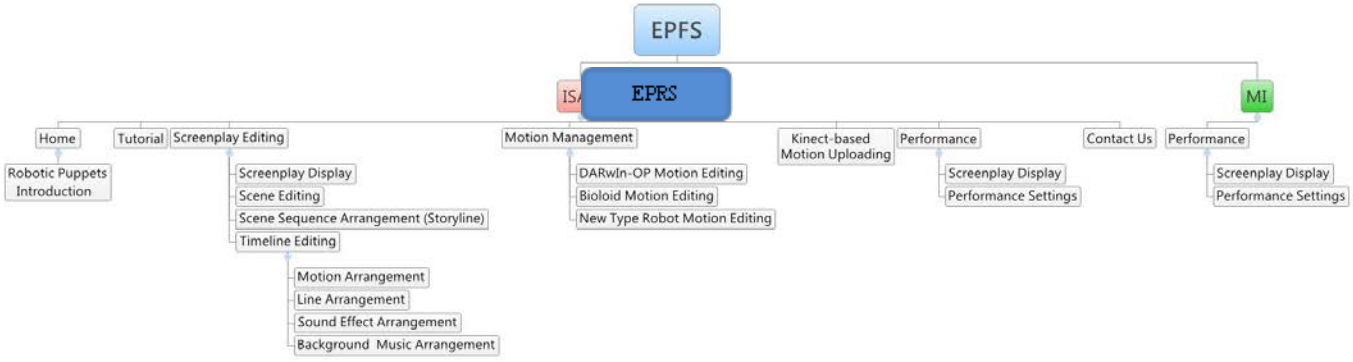


Fig. 2. The entire EPRS map.

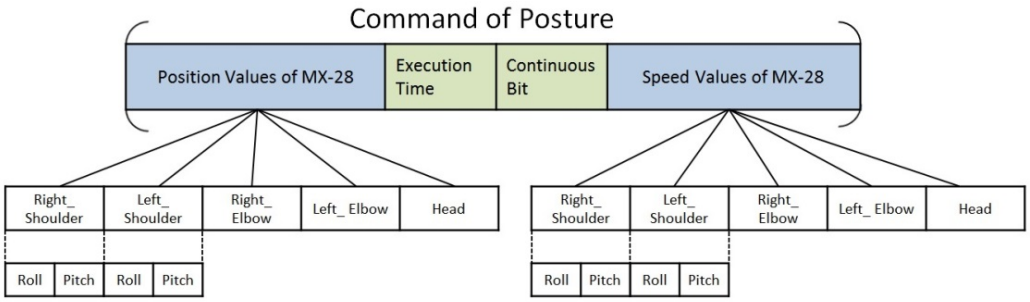


Fig. 3. Posture within the Kinect-based motion file.

are then incorporated into the motion library and arranged within the timeline of each scene. The scenes are then organized and become the content of the screenplay. Screenplays can be performed and shared by any user who has signed up for the platform. The entire map of the EPRS is seen in Fig. 2. This platform contains two different parts: the interactive screenplay authoring platform (ISAP) and the mobile interpreter (MI). The ISAP provides users with six different function webpages: home, tutorial, screenplay editing, motion management, Kinect-based motion uploading, screenplay performance, and contact. The MI only provides the screenplay performance function.

B. Motion Generation

- Preprogramed Robotic Motion

Robotic puppets such as DARwIn-OP and NAO allow user to rapidly develop a variety of postures and motions that are then stored within the robot and can be used for robotic puppet shows. For DARwIn-OP robots, the official software, which is called “RoboPlus motion”, allows users to edit the continuous motion of a robot and this is then saved as a page in a fast and simple way. Each page contains the information for one single motion, including the duration for each step, the speed and value for the actuators and the number of repetitions that are needed to complete a full motion. Choregraphe is software that was developed for NAO robots. It allows users to control the robot by dragging or customizing a variety of event boxes within the GUI. It also allows the developer to easily observe the angles between each joint in the NAO so that the motion can be programed as a file in Python and recorded within the robot.

When these motion pages or files are created and stored within the robots, the director then uses the motion management page in ISAP to add these motions into the platform and then use them for robotic puppet shows.

- Uploading Kinect-based Motion

This platform allows the user to make use of Kinect to capture a demonstrated motion and add it as an element in the motion library, so motion generation is more intuitive and efficient and a greater range of motion becomes available. Detailed below are the methods proposed in [7] to emulate and represent motion using Kinect. When Kinect successfully captures a human image, a set of line segments connects the nodes on a skeleton-like image that is superimposed on the silhouette of the user’s body. The program then collects important data, such as joint position and orientation. Inverse kinematics involves the use of kinematics equations to determine the parameters for joints, in order to achieve the desired posture and motion, so the change in angle for each joint can be calculated. According to the formula proposed in [7], actual rotation angles can be calculated for each actuator. In order to calculate how to reach a desired position, the change in the angles must be translated into units. When the movements are recorded they are automatically saved in the system for future use. Large quantities of data can be saved because data is rapidly generated during the movement capture process. Therefore, key-poses are separated from redundant postures that occupy resources. When a new pose is captured, the time is recorded and the difference between the time for the two key-poses is calculated, to obtain the execution time. This time is then stored in the time_buffer and is used to calculate the speed of the actuator. The continuous_bit determines whether an actuator is in continuous motion. If the key pose is in continuous motion then the continuous_bit recognizes it as a one, so while the actuator is moving to the continuous key-pose, the subsequent pose begins to load. The continuous_bit allows a smoother representation of the key-poses. The packet that is constructed using these methods contains the actuator’s position, speed, continuous_bit and execution time from the key-pose, as shown in Fig. 3. These packets are used as postures within Kinect-based motion files to represent for robotic motion.

C. Screenplay Editing

- The Scene Tree

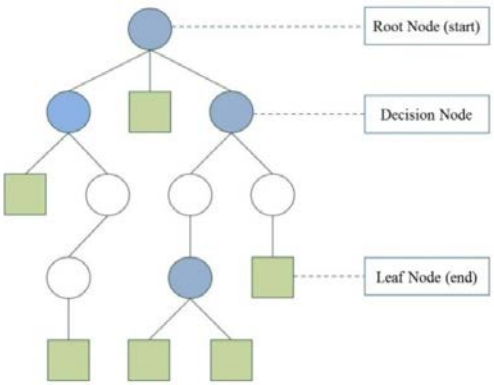


Fig. 4. The structure of the scene tree.

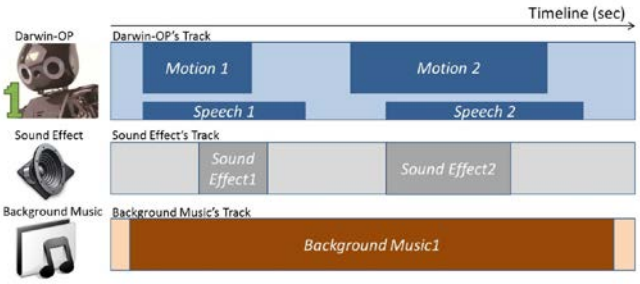


Fig. 5. Timeline tracks within a scene.

A rooted n-ary tree structure is used to construct an entire screenplay. This type of tree allows users to choose the direction of the plot. There are three key types of nodes in the tree: the root node, the decision node, and the leaf node. Each node represents a single scene, as shown in Fig. 4. The root node is the beginning of the story. A node that is connected to two or more child nodes is referred to as a decision node. Branching occurs when the first node has finished performing and the user chooses a scene at the next level of the tree, in effect selecting the direction of the plot. The leaf node is simply the final scene of the screenplay. Using branching allows the same screenplay to yield different plots according to user preferences.

- The Time Stamp

Compared to event-driven story arrangements, the time stamp feature provides the director with a precise method to arrange the content of the performance, so the robotic puppet show is more lively and attractive. The feature can be applied to two parts. Firstly, motion and speech are used to express the actor's emotions in a performance. Therefore, the fact that the motion and speech for each robot starts at the same time or overlaps, in order to fit the director's requirements, plays an essential part in the scene editing process. Secondly, similarly to a real drama, a robotic puppet shows has a cast of actors that perform on stage. The task of arranging and recording the interaction between puppets requires sophisticated controls. EPRS incorporates sound effects and background music to increase dramatic tension. In allow this expression and interaction, the timeline (which is measured in seconds) is built within each scene as shown in Fig. 5. Each puppet, piece of music and sound effect has its own track within the timeline that records details of the performance, such as motion, speech, sound effects and background music. The robotic puppet track has two sub tracks that individually record the content for motion and speech. GUI makes the colored boxes that represent the different performance details.

- The Screenplay Format

The JSON (JavaScript Object Notation), which is a subset of JavaScript syntax, is a text format that is mainly used to exchange data [10]. It is can be understood and used by computers and people so JSON is chosen as the format for the screenplay for the ISAP. When the user selects and performs a screenplay using a web browser, the JSON data, which records the details of this screenplay, is automatically generated from the relational database. The JSON data is formatted as shown in Fig. 6.

D. Performance of the Screenplay

As shown in Fig. 7, an editorial platform to produce screenplays for robotic puppet shows comprises the ISAP and a MI.

- The Interactive Screenplay Authoring Platform (ISAP)

The ISAP features two components: the website and Node.js. The primary purpose of the website is to provide users with ways to display, edit, and store screenplays. The design of the website uses the MVC architectural pattern. In a Wi-Fi environment, anyone can use a web browser to access the platform and edit screenplays. All of the details are automatically stored on the server database.

A virtual IP (VIP) is assigned to the device with the web browser and robotic puppets when they are connected to the local network via Wi-Fi. In order to deliver the packet for the screenplay to the robotic puppets that support VIP, the lite-server, Node.js, is inserted between the web browser and the robotic puppets, as shown in Fig. 8. Node.js is a server-side scripting language that uses JavaScript. It is

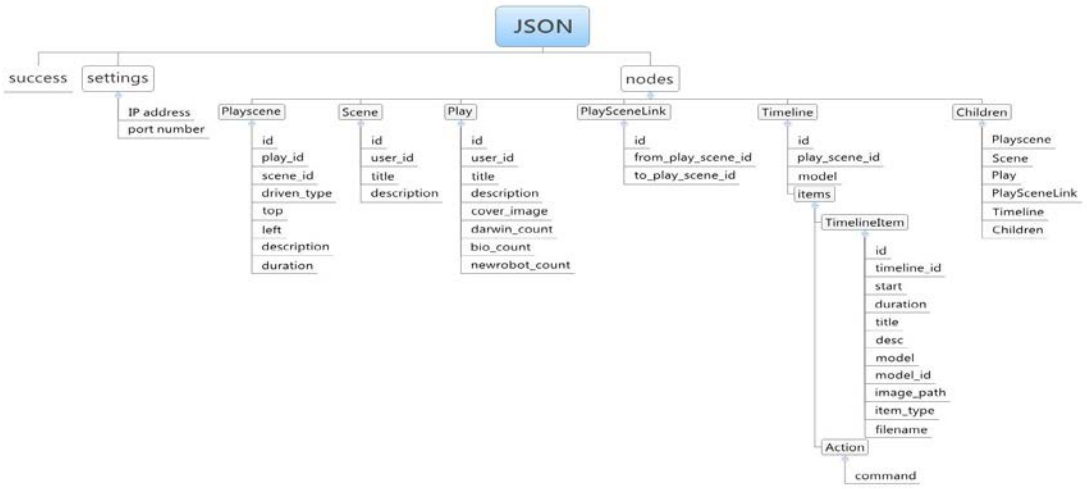


Fig. 6. Screenplay formatted using JSON.

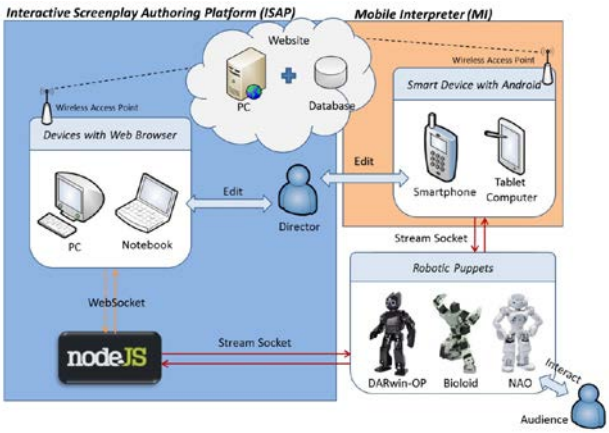


Fig. 7. The EPRS architecture.

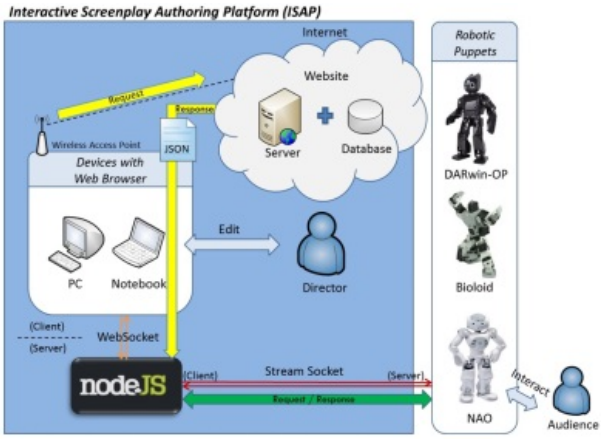


Fig. 8. The process for performing a screenplay.

allows network applications to be easily developed [10]. Using Node.js in the ISAP, the two methods that are used for communication are WebSocket [11] and Stream Socket [12].

Once established, the WebSocket data frames can be sent back and forth between the client (web browser) and the server (Node.js) in the ISAP. On the other side, stream sockets are created to implement a bidirectional channel between the client (Node.js) and the servers (robotic puppets). Firstly, the Node.js application is run as a Websocket server in the background, to start listen to the client-side web browser. When the director selects a screenplay and enters the settings page using a web browser, the WebSocket is connected. When all of the settings, including the IP addresses and port numbers, have been submitted by the director, the browser sends a request to the web server to fetch the designated screenplay that was formatted using JSON, as shown in Fig. 8. The JSON file is then delivered to the Node.js server via WebSocket and interpreted, level by level, to read the settings and nodes objects within this screenplay. In this step, Node.js plays not only in the WebSocket server, but also in the stream socket client, in order to connect to each robotic puppet using the specific IP addresses and port numbers that are recorded in the JSON file.

Fig. 9 shows that when the stream sockets is created, the Node.js selects the root scene and all of commands that belong to the timeline item are executed using a designated

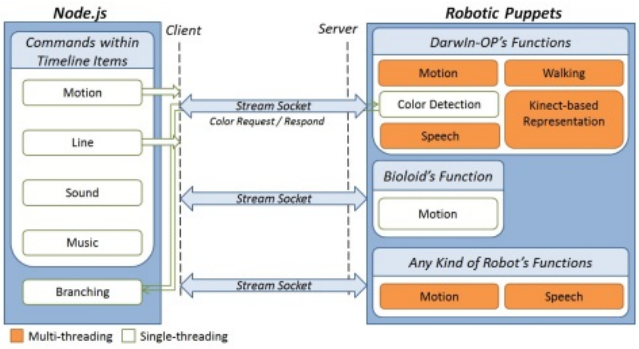


Fig. 9. The execution of a command by Node.js using stream sockets.

delay time. If the commands belong to the robotic puppets, they are taken into the socket array and sent to each robotic puppet through the stream socket. The sound effects and music commands are played by calling the external media software. When the current scene finishes, Node.js checks whether the scene has children. If it does, the children's scene is performed. Scenes that have more than two children result in branching. In this case, Node.js sends a request to the DARwIn-OP robot to run the color-detection function. When the user chooses the color, the result is sent back to Node.js to select and perform the children scene with the corresponding color. In order to translate different commands to specific robotic puppets through stream sockets, EPRS has three different types of packets. When the robots receive the packets, the command within each packet is extracted and executed by the relevant functions.

- The Mobile Interpreter (MI)

The Mobile Interpreter (MI) is an Android application that works within the existing MVC framework. MI gives additional view, which obviates the need to redesign the model or the controller. MI only plays and displays screenplays that are already in the system. The advantage of this is that the end user no longer needs to access a browser and install Node.js, as shown in Fig. 10. A user-friendly interface that combined the portability of a tablet computer allows children to easily access this type of educational entainment tool.

E. Hardware Architecture

1) DARwIn-OP

DARwIn-OP is equipped with twenty Dynamixel MX-28 servo actuators, a USB camera, a microphone, a speaker and multiple I/O devices that include HDMI and USB ports [13]. The DARwIn-OP operates in real time and comes pre-installed with the Ubuntu 9.04 operating system.

2) Bioloid

The Bioloid consists of eighteen Dynamixel AX-12 servo actuators and unfixed units. The name, Bioloid, means that all of the main mechanisms of biological entities can be

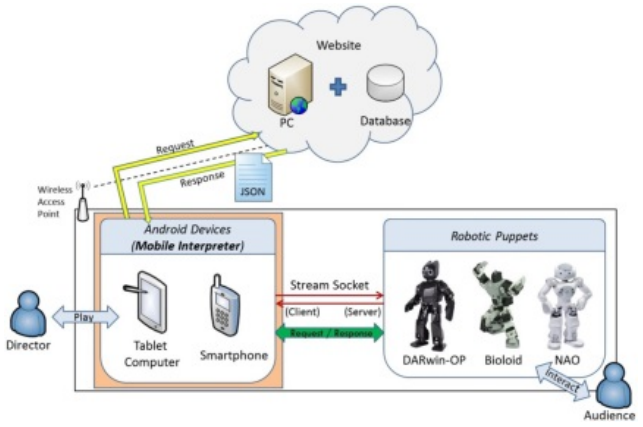


Fig. 10. MI architecture.

separated and reassembled. This feature allows users to design their own robots that can adapt to a variety of different environments.

3) The NAO

The NAO is a highly efficient, programmable humanoid robot that has 25 degrees of freedom, and a variety of visual, audio, language, movement coordination and tactile sensors [13]. These sensors and the efficient design of the robot's body allow developers to implement creative, powerful and reliable applications.

4) The Smart Device

The MI allows users of all ages to play a screenplay for robotic puppets. Therefore, a large panel provides users with a better experience while reviewing the list of screenplays and operating the robotic puppet show. The device must also support Wi-Fi, in order to create the stream sockets with robotic puppets. An mp3 encoder is also required, to play the music for the sound effects and background. For these reasons, the ASUS MEMO Pad FHD 10 is used as the mobile controller for the robots.

F. Software Architecture

1) The DARwIn-OP API

The functions of the DARwIn-OP robot have two modes: multi-threading and single-threading. The muliti-threading mode includes motion, walking, Kinect-based representation and speech functions. Color detection is developed in the single-threading mode. In the multi-threading mode, the Pthread library allows four functions to be run simultaneously. These threads release their resources and destroy themselves when the processes are completed.

The color detection function is developed in the single-threading mode, so that the color detection results can be sent back to the client. The specifications of these functions are given herewith.

2) Basic Function

The basic functions within the DARwIn-OP can be divided into three parts: motion, walking and speech functions. When the DARwIn-OP receives a type I packet, a portion of the motion page or line is extracted using the identifier and executed using its related functions. For example, if a packet is written as "A001", the robotic puppet must run the motion function in accordance with motion page (001). If the page number does not belong to a motion page, it is translated into the units that are used to set the step values along the x axis for the walking function. The "Chello" packet tells DARwIn-OP to run the speech function, in order to say the line, hello.

3) Kinect-based Representation

Type II packets provide the DARwIn-OP with sufficient information to perform the Kinect-based representation. When the DARwIn-OP receives type II packets, it reserves enough memory space to temporarily store the subsequent posture command by using the information that was received from the first packet. Subsequent postures that contain the actuator's position, speed, continuous_bit and execution time are then used in the representation, so the key-poses that are captured by Kinect can be applied to the robotic puppet show.

4) The Color Detection Function

When the request from Node.js is received, the DARwIn-OP executes the color detection function. In [14][15], the one of the main properties of color is used to implement the color detection function. During the process of color detection, DARwIn-OP captures an image with a camera. Each element in the given array represents a single pixel. The function uses the degrees of hue to calculate the similarity between each single pixel's hue and the target color,with a specific tolerance. If more than sixty percent of the pixels are found to be the same as the target color, the color card that is seen by DARwIn-OP can pinpoint the pixels that are the same color as the target color.

5) The NAO API

Since the NAO is a new type actor for the EPRS, it is supported by type I packets when it is a stream socket client. The command that is stored in the packet is extracted using the identifier, in order to execute a python motion file or to run a speech function with a line. Both are executed using multi-threading so that two different emotions can be expressed at the same time.

6) The Tablet Computer

Android is an operating system that uses Linux. It is mainly used for mobile devices, such as smartphones and tablets. Unlike normal programming for an Android application in JAVA language, the Titanium Studio IDE provides exclusive APIs to develop cross-platform applications by only using JavaScript, HTML and CSS. The related Software Development Kit (SDK) must be installed at first. Titanium

Fig. 11. The interface that is used to create screenplay.

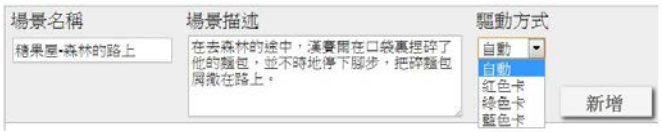


Fig. 12. Adding a new scene.

Fig. 13. The appearance of the scene box.

packs the mobile web application into a native application that can be run on Android, iOS or Blackberry OS.

IV. Implementation

This study uses a rational database that provides users with a screenplay editing service and the ability to record essential related data.

A. The User Interface for the ISAP

- Creating a Screenplay

Users can add personal screenplays by entering the title and description and by choosing the number of each type of robot to be used and a picture that becomes the cover for the screenplay. The interface that is used to create the screenplay is shown in Fig. 11. ISAP allows the user to add all of the scenes that are required for the screenplay, including the initial, the middle and the final scenes. As shown in Fig.12, when adding a new scene, the user first enters the scene's title and description and then, in the scene selector, chooses between one of two modes: automatic mode and colored card mode. In automatic mode, the scene is played in a linear fashion, as defined in the root node. That is, the story follows a pre-set pattern. The colored card mode allows the user to choose a colored card that is then interpreted by the DARwIn-OP as the selection of a plot direction. As shown in Fig. 13, a scene box appears to show that the new scene has been successfully created. There is an automatic mode or

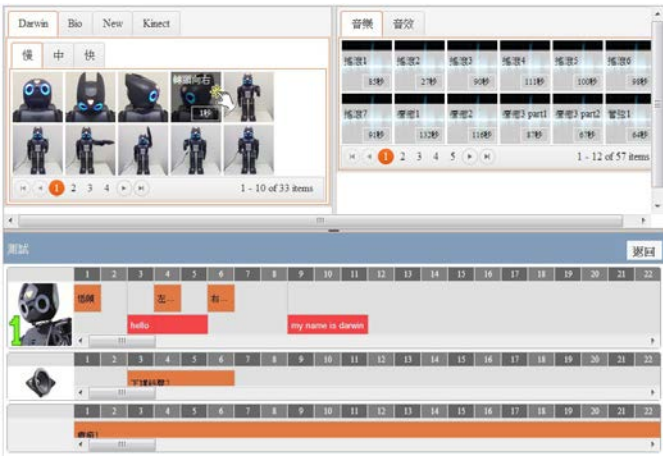


Fig. 14. Scene-editing page.

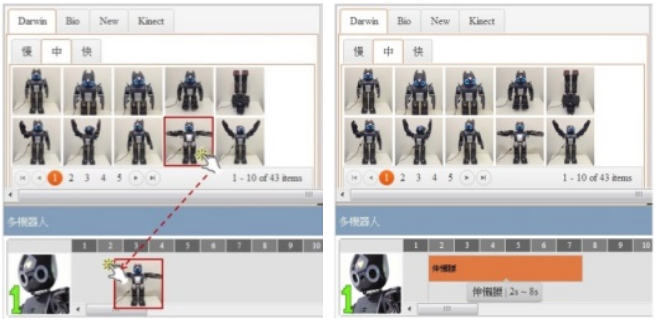


Fig. 15. Adding a new motion into a track.

Fig. 16. Speech function.

color-card mode icon at the top left corner. The tooltip icon at the bottom of the scene box allows the user to see a description of the scene.

- Editing a Scene

Having selected a scene, the user then double-clicks on the scene icon to edit the scene's content. The scene-editing page comprises three divisions: the motion library, the music library and the timeline, as shown in Fig. 14. The motion library allows the user to choose from a variety of postures and motions, ranging from pre-programmed robotic motions to Kinect motions. The music library features a diverse collection of sound effects and background music. In Fig. 15, the timeline consists of the sound effects, the background music and the motion and speech for each robot. Each scene has only one local timeline that is measured in seconds. The files in the motion and music library can be dragged into a specific track, in order to add a new event to the timeline. Fig. 16 shows the user picking up a motion file and dragging it into a track. In the next step the motion file is placed at the desired location on the timeline. Finally, the new action

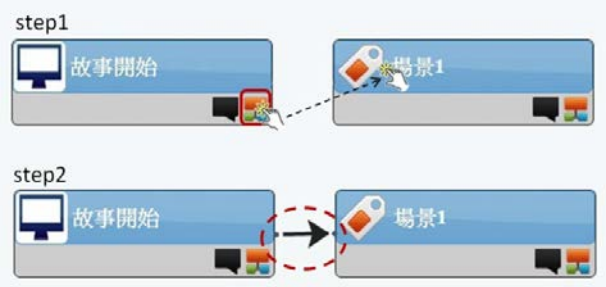


Fig. 17. Linking tool for a scene.

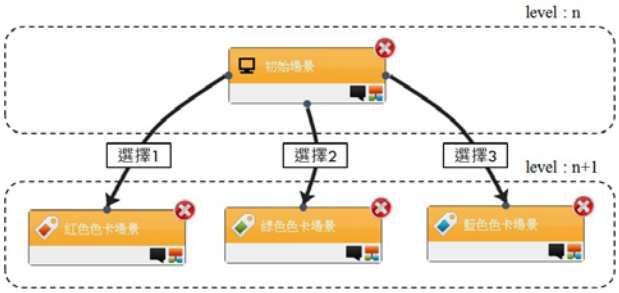


Fig. 18. Branching.

appears with its name on the timeline, along with a box that represents the start and end times for the action. The user can cancel the action by double clicking on the box that represents the action on the timeline. In Fig. 16, if the user double clicks on the blank area below the numbers and the timeline a box appears where the user can type what the robot are to say. This box can be deleted only by erasing this line.

- The Relationships between Scenes

When scenes have been created, a linking tool is used to establish the relationships between scenes. As shown in Fig. 17, the user drags the linking tools from the source scene to the target scene. An arrow automatically appears between the two boxes that represent the scenes, signalling completion of the linking process. The link is removed by double-clicking. Fig. 18 shows a scene box that has more than two child nodes at the next level, which result in branching at the end of the current scene.

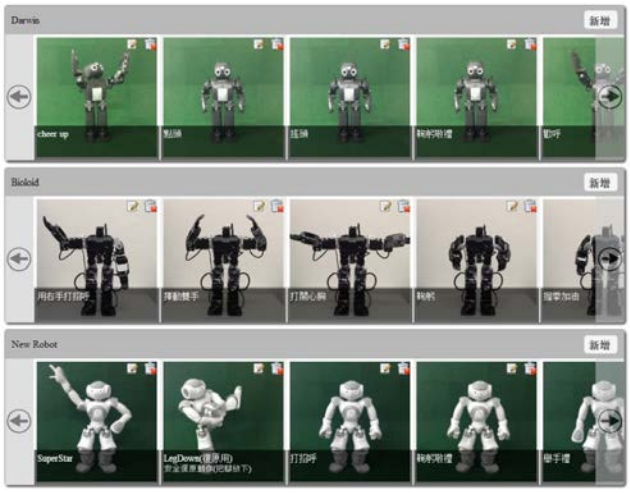


Fig. 19. Motion management page.

TABLE I
COMPARISON BETWEEN TWO PLATFORMS I, II AND EPRS

Platform / Function	Proposed platform I	Proposed platform II	EPRS
Type of Robots	Bioloid	DARwIn-OP NAO	Any kind
Vocal Expression	Yes	Yes	Yes
Web-based Authoring	Yes	Yes	Yes
Driven Type	Timeline	Event	Timeline
Expression in Motion and Speech.	Yes	Yes	Yes
Interaction between Robots, Sound Effects and Background Music	Yes	No	Yes
Effective Atmosphere	Facial Expression	No	Sound effects Background Music
Dynamic Interaction	No	No	Yes
Motion Extension	Only Posture	No	Yes

- Motion Management

Fig. 19 shows that the EPRS allows the administrator of the website to add or modify existing DARwIn-OP and Bioloid motion commands, as well as different types of performance robots with motion functions, such as the NAO. GIF and JPEG file formats are both supported. GIF is used for motion and JPEG for posture.

B. Improvement

There are six essential elements for improvement in the EPRS shown below. These are compared to platform I [8][9] and platform II [2][3]. Table I shows the comparison between the results for a previous work and those for the EPRS. The EPRS supports any type of robot that has built-in motion and speech functions. Users can upload or modify a robot's motion commands using this platform. The visual timeline tracks within each scene allow users to edit the details of screenplays more accurately and simply. The addition of background music and sound effects adds to the entertainment value of the platform and enhances the robotic puppets' performance and increases viewing pleasure. In the EPRS, the dynamic screenplay divides the story into many scenes using a flexible timeline. The order of these scenes can be easily rearranged using GUI. The branching feature allows users to influence the development of the story. Even if the screenplay is performed many times, different plot are possible. The EPRS allows the intuitive and simple addition of new motions using Kinect. Different Kinect based files can also be shared between users. As long as the Android devices and the robotic actors are connected to Wi-Fi, robotic puppet shows can be performed anytime. The video recording of the

platform showing the final results of the implementation can be accessed at [16].

V. Conclusion

This study proposes a streamlined graphical user interface for the editing of motion and sound-enabled screenplays. These dynamic screenplays allow the audience to interact with robotic puppets and create participatory experiences. The ability to add robot types and user-generated motion makes the platform a powerful tool that has creative and educational applications. Web-based architecture enables the EPRS to be accessed from any web browser or smart device that is connected to Wi-Fi, which eliminates the traditional physical restrictions and allows robotic puppet shows to be presented at any time.

In spite of the fact that the EPRS has many useful functions, there is still room for improvement or modification in the future. Three possible improvements are described below.

- Diverse Ways for Users to Interact with Robots

In order to ease interaction, the color detection could be substituted by facial or voice recognition, to allow more interesting and significant scenario to be created when the audience interacts with robotic puppets.

- More Performance Element

Video could also be used for the robotic puppet show, to make the performance more attractive and complete. An additional timeline track could be added within a scene to record the video arrangement and the designated videos could be synchronized with the robotic puppets, the sound effects and the music. The video could also be played on a monitor or on the panel of a smart device behind the robotic puppets.

- Key-pose Reduction

There are still some redundant postures that are regarded as key-poses and recorded in the files. If the redundant postures could be filtered out, there could be a resultant reduction in the use of system resources and storage space.

References

[1] A. Rollings and E. Adams, *Andrew Rollings and Ernest Adams on game design*, USA: New Riders, 2003.
[2] J. J. Siao, "The Graphic Authoring Platform of Screenplays for Robotic Puppet Shows," M.S. thesis, E. E. Dept., National Sun Yat-sen Univ., Kaohsiung, Taiwan 2012.
[3] M. C. Wu, "The Development of Screenplay Interpreter for Multi-morphic Robots," M.S. thesis, E. E. Dept., National Sun Yat-sen Univ., Kaohsiung, Taiwan 2012.
[4] K. S. Hwang, Y. C. Kuo, W. C. Jiang, "An Editorial Platform for Screenplay of Interactive Robotic Puppet Show," *In: International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines* (*CLAWAR 2015*), 2015.
[5] S. Singh and J. Iyer, "Comparative Study of MVC (Model View Controller) Architecture with respect to Struts Framework and PHP," *International Journal of Computer Science Engineering*, vol. 5, no. 3, pp. 142-150, 2016.
[6] C. Supaartagorn, "PHP framework for database management based on MVC pattern," *International Journal of Computer Science and Information Technology*, vol. 3, no. 2, pp. 251-258, 2011.
[7] Y. A. Zheng, "Humanoid Robot Behavior Emulation and Representation," M.S. thesis, E. E. Dept., National Sun Yat-sen Univ., Kaohsiung, Taiwan 2012.
[8] J. L. Lin, and K. S. Hwang, "A Collaborative Authoring Workspace and Script-Based Control Platform for Heterogeneous Robots," *AUSMT*, vol. 5, no. 3, pp. 163-171, 2015.
[9] C. A. Chen, "The Authoring tool and Performance Platform for Robotic Puppets Show," M.S. thesis, E. E. Dept., National Chung Cheng Univ., Kaohsiung, Taiwan 2011.
[10] M. Madsen, F. Tip, and O. Lhoták, "Static analysis of event-driven node.js JavaScript applications," *In Proc. ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications* (*OOPSLA*), 2015.
[11] R. Park, H. Hwang, J. Yun, I. Moon, "Study of HTML5 WebSocket for a Multimedia Communication," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 7, pp. 61-72, 2014.
[12] S. Damianakis, C. Dubnicki, and E.W. Felten, "Stream Sockets on SHRIMP," Tech. Report TR-513-96, Dept. of Computer Science, Princeton Univ., Princeton, N.J., 1996.
[13] I. Ha, Y. Tamura and H. Asama, "Development of Open Humanoid Platform DARwIn-OP," *SICE Annual Conference*, pp. 13-18, 2011.
[14] B. R. Conway and D. Y. Tsao, "Color-tuned neurons are spatially clustered according to color preference within alert macaque posterior inferior temporal cortex," *PNAS*, vol. 106, no. 42, pp. 18 034-18 039, 2009.
[15] R. Veale, P. Schermerhorn and M. Scheutz, "Temporal, Environmental, and Social Constraints of Word-Referent Learning in Young Infants: A Neurorobotic Model of Multimodal Habituation," *IEEE Transactions on Autonomous Mental Development*, vol. 3, no. 2, pp. 129-145, 2011.
[16] IRIS. (February 27, 2014). An Editorial Platform for Screenplay of Interactive Robotic Puppet Shows. IRIS Lab. National Sun Yat-sen University, Kaohsiung, Taiwan. [Online]. Available: https://www.youtube.com/watch?v=NpwmK9MMGys, accessed Nov. 20, 2014.

Kao-Shing Hwang (M'93–SM'09) is a professor of Electrical Engineering Department at National Sun Yat-sen University, and an adjunct professor of the department of Healthcare Administration and Medical Informatic, Kaohsiung Medical University, Taiwan. He received the M.M.E. and Ph.D. degrees in Electrical and Computer Engineering from Northwestern University, Evanston, IL, U.S.A., in 1989 and 1993, respectively. He had been with National Chung Cheng University in Taiwan from 1993-2011. He was the deputy director of Computer Center (1998-1999), the chairman of the Electrical Engineering Department (2003~2006), and the director of the Opti-mechatronics Institute of the university (2010~2011) in National Chung Cheng University. He has been a member of IEEE since 1993 and a Fellow of the Institution of Engineering and Technology (FIET). His research interest includes methodologies and analysis for various intelligent robot systems, machine learning, embedded system design, and ASIC design for robotic applications.

Wei-Cheng Jiang received the B.S. degree in Computer Science and information Engineering and M.S. degree in Electro-Optical and Materials Science from National Formosa University, Yunlin, Taiwan, in 2007 and 2009, respectively. He received the Ph.D. degree in Electric Engineering from National Chung Cheng University, Chiayi, Taiwan, in 2013. He was a posdoc with the Electrical Engineering Department at National Sun Yat-sen University, Kaohsiung, Taiwan. He is currently a visiting scholar in the Department of Electrical and Systems Engineering, Washington University in St. Louis, Saint Louis, MO, USA. His research interests include machine learning, neural networks, and intelligent control.

Yu-Jen Chen received the B.S. degree in electrical engineering from the Tatung Institute of Technology, Taipei, Taiwan, in 1994, and the M.S. and Ph.D. degrees in electrical engineering from National Chung Cheng University, Chia-Yi, Taiwan, in 1997 and 2009, respectively. From 2004 to 2009, he was an Adjunct Lecturer with the Center for General Education, National Chung Cheng University. Since 2010, he has been an Assistant Professor with the Electrical Engineering Department, National Chung Cheng University.

His current research interests include machine learning, robotics, neural networks, and embedded systems.

Collision-Free Guidance for a Passive Robot Walking Helper

Yi-Hung Hsieh[1], Yu-Hsuan Su[1], Chih-Chung Chiu[1], Kuu-Young Young[1], and Chun-Hsu Ko[2]

***Abstract*—In facing the coming of the aging society, various types of robots have been introduced for elderly assistance. Among them, the passive robot walking helper, which features continuous energy dissipation from the system and is thus intrinsically safe, is appealing in providing safe mobility. For its practical application in daily activities, the capability in obstacle avoidance during guidance is very imperative. As both static and moving obstacles may be present, it motivates us to propose such a scheme that is able to avoid both types of obstacle in a real-time manner. With a desired goal to reach, the proposed scheme first plans a smooth path for the walking helper to follow. A novel strategy, combined with receding horizon control, is then proposed to avoid the obstacles detected by the equipped sensing system. Experiments based on i-Go, a passive robot walking helper developed in our laboratory, are conducted in real environments for demonstration.**

***Index Terms*—collision-free guidance, static and moving obstacle, passive robot walking helper.**

I. Introduction

With the increase in elderly population, elder care is a focus within our society [1]. Many different types of robots have been introduced to assist the elderly in various areas. The elders tend to fall during walking due to the inability to balance. In addition to the development of electric wheelchairs [2-3], research has also been proposed to use a robot walking helper for their assistance. Even when the elders are able to walk, frequent use of a wheelchair may lead to atrophy in lower-limb muscles [4]. It thus makes the robot walking helper more appealing. According to the source of power, they can be classified into active and passive types. The active robot walking helpers use servo motors to guide the user, while actively adding energy to the system [5-11]. The passive ones move only by the forces supplied by the user. Controlled brakes are used to steer the robot walking helper while continuously dissipating energy from the system. With energy being continuously dissipated from the system, these are intrinsically safe [12-18]. For both types of robot walking helpers, effective maneuver and guidance strategies are crucial for their practicality in daily lives. Up to date, they have been equipped with functions such as path guidance [5,19-21], obstacle avoidance [14,22], health monitoring [11,19,23], preventive fall [23], and intentional manipulation [5,7,14-15], among others.

For safety concern, in this study, we dedicate to the passive robot walking helper. We thus need to face a very challenging demand in control, as its movement is dictated by both applied force from the user and braking torque from the walking helper. In other words, the governing guidance system needs to determine proper braking torque to go along with user-applied force for path following in a real-time manner. Furthermore, as both static and moving obstacles may be present in the daily activities, it is very important for the robot walking helper to be capable of obstacle avoidance during guidance. Although there have been many approaches proposed for the mobile robot [24,25] and also active robot walking helper [13,25] for obstacle avoidance. Comparatively, few researches have been devoted to the passive robot walking helper. Among them, Kosuge et al. propose potential canal for path planning and use the artificial potential field method for obstacle avoidance [27]. In our previous works, we have developed a scheme based on the concept of collision-free area for avoiding static obstacle [28] and conducted preliminary study on moving obstacle avoidance [29]. Up to now, a scheme for the passive robot walking helper that can tackle moving obstacles well is still in demand. It thus motivates us to develop such a scheme.

To tackle moving obstacles, especially under the condition of limited mobility provided by the walker via the brakes, we come up with a novel strategy to respond to the incoming obstacles of various speeds from different directions. According to detected obstacle information, the proposed strategy will determine whether the walker should continue the planned guidance, wait for the obstacle to pass, or steer the walker to deviate from it. We then propose using receding horizon control to execute those actions solicited by the strategy for obstacle avoidance. Receding horizon control is an online control method, which incorporates state and control constraints into a planning problem. It predicts system states and control inputs using the optimization method, and controls the system with the predictive control profile over a small period. The desired goal can thus be achieved by repetitively executing the procedure. Based on our previous work [30], in which receding horizon control was applied to

Yi-Hung Hsieh, Yu-Hsuan Su, Chih-Chung Chiu, Kuu-Young Young are with the Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan (corresponding author Kuu-young Young, e-mail: kyoung@mail.nctu.edu.tw).

Chun-Hsu Ko is with the Department of Electrical Engineering, I-Shou University, Kaohsiung, Taiwan (e-mail: chko@isu.edu.tw).

This work was supported in part by the Ministry of Science and Technology, Taiwan under grant MOST 102-2221-E-009-138-MY3 and 103-2221-E-214-049.

Part of the results has been presented in IEEE International Conference on Systems, Man, and Cybernetics, 2016.

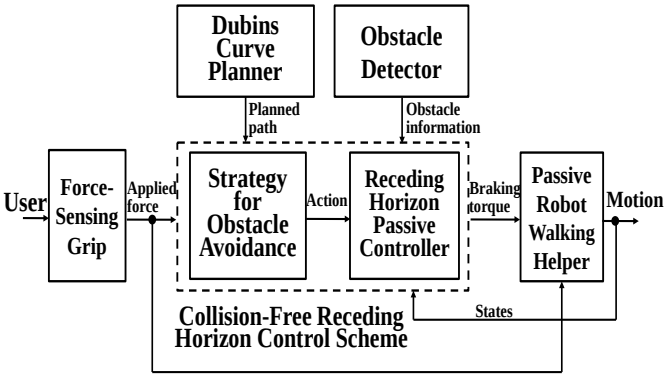


Figure 1 i-Go equipped with sensing and control modules.

govern a passive robot walking helper in a free space, we further enhance it for obstacle avoidance.

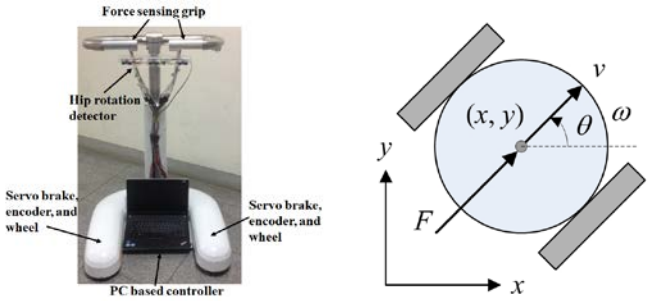
For demonstration, we conduct experiments based on a passive robot walking helper, i-Go, developed in our laboratory. The rest of this paper is organized as: Section II describes the system of a passive robot walking helper, including its mechanism, sensing, and control modules. Section III discusses the proposed scheme. Section IV presents simulation and experimental results, including the case with both static and moving obstacles. Finally, section V gives the conclusions.

II. System Description

In this section, we explain how the proposed scheme operates and also give a brief description on the passive robot walking helper, i-Go, developed in our laboratory. Figure 1 shows the system block diagram of i-Go, which is equipped with the sensing and control modules. In Figure 1, the proposed collision-free receding horizon control scheme receives the inputs of applied force from the user via the force-sensing grip, obstacle's position and velocity from the obstacle detector, and planned path from the Dubins curve planner [31], and also the feedback of the motion states of i-Go, to determine proper braking torque for collision-free guidance. This scheme mainly consists of the strategy for obstacle avoidance and receding horizon passive controller, which will be discussed in Sec. III. Before its discussion, we briefly introduce the system implementation and dynamic model for i-Go below.

A. System Description

Figure 2(a) shows the outlook of the i-Go, which consists of a support frame made of aluminum and designed to be U-shaped that allows the user to walk in its center for better support and stability, two wheels equipped with servo brakes and encoders for position and velocity estimation, the force sensing grip system on the handles, the hip rotation detector system, a controller installed in the notebook, and sensors for obstacle and slope detection. The laser sensor system (URG-04LX-UR01, Hokuyo Corp., Japan) and force-sensing grip previously developed [29] are used to detect obstacle information and user-applied force, respectively. The servo brakes are used to regulate the velocity and directional angle



(a) Outlook (b) Configuration

Figure 2 The developed passive robot walking helper, i-Go: (a) outlook and (b) configuration.

Figure 3 Laser range finder (URG-04LX-UR01) used for obstacle detection.

of i-Go. The force sensing grip and hip rotation detector are attached on i-Go, as shown in Figure 2(a).

B. Dynamic Model

Figure 2(b) shows the configuration of i-Go in Cartesian coordinates, described as

$$q = [x \quad y \quad \theta]^T \tag{1}$$

where x and y are the coordinates of the center of the rear axle, and θ the heading angle. With the assumption of no-slip at the wheel contact points, the velocities of the wheel centers are parallel to the heading direction, and $\dot{q}$ can then be expressed as

$$\dot{q} = S(q)V \tag{2}$$

where $V=[v\ \omega]^T$ is a vector consisting of the heading speed v and turning speed ω, and $S(q)$ is given as

$$S(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \tag{3}$$

Assume that this i-Go is symmetric and the user applies force on its center. The motion equations are thus formulated as

$$M\ddot{q} = E\tau_b + DF_d - C\lambda \tag{4}$$

where

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, E = \begin{bmatrix} \cos\theta/r & \cos\theta/r \\ \sin\theta/r & \sin\theta/r \\ b/r & -b/r \end{bmatrix}, \tau_b = \begin{bmatrix} \tau_{br} \\ \tau_{bl} \end{bmatrix}$$
$$D = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}, F_d = \begin{bmatrix} f_d \\ \tau_d \end{bmatrix}, C = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix} \tag{5}$$

with m the mass, I the moment of inertia about the center of the rear axle, r the wheel radius, b half distance between the two wheels, br and bl the servo torques applied to the right and left wheels, respectively, f_d and τ_d the driving force and torque imposed on the robot walking helper, and λ the constraint force. For the robot walking helper to be passive and dissipative, we design the control law to be

$$\tau_{br} = -K_r\dot{\theta}_r,\ \tau_{bl} = -K_l\dot{\theta}_l \tag{6}$$

where $\dot{\theta}_r$ and $\dot{\theta}_l$ are the angular speeds of the right and left wheels, respectively, and K_r and K_l the non-negative braking gains. And, the angular speeds $\dot{\theta}_r$ and $\dot{\theta}_l$ can be calculated under the no-slip condition as

$$\dot{\theta}_r = (v + b\omega)/r,\ \dot{\theta}_l = (v - b\omega)/r \tag{7}$$

To make sure i-Go be passive and dissipative, the braking torques in Eq. (6) need to be non-negative, which demands the minimum allowable curvature radius of the trajectory to be larger than b for the wheel angular speeds in Eq. (7) to be nonnegative. In Eq. (6), the selection of braking gain $K=[K_r\ K_l]^T$ is very crucial for system performance, and its derivation can be referred to our previous work [34].

III. Proposed Scheme

The proposed collision-free receding horizon control scheme is developed to deal with the static and moving obstacles that may be present in the planned path for guidance. According to the direction and velocity of the detected obstacles, the strategy for obstacle avoidance will first determine what action should be taken to avoid collision, and then forward it the receding horizon passive controller for execution.

A. Strategy for Obstacle Avoidance

Because i-Go is intended for elderly assistance, it is not expected to move fast or to be used in a crowded environment. The moving obstacles are thus assumed to be few and move in fixed directions, like pedestrians in the neighborhood area. In responding to such situation, the actions for the robot walking helper to avoid collision are designed to be:

- Guiding: maintain moving on the planned path when the obstacles would not affect its motion.
- Waiting: wait for the obstacles to pass when its moving direction is different from that of them.
- Steering: steer it away from the obstacles when they may come to hit each other in the same direction.

The strategy is activated whenever obstacles are detected. To evaluate the influence from the obstacle, we first encompass the obstacle with a rectangle, shown in Figure 4, where (x_o, y_o) is the center of the obstacle, v_o its velocity (taken to be in x' direction), and $2r_o$ the width of each of the four boundaries with r_o that of the obstacle, where x'- and y'-axis are taken to be its moving and perpendicular direction, respectively. When the trajectory of i-Go, denoted as $q'(t)=(x'(t), y'(t))$, may encounter the moving rectangle at certain moment, we consider there is a potential for collision. When the collision may come from the front, the steering action will be taken, which may deviate i-Go to move away from the obstacle from the side, as shown in Figure 5(a), where τ_c is the time for the helper to move from the current location to reach the boundary of the rectangle. When it comes from the right, as shown in Figure 5(b), the waiting action will be taken with the time for waiting t_w calculated as

$$t_w = \frac{x'(t+\tau_c) + r_o}{v_o} \tag{8}$$

When the obstacle may not affect the planned guidance, the guiding action is taken. Note that for the case that the waiting time t_w is longer than a preset limit T_w, the steering action is suggested for speeding up the guidance. The algorithm for obstacle avoidance is summarized as follows:

Algorithm for Obstacle Avoidance: Input detected obstacle information, output the action of guiding, waiting, or steering to avoid possible collision.

Step 1: Initialize the system with the guiding action.

Step 2: Detect position (x_o, y_o) and velocity v_o of the obstacle.

Step 3: Generate a rectangle that encompasses the obstacle with (x_o, y_o) as the center and $2r_o$ as the width of the boundary.

Step 4: If the collision may come from the front, take the steering action, and go to Step 6;
else if the collision may come from the right or left, take the waiting action, calculate waiting time t_w using Eq. (8) and go to Step 5;
else, take the guiding action, and go to Step 6.

Step 5: If $t_w < T_w$, take the waiting action, and go to Step 6;
else, take the steering action, and go to Step 6.

Step 6: Send the selected action to the receding horizon passive controller to derive corresponding braking torques. Go to Step 2.

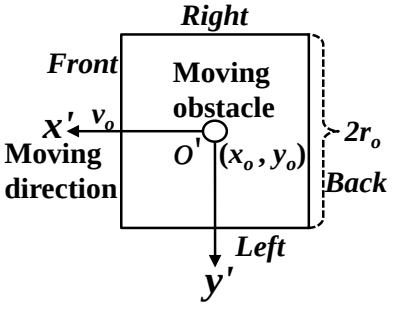
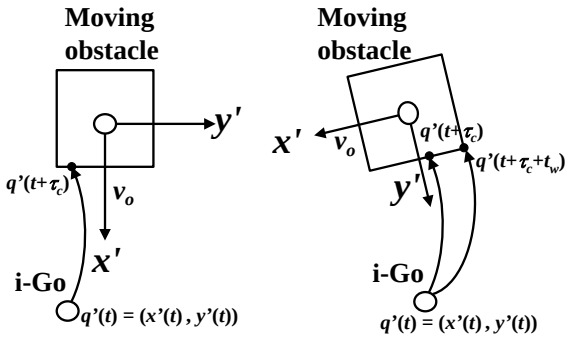


Figure 4 The rectangle that encompasses the moving obstacle for evaluating possible collision.



(a) Collision from the front (b) Collision from the right

Figure 5 Possible collision between the robot walking helper and moving obstacle: (a) collision from the front and (b) collision from the right.

B. Receding Horizon Passive Controller

The receding horizon passive controller is developed to derive corresponding braking torques for executing those actions determined by the strategy for obstacle avoidance. The process for control may be adjusted in responding to each of the actions. The control scheme is formulated as follows. Given the current state $q(t)=[x(t)\ y(t)\ \theta(t)\ v(t)\ \omega(t)]^T$ of the robot walking helper, where $x(t)$ and $y(t)$ are position, $\theta(t)$ directional angle, $v(t)$ velocity, and $\omega(t)$ angular velocity, respectively, and the initial path connecting the waypoints (x_i, y_i), $i=1,\ldots, n$, the acceleration and angular acceleration, $u=[a(t),\ \alpha(t)]^T$, can be obtained by solving the following optimal control problem:

$$\begin{aligned} &\min_{u} \quad J(q(t+T)) \\ &\textit{subject to } \ \dot{q} = f(q,u), q \in Q, u \in U \end{aligned} \tag{9}$$

where J and T are the cost function and time duration, respectively. Here, J is defined as

$$J = \frac{1}{2}\left(\left(x(t+T)-x_p\right)^2 + \left(y(t+T)-y_p\right)^2 + \left(\theta(t+T)-\theta_p\right)^2\right) \tag{10}$$

where x_p, y_p, and θ_p are current target point and orientation, (x_p, y_p) the position of the ith line segment connected with i-1th and ith waypoint, and θ_p the angle between the robot walking helper and ith line segment. Here, i is decided by $(x(t+T), y(t+T))$ which has minimum distance to ith segment path. To efficiently solve the optimal control problem, the input is defined as

$$u(\tau) = \begin{cases} [a_1 \quad \alpha_1], & \textit{if} \quad 0 \le \tau \le T_1 \\ [0 \quad 0], & \textit{otherwise} \end{cases} \tag{11}$$

where a_1 and α_1 are the constants and T_1 the control time. Motion state at time $t+\tau$, $\tau > T_1$ can then be obtained as

$$\begin{aligned} &x(t+\tau) = x(t) + v(t)F_c + \left(F_{ct1} + T_1 F_{c2}\right)a \\ &y(t+\tau) = y(t) + v(t)F_s + \left(F_{st1} + T_1 F_{s2}\right)a \\ &\theta(t+\tau) = \theta(t) + \omega(t)\tau + \left(\tau - T_1/2\right)T_1\alpha \\ &v(t+\tau) = v(t) + T_1 a \\ &\omega(t+\tau) = \omega(t) + T_1\alpha \\ &\textit{with} \\ &F_c = \int_0^{\tau} \cos\theta(t+s)ds\ , F_s = \int_0^{\tau} \sin\theta(t+s)ds \\ &F_{ct1} = \int_0^{T_1} s\cos\theta(t+s)ds\ , F_{st1} = \int_0^{T_1} s\sin\theta(t+s)ds \\ &F_{c2} = \int_{T_1}^{\tau} \cos\theta(t+s)ds\ , F_{s2} = \int_{T_1}^{\tau} \sin\theta(t+s)ds \end{aligned} \tag{12}$$

where F_c, F_s, F_{ct1}, F_{st1}, F_{c2}, F_{s2} can be calculated via Fresnel integrals [32]. With the function J of a_1 and α_1, the optimal control problem can be simplified to

$$\begin{aligned} &\min_{a_1,\alpha_1} \quad J(a_1,\alpha_1) \\ &\textit{subject to } \ ma_1 \pm \frac{I}{b}\alpha_1 - F \le 0 \end{aligned} \tag{13}$$

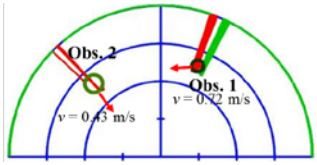
To efficiently solve the optimal control problem, we first fix α, and then calculate a that satisfies the passive constraints. The precise angular acceleration can be obtained by using golden section search [33].

As the steering action is activated when the obstacle approaches the robot walking helper from the front, it needs to turn left or right to avoid the incoming obstacle. Refer to Figure 5(a), when $y'(t+\tau_c) < 0$, i.e., the collision may come from the left, the turning direction should be left; when $y'(t+\tau_c) > 0$, it is vice versa. The controller will then regulate the angular acceleration to achieve the turning. As for the waiting action, which responds to the situation that the obstacle may come in from the right or left side, the speed needs to be slowed down enough for the obstacle to pass. The controller will then determine proper deceleration according to the distance between the moving obstacle and robot walking helper, with a safe margin taken as the buffer. Once the acceleration and angular acceleration are determined for the current action, corresponding control torques can then be calculated. This procedure for path guidance and obstacle avoidance is repetitively executed until the target location is reached.

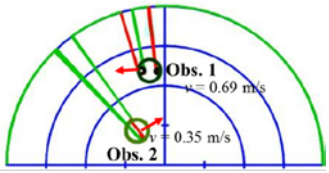
IV. Simulation and Experimental Results

We conducted simulations and experiments to evaluate the performance of the proposed scheme. Before that conducted for obstacle avoidance, we first validated the ability of the laser ranger finder in detecting the size and velocity of the moving obstacle. Figure 6(a) shows the experimental scene, and Figs. 6 (b) and (c) the detected results with the mean walking speeds of the subjects about 0.7 m/s and 0.39m/s, respectively. As the moving obstacles could be successfully detected using the laser range finder, we went on with the simulations and experiments for obstacle avoidance.

(a) Experimental scene



(b) First detection



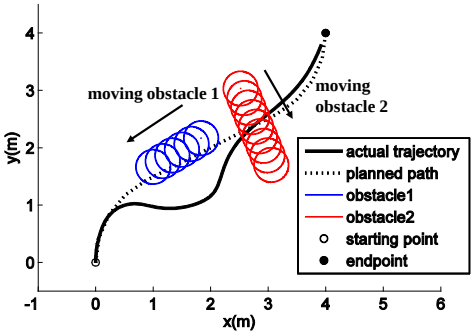
(c) Second detection

Fig. 6 Moving obstacle detection: (a) experimental scene, (b) first detection, and (c) second detection.

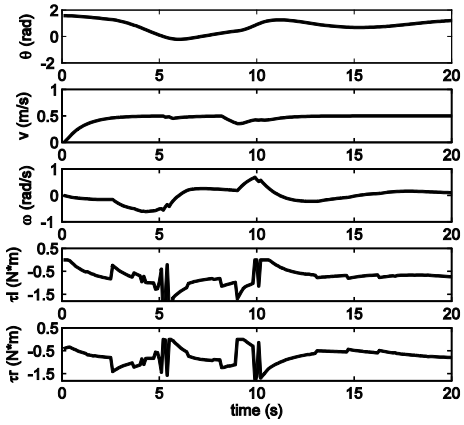
For the simulations, i-Go was used to guide the user to follow a planned path, while avoiding the incoming obstacles, with the parameters set to be m = 50 kg, I = 11.56 kgm^2, r = 0.0616 m and b = 0.34 m, which were the same as those of the real i-Go. The user-applied force was assumed to follow the function of $10(1+0.1\sin(\pi t/2))$, with a sampling time of 0.1s. Those parameters for the proposed scheme were chosen as T = 7 s, T_1 = 1 s, v_{max} = 0.5 m/s, ω_{max} = 1 rad/s, T_w = 5 s, and T_s = 1 s, respectively. The detecting range of the laser sensor was set to be 4 m in its front half circle. The start and end points (x, y, θ) were set to be (0 m, 0 m, π/2 rad) and (4 m,4 m, π/2), respectively. The initial position and velocity (x_o, y_o, v_o) of the moving obstacles were set as (2 m, 2 m, 0.5 m/s) and (1 m, 4.5 m, 0.5 m/s), respectively. Figures 7(a)-(c) show trajectories of i-Go and moving obstacles, motion states θ, v, ω, along with the braking torques τ_r and τl, and corresponding actions, respectively. As the first obstacle approached from the front at about 4 s., the steering action was taken, which turned i-Go to the right to avoid it. The waiting action was activated to wait for the second obstacle from the left to pass at about 9 s. Finally, i-Go reached the end point accurately at about 20 s., with braking torques all non-positive.

With the satisfactory simulation results, we then conducted the experiments in the hallway of our engineering building. The parameters for the proposed scheme were set to be the same as those used for simulations. We asked the user to push i-Go to follow a planned path, during which two pedestrians walked by in normal speeds. Figure 8 shows the experimental scene with the subject and i-Go. Figures 9(a)-(f) show the trajectories of i-Go and moving obstacles, motion states and braking torques, user-applied force, corresponding actions, and also the motions states of the two obstacles, respectively.

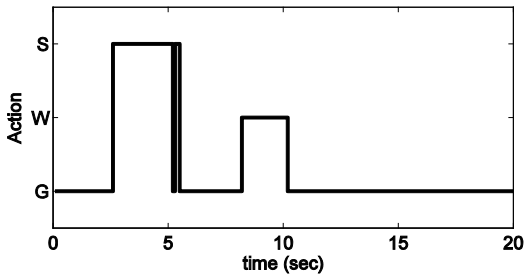
In Figure 9, the steering action was taken to avoid the first pedestrian by turning i-Go to the right, and the waiting action activated to avoid the second pedestrian from the left by reducing its speed to almost zero. In Figure 9(c), user-applied force was observed to be slightly oscillating at around 10 s. and 20 s., which corresponded to the transition between actions. This might be due to imprecise identification of the action caused by not so accurate obstacle detection, as shown in Figures 9(e)-(f). However, the accuracy of obstacle detection increased as the obstacle moved closer, which still yielded enough time for the scheme to activate the desired action. We also performed experiments under different setups with obstacles of various velocities. The results show that the proposed scheme well tackled obstacles under the speed of 0.8 m/s in not very crowded environments. Meanwhile, when the system is equipped with a better capability for obstacle detection, it should be able to handle obstacles of higher speed.



(a) Trajectories of i-Go and moving obstacles



(b) Motion states and braking torques



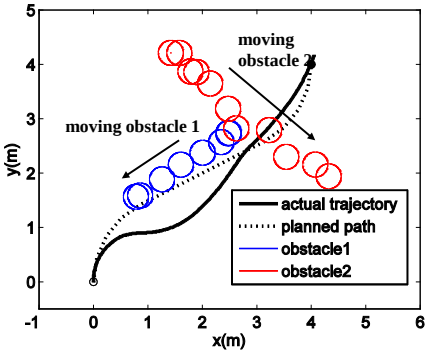
(c) Action

Figure 7 Simulation results for path guidance and obstacles avoidance: (a) trajectories of i-Go and moving obstacles, (b) Motion states and braking torques, and (c) corresponding actions.

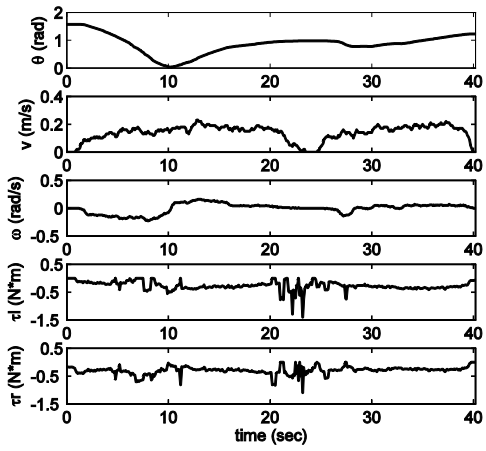
(a) Experimental scene

(b) Subject and i-Go

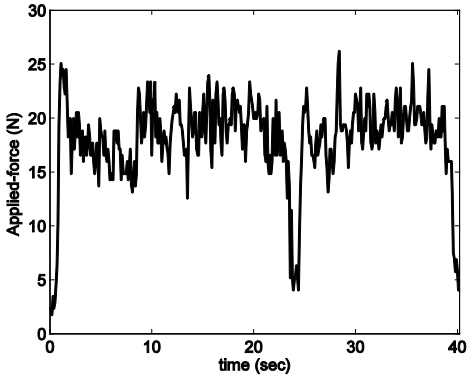
Figure 8 Experimental scene with the subject and i-Go.



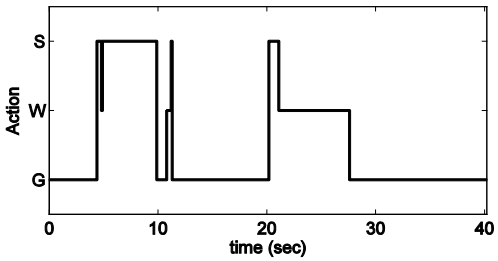
(a) Trajectory of i-Go and moving obstacles



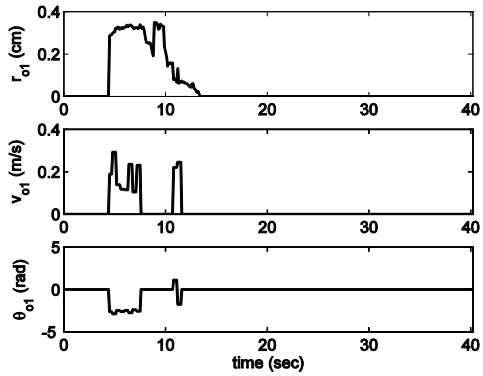
(b) Motion states and braking torques



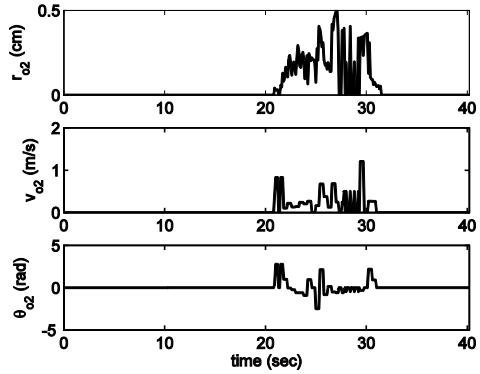
(c) User-applied force



(d) Action



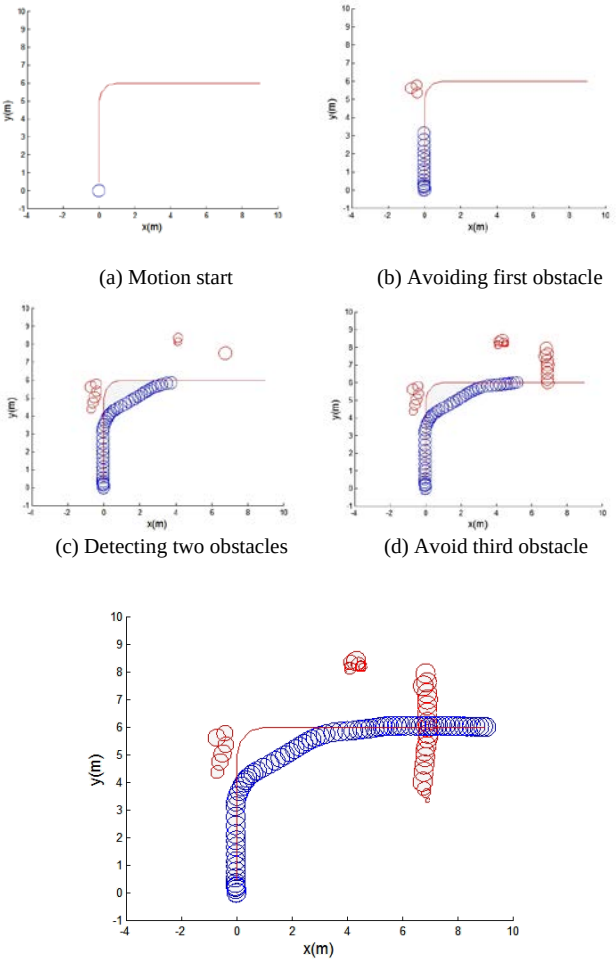
(e) Motion states (obstacle 1)



(f) Motion states (obstacle 2)

Figure 9 Experimental results for path guidance and obstacle avoidance: (a) trajectories of i-Go and moving obstacles, (b) motion states and braking torques, (c) user-applied force, (d) corresponding actions, (e) motion states (obstacle 1), and (f) motion states (obstacle 2).

We further conducted the experiment for the case with both static and moving obstacles. During the experiment, the subject would encounter one static and two moving obstacles. The initial locations for the two moving obstacles (obj 1 and obj 2) were set to be (-0.5m, 10m) and (7m, 17.5m), respectively, with their velocities around 0.2~0.5m/s. The static obstacle (obj 3) was a pillar, located at (4.7m, 8.6m). The subject was asked to move from (0m, 0m) to (6m, 9m) along an L-shaped path, as shown in Figure 10(a). During the guidance, the i-Go first detected an incoming obstacle (obj 1) and took the steering action to avoid it, as shown in Figure 10(b). It later detected two obstacles approaching, the second moving obstacle (obj 2) and the pillar (obj 3). The i-Go determined that the presence of obj 3 would not affect its motion and thus took no action, as shown in Figure 10(b). Meanwhile, obj 2 was considered dangerous, and avoided by taking the waiting action, as shown in Figure 10(d). Figure 10(e) shows the whole process, during which i-Go successful



(a) Motion start (b) Avoiding first obstacle

(c) Detecting two obstacles (d) Avoid third obstacle

(e) The whole process

Figure 10 The process for i-Go to avoid both static and moving obstacles.

guided the subject to pass through all the obstacles and reach the final goal, demonstrating the proposed scheme was capable of dealing with both static and moving obstacles altogether.

V. Conclusions

For its application in daily lives, in this paper, we have developed a collision-free guidance scheme for the passive robot walking helper, which includes a novel strategy for moving obstacle avoidance, and a receding horizon passive controller for execution. To demonstrate its effectiveness, experiments have been conducted in real environments based on our developed passive walking helper i-Go. In future works, we plan to apply it for more challenging environments, in which more pedestrians and objects with higher moving speeds may be present.

References

[1] C. Galindo, J. Gonzalez, and J. Fernández-Madrigal, "Control Architecture for Human–Robot Integration: Application to a Robotic Wheelchair," *IEEE Trans. Syst., Man, Cybern. B Cybern.*, vol. 36, no. 5, pp. 1053-1067, Oct. 2006.

[2] T. Carlson and Y. Demiris, "Collaborative Control for a Robotic Wheelchair: Evaluation of Performance, Attention, and Workload," *IEEE Trans. Syst., Man, Cybern. B Cybern.*, vol. 42, no. 3, pp. 876-888, Jun. 2012.

[3] H. Seki and N. Tanohata, "Fuzzy Control for Electric Power-Assisted Wheelchair Driving on Disturbance Roads," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no.6, pp. 1624-1632, Nov. 2012.

[4] D. Chugo, W. Mastuoka, S. Jia, and K. Takase, "The Wheel Control of a Robotic Walker for Standing and Walking Assistance with Stability,'' *IEEE RO-MAN*, pp. 297-302, Aug. 2008.

[5] K. Wakita, J. Huang, P. Di, K. Sekiyama, and T. Fukuda, "Human-Walking-Intention-Based Motion Control of an Omnidirectional-type Cane Robot," *IEEE/ASME Trans. on Mechatronics*, vol. 18, no. 1, Feb. 2013.

[6] P. Di, J. Huang, S. Nakagawa, K. Sekiyama and T. Fukuda, "Fall Detection and Prevention in the Elderly based on the ZMP Stability Control," *IEEE Int. Conf. on Workshop on Advanced Robotics and its Social Impacts*, pp. 82 - 87, 7-9, Nov. 2013.

[7] G. Lee, T. Ohnuma, and N. Y. Chong, "Design and Control of JAIST Active Robotic Walker," *Journal of Intelligent Service Robotics*, vol. 3, no. 3, pp. 125-135, Jul. 2010.

[8] W. H. Mou, M. F. Chang, C. K. Liao, Y. H. Hsu, S. H. Tseng , and L. C. Fu, "Context-Aware Assisted Interactive Robotic Walker for Parkinson's Disease Patients," *IEEE Int. Conf. on Intelligent Robots and Systems*, Oct. 2012.

[9] S. K. Banala, S. H. Kim, S. K. Agrawal, and J. P. Scholz, " Robot Assisted Gait Training With Active Leg Exoskeleton (ALEX)," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 17, pp. 2-8, Feb. 2009.

[10] E. S. Boy, E. Burdet, C. L. Teo, and J. E. Colgate, "Investigation of Motion Guidance With Scooter Cobot and Collaborative Learning," *IEEE Trans. on Robotics*, vol. 23, no. 2, pp. 245–254, Apr. 2007.

[11] H. Y. Yu, M. Spenko, and S. Dubowsky, "An Adaptive Shared Control System for an Intelligent Mobility Aid for the Elderly," *Journal of Autonomous Robots*, vol. 15, pp. 53–66, Apr. 2003.

[12] J. C. Ryu, K. Pathak, and S. K. Agrawal, "Control of a passive mobility assist robot," *Journal of Med. Devices*, vol. 2, no. 1, pp. 011002, Mar., 2008.

[13] S. Suzuki, Y. Hirata, K. Kosuge, "Development of Intelligent Passive Cane Controlled by Servo Brakes," *IEEE Int. Symp. on Robot and Human Interactive Communication*, Sep., 2009.

[14] Y. Hirata, A. Hara, and K. Kosuge, "Motion Control of Passive Intelligent Walker Using Servo Brakes," *IEEE Trans. on Robotics*, vol. 23, no. 5, pp. 981-990, Oct. 2007.

[15] Y. C. Huang, H. P. Yang, C. H. Ko, and K. Y. Young, "Human intention recognition for robot walking helper using ANFIS," *IEEE Int. Conf. on Asian Control*, pp. 311-316, May 2011.

[16] S. Taghvaei, Y. Hirata, and K. Kosuge "Comparative Study of Visual Human State Classication; an Application for a Walker Robot," *IEEE Int. Conf. on Biomedical Robotics and Biomechatronics*, Jun. 2012.

[17] C. H. Ko, K. Y. Young, Y. C. Huang, and S. K. Agrawal, "Active and Passive Control of Walk-Assist Robot for Outdoor Guidance," *IEEE/ASME Trans. Mechatronics*, vol. 18 , no.3, pp. 1211–1220, Jun. 2013.

[18] S. Taghvaei, Y. Hirata, and K. Kosuge "Vision-based Human State Estimation to Control an Intelligent Passive Walker," *IEEE/SICE Int. Symp. Syst. Integration*, pp. 146–151, Dec. 2010.

[19] M. Spenko, H. Yu, and S. Dubowsky, "Robotic Personal Aids for Mobility and Monitoring for the Elderly," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 3, pp. 344–351, Sep., 2006.

[20] B. Graf, "An Adaptive Guidance System for Robotic Walking Aids," *Journal of Comput. Inf. Technol.*, vol. 17, no. 1, pp. 109–120, 2009.

[21] J. Manuel, H. Wandosell, and B. Graf, "Non-Holonomic Navigation System of a Walking-Aid Robot," *IEEE Int. Conf. on Workshop Robot Human Interact. Commun.*, pp. 518–523, Sep., 2002,.

[22] O. Chuy, Y. Hirata, Z. Wang, and K. Kosuge, "A Control Approach Based on Passive Behavior to Enhance User Interaction," *IEEE Trans. on Robotics*, vol. 23, no. 5 pp. 899–908, Oct., 2007.

[23] M. M. Martins, C. P. Santos, A. Frizera-Neto, and R Ceres, "Assistive Mobility Devices Focusing on Smart Walkers: Classification and

Review," *Journal of Robot. Auton. Syst.*, vol. 60, no.4, pp. 548–562, Apr., 2012.

[24] C. H. Ko, K. Y. Young, and Y. H. Hsieh, "Optimized Trajectory Planning for Mobile Robot in the Presence of Moving Obstacles," *IEEE Int. Conf. on Mechatronics*. pp.66–71, Mar., 2015.

[25] M. Seder and I. Petrovic, "Dynamic Window Based Approach to Mobile Robot Motion Control in the Presence of Moving Obstacles," *IEEE Int. Conf. on Robot. Autom.*, pp. 1986–1991, Apr., 2007.

[26] K. T. Yu, C. P. Lam, M. F. Chang, W. H. Mou, S. H. Tseng, and L. C. Fu, "An Interactive Robotic Walker for Assisting Elderly Mobility in Senior Care Unit," *IEEE Int. Conf. on Workshop on Adv. Robot. Social Impacts*, pp. 24–29, Oct., 2010.

[27] N. Nejatbaksh and K. Kosuge, "User-Environment Based Navigation Algorithm for an Omnidirectional Passive Walking Aid System," *IEEE Int. Conf. on Rehabi. Robot.*, pp. 178–181, Jul. 2005.

[28] Y. C. Huang, C. J. Wu, C. H. Ko, K. Y. Young, "Collision-Free Guidance for Passive Robot Walking Helper, " *IEEE Int. Conf. on Syst., Man, Cybern.*, pp. 3129–3134, Oct. 2012.

[29] C. H. Ko, Y. H. Hsieh, Y. T.Chang, S. K. Agrawal, and K. Y. Young, "Guidance and Obstacle Avoidance of Passive Robot Walking Helper Based on Receding Horizon Control," *IEEE Int. Conf. on Autom. Sci. Eng.*, pp. 1032-1037, Aug. 2014.

[30] C. H. Ko, K. Y. Young, and S. K. Agrawal, "Receding Horizon Passive Control for a Walk-Assist Robot," *IEEE Int. Conf. on Cont. Autom. Syst.*, pp.1474–1479, Oct. 2012.

[31] S. M. LaValle, *Planning Algorithm*, Cambridge University Press, 2006.

[32] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: the Art of Scientific Computing*, Cambridge University Press, 2007.

[33] M. Missura, and S. Behnke, "Efficient Kinodynamic Trajectory Generation for Wheeled Robots," *IEEE Int. Conf. on Robot. Autom.*, pp. 4883–4888, May 2011.

[34] Y. H. Hsieh, K. Y. Young, and C. H. Ko, "Effective Maneuver for Passive Robot Walking Helper Based on User Intention," *IEEE Trans. Ind. Electron.*, vol. 62, issue 10, Oct. 2015.

Yi-Hung Hsieh received the Ph.D. degree in electrical and control engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2016.

He is currently a Supervisor Engineer with Silicon Motion Technology Corp. His main research interests include robot walking helpers, modeling and control of dynamic systems, and learning systems.

Yu-Hsuan Su is a Ph.D. student in School of Electrical Control Engineering at National Chiao Tung University, Hsinchu, Taiwan. He received his Master degree in Mechatronics Engineering from Changhua University of Education in 2011.

His main research interests include robot control systems, mechatronics, and machine vision systems.

Chih-Chung Chiu is a Ph.D. student in School of Electrical Control Engineering at National Chiao Tung University, Hsinchu, Taiwan. He is currently working in Industrial Technology Research Institute, Taiwan.

His main research interests include robot walking helpers and learning systems.

Kuu-Young Young (M'86–SM'04) received his B.S. degree in electrical engineering from National Taiwan University, Taiwan, in 1983, and M.S. and Ph.D. degrees in electrical engineering from Northwestern University, Evanston, IL, U.S.A., in 1987 and 1990, respectively.

Since 1990, he has been with the Department of Electrical Engineering at National Chiao Tung University, Hsinchu, Taiwan, where he is currently a Professor. He served as the chairman of the department from 2003 to 2006, and the associate dean of Electrical and Computer Engineering College, NCTU, from 2007 to 2010 and also 2014 to 2015. His research interests include robot compliance control, robot learning control, robot calibration and path planning, teleoperation, and robot walking helper.

Chun-Hsu Ko received the M.S. degree in power mechanical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1991, and the Ph.D. degree in electrical and control engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2003. From 1994 to 1998, he was with the Industrial Technology Research Institute, Hsinchu, Taiwan, as an Associate Researcher.

He is currently a Professor in the Department of Electrical Engineering, I-Shou University, Kaohsiung, Taiwan. His research interests include robot control, robot walking helpers, and optimization.

Simultaneous Localization and Mapping with Neuro-Fuzzy Assisted Extended Kalman Filtering

Cong Hung Do[1], Huei-Yung Lin[2], and Yi-Chun Huang[1]

***Abstract*—We want to improve the performance in larger scale of environment features, let robot move itself more accuracy inspired by neuro-fuzzy based adaptive EKF is provided. In this case, we present the development based on it and adjusted R matrix at each running step estimating proper values. Needing robot move more accuracy, should reduce the mismatch between the two covariance matrix: theoretical and actual. Using the particle swarm optimization (PSO) replaced Back propagation (BP) trained in this system because PSO iterated and adjusted the weight factors and the consequent of the network to let error cost function minimum. By employing PSO we can exploit the advantages of the high-dimensional search space algorithm for more effective training of ANFEKF. Our experiment result divided two part: simulation and real implement, using the mobile robot platform under two benchmark of environment situation with a number of landmarks.**

I. Introduction

The SLAM problem has been studied and developed for many decades and now is still a challenging problem that attracts many concerns from researchers. The SLAM based mobile robot platform is used to deal with the problem of localizing the robot position relative to the environment and simultaneously building the environment map.

The relative SLAM and absolute SLAM is important in this kind of research. The relative SLAM means that the robot localizes itself and builds a map based on the odometry data provided by the encoder attached on the robot wheels. The relative SLAM is also known as the dead reckoning [1] method and the main disadvantages of this approach is that the error accumulation over the running time, which leads to the significant degradation of robot localization and mapping. The absolute method takes the advantage of environment measurements provided from a set of external sensors to perform localization and mapping functions. This approach is also known as the beacon based SLAM or SLAM based on landmarks observation [2]. The SLAM algorithm had drawback on accuracy that it significantly rely on the observed features in the environment and the reliability of the external sensors. Possible changes to the environment features or too noisy sensor reading information may lead to the wrong inference of the SLAM results.

To overcome those foregoing drawbacks, this paper that synergistically combines the information from multiple sensors to provide more reliable and accurate information. In this work, the odometry sensor and the range finder sensor information through EKF combined each other. There are some similar works proposed before [3], [4], [5]. However, designing an effective EKF method requires priori knowledge and the measurement, it represented by matrix Q and matrix R. During most practical situations these parameters are undefined and usually set as fixed values throughout the running time. This set up usually results not only caused the measurement noise but also the process in the poor estimation. Unfortunately, it leads to the significant degradation of the EKF performance. Another widely known problem is the computational burden higher in the classical full EKF-based SLAM with a large number of features, effecting the total robot state and total covariance matrix. One of the effective solutions to overcome this weakness is to employ adaptive algorithms for SLAM. In the adaptive SLAM, there are also many methods proposed [6], [7], [8], [9], [10].

In this paper, a SLAM with neuro-fuzzy aided EKF approach is presented. Particularly, the matrix R in adaptive neuro fuzzy want to adapt by itself while maintaining the value of Q fixed (assume Q is completely known). Then many free parameters exist in the neuro-fuzzy model, learned by the particle swarm optimization (PSO), which is a random search based metaheuristic optimization procedure. The contribution of this paper is designed and adjustment of adaptive neuro fuzzy EKF (ANFEKF). It improves the performance of SLAM in the larger scale of environment features which cannot be achieved by conventional EKF approaches in terms of computational cost, `performance efficiency and real time implementation.

There are many other studies on this topic available, most of them are published in the recent years and currently stand still the ongoing development. This begs the question: why yet another paper on the topic of adaptive fuzzy neuro based mobile robot localization? The reason that this work is to offer an intuitive solution to the SLAM problem in three layers. Firstly, the solution is proposed primarily based on the most well-known optimal state estimation approach, extended Kalman filter [11], [12], [9]. Secondly, the artificial intelligence approach, or to be more specifically the adaptive neuro fuzzy [13], [14], is also applied with the aim of optimizing the basic EKF performance. Finally, the heuristic search [15], PSO, is employed for assisting the adaptive neuro fuzzy in terms of the learning process. A few papers address some of the above solutions [16], [14], but not many offer all three layers of solution presented in this work.

[1]Cong Hung Do and Yi-Chun Huang are with the Department of Electrical Engineering, National Chung Cheng University, Chiayi 621, Taiwan.

[2]Huei-Yung Lin is with the Department of Electrical Engineering and Advanced Institute of Manufacturing with High-Tech Innovation,

II. Approach

A. Robot Kinematics Modeling and Extended Kalman Filter

If we heard wheeled cars, they would be famous on class of vehicles research. It seems like the typical archetype of most ground robots. Generally, it had commonly the bicycle model to represent for a four-wheeled car-like vehicle followed by

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix} = f(X) = \begin{bmatrix}(V+v_v)\cos(\phi + [\gamma + v\gamma])\\ (V+v_v)sin(\phi + [\gamma + v\gamma])\\ \frac{(V+v_v)}{B}sin(\gamma + v_v)\end{bmatrix} \quad (1)$$

Dynamic model is written by $\begin{bmatrix}\dot{x} & \dot{y} & \dot{\phi}\end{bmatrix}$. It related with the Cartesian coordinates [*x, y*] and the robot orientation ϕ, which respect to the global coordinates respectively. Let u = [*V*, γ], two parameters are separated control velocity V and steering angle γ. Dynamic situation might consider noise existed. The covariance matrix of the process noise v = [v_v v_γ], one is the noise of input *V*, the other is steering angle. Finally, *B* is the baseline of the vehicle or robot.

A mobile robot adopts external sensors for sensing the features of the physical environment. In this work, the sonar sensor used the external sensor of the robot to provide the measurements of range r_i and bearing θ_i of the observed features ($\underline{x_i}$ y_i). The mathematical model can be represented by

$$\begin{bmatrix}r_i\\ \theta_i\end{bmatrix} = h(X) = \begin{bmatrix}\sqrt{(x-x_i)^2 + (y-y_i)^2}\\ +\omega_r tan^{-1}\frac{y-y_i}{x-x_i} - \phi + \omega_\theta\end{bmatrix} \quad (2)$$

where the two noise variances ω_r and ω_θ is applied to the range and bearing measurements respectively, and W = [ω_r ω_θ] is the covariance matrix of the noise observation.

In 1960, Kalman solved the linear-quadratic problem used a recursive solution, which is he presented. The linear-quadratic problem is associated with the current state of a linear dynamic system (or state of process) perturbed by white noise estimated. To estimate (or predict) the current state of the process, Kalman filter kept recursive way with the current measurement and the priori state of the process simultaneously.

EKF is an expanded version of Kalman filter, that aimed to deal with non-linear models. The only difference between Kalman filter and EKF is adding extra linearization model calculating the partial derivatives of the state variables on prediction step. The general non-linear system and the measurement mathematical form is described by

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + \omega_k \\ z_k &= h(x_k) + v_k \end{aligned} \quad (3)$$

where x_k, z_k, u_k are appeared on function (3).First and Second parameters are severally means at sampling instant *k*, having *n*×1 process state vector and the m × 1 measurement vector. The u_k is the control input. Random variables w_k and v_k, means the Gaussian white process noise and measurement noise respectively. P_k, Q_k and R_k are totally equivalent to covariance matrices for x_k, ω_k and v_k severally.

B. SLAM based on EKF

In this work, the map of the environment is given. It means that the robot knows the map and the environment features in advance. The EKF filter estimates the robot pose by fusing the information provided by the odometer, map and range finder sensor. The Robot need to move accuracy by following the algorithm, which recursive steps of position prediction, observation, measurement prediction, matching and estimation.

1) Robot Position Prediction: Predicting at time step k+1 robot position, it is related with the time step k location. The control input in movement u_k is given by

$$\begin{aligned} \hat{x}_k^- &= f(x_k, u_k) \\ P_k^- &= F_k P_k F_k^T + G_u Q_k G_u^T \end{aligned} \quad (4)$$

where Fk = $\frac{\partial f}{\partial X}$ and Gu = $\frac{\partial f}{\partial u}$.

2) Observation: The second step about the robot at time k+1 sensor measurements Z_{k+1}.

3) Measurement Prediction: Pass by (4) calculating, we have the predicted robot position $\hat{x}_{k+1}^-$.Using this information and the current map,we can produce the predicted measurement $\hat{z}_k$ according to

$$\hat{z}_k = \begin{bmatrix}\sqrt{(\hat{x}-x_i)^2 + (\hat{y}-y_i)^2}\\ tan^{-1}\frac{\hat{y}-y_i}{\hat{x}-x_i} - \hat{\phi}\end{bmatrix} \quad (5)$$

Predicted measurement and actual measurement z_k might have little difference or existed error. Estimation side consid- ered it and followed by innovation sequence v_k (or residual) with the covariance $S_k = H_k P_k^- H_k^T + R_k$ and $H_k = \frac{\partial h}{\partial X}$. It is simple written by $v_k = z_k - \hat{z}_k$.

4) Matching: Matching part will use landmarks (store in the map) and observe measurements to let robot adjust itself navigating closed to original path.

5) Estimation: The estimation section can be described by

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + K_k\left(z_k - h(\hat{x}_k^-)\right) \\ K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ P_k &= (I - K_k H_k) P_k^- \end{aligned} \quad (6)$$

Done this function, we can get Kalman gain K_k and new state covariance matrix P_k.

C. Neuro-Fuzzy Assisted EKF applied on Slam Problems

1) SLAM with ANFIS: In Kalman filter, it might have many factors decrease the estimation performance. The most of all is the noise covariance. Noise covariance divided into the process Q and measurement R, there are matrices form. The initial values of Q and R significantly affect the performance and Kalman gain of the filter. It is simply that the values of R and Q tell the system to trust more on the prediction equation or on the measurement equation.

To achieve the well-defined model of Q and R, trial and error is a common way but it is often time-consuming and the result may not yet be an optimal model. In our work, we

combined two ideas to adjust the values of R (with Q fixed). One is an innovation adaptive estimation (IAE) scheme of the EKF, the other one is adaptive neuron fuzzy inference system (ANFIS). ANFIS is employed to quickly find the optimal model for R and prevent the EKF from being diverged.

2) Localization Based on ANFIS EKF (with Q fixed): As mentioned previously, we knew that R and Q were noise covariance matrices, they could be written by R_k and Q_k. There are two major ways to make EKF become adaptive. One way is to adjust R and Q simultaneously, and the other is to set one of them fixed and adjust the remain. In this paper, our algorithm based on adaptive estimation (IAE) to adapt the EKFs noise covariance matrix R with Q is fixed. The basic idea of IAE approach is to make the actual value C_k and theoretical value of S_k equally. S_k is derived from EKF, then actual covariance C_k can close to averaging its sampling covariance r_k with fixed moving window of size N. The form is

$$\hat{C}_k = \frac{1}{N}\sum_{i=k-N+1}^{k} (r_i^T r_i) \tag{7}$$

Inside the estimation window had k-th sample, equal to i_k. For simplicity, the objective of the IAE approach can be considered as degrading the mismatch between S_k and C_k as far as possible. The mismatch is denoted as DOM (degree of mismatch) and defined as

$$\mathrm{DOM}_k = S_k - \hat{C}_k \tag{8}$$

To reduce the DOM, the value of S_k is adjusted by varying R. In other words, the adaption of element (i, i) of R is made on the basis of the corresponding element (i, i) of DOM by employing ANFIS. The general rules for the adaption of R are as follows:

- If $\mathrm{DOM}_k(i, i) = 0$ then unchanged R_k;
- If $\mathrm{DOM}_k(i, i) > 0$ then declined R_k;
- If $\mathrm{DOM}_k(i, i) < 0$ then enhanced R_k;

3) The ANFIS Architecture: In this work, the ANFIS model is designed for single input and single output. The input of ANFIS is DOM_k and the output is the adjustment factor △R. We used the covariance matrix R adjusted by the following relation:

$$R_k \leftarrow R_k + \Delta R_k \tag{9}$$

ΔR_k is the output of ANFIS. Since the size of DOM_k and R_k is a two-by-two matrix and the two diagonal elements of these matrices corresponding to the error variances of range and bearing of the measurement, two ANFIS systems are employed as shown in Fig. 1. The ANFIS system employed to adjust ΔR_k has the construction of four-layer network as shown in Fig. 2. In our ANFIS model designed, we changed original multiple input and single output concept, tend to let model simple and improve performance, the same designed is our system with five-layer architecture. Basically, using u_i^l and o_i^l represent input and output with i-th node and the l-th layer. Each layer would describe clearly as follows:

- **Layer1** Input layer. In our work, the input value directly transfers to the next layer. Input-output relation of this node is $O^1 = u^1 = DOM_k$.

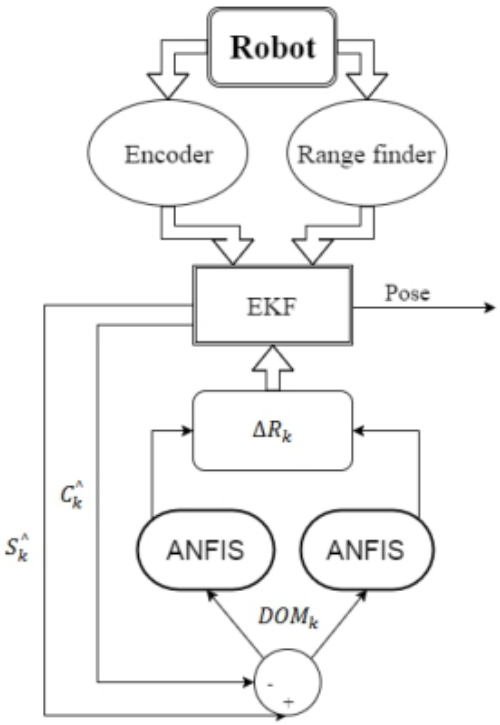


Fig. 1: The designed ANFIS system employed to adjust the ∆R value.

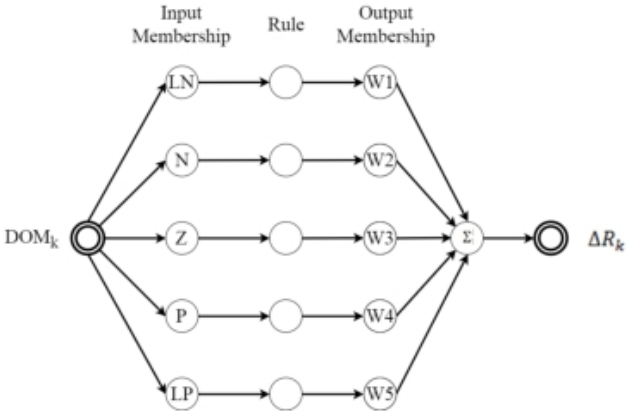


Fig. 2: Four-layer network of the employed ANFIS structure.

- **Layer2** Input membership function layer. Due to input variable are fuzzified by referred to five member-ship functions (MFs), it might divided into Large negative (LN). negative (N), zero (Z), positive (P) and final is Large positive(LP). The output of the i-th MF is denoted by

$$o_{ij}^2 = \mu_{ij}(u^2) = \exp\left\{\frac{-(u_{ij}^2 - {m_{ij}}^2)}{\delta_{ij}^2}\right\} \tag{10}$$

where m_{ij} and δ_{ij} are the mean and the width of the Gaussian MF. In each node, there are two free parameters which are adjustable.

- **Layer3** This layer let input nodes (value) through fuzzy and operation (or product inference) for calculating the firing strength by rule nodes.

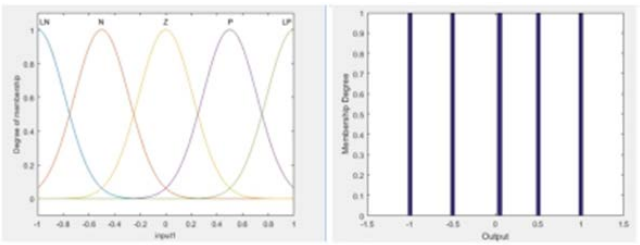


Fig. 3: Initial input output structure of ANFIS.

Label		LN	N	Z	P	LP
Input	Center	-1	-0.5	0	0.5	1
	Width	0.2123	0.2123	0.2123	0.2123	0.2123
Output	Weight	1	1	1	1	1

Fig. 4: Initial ANFIS membership functions for input (DOM) and the weight for output (∆R).

It seems like this

$$o_l^3 = \prod_i u_i^3 \tag{11}$$

- **Layer4** It let nodes of firing strength normalization from layer 3. Formulation show this layer's operating process

$$o_l^4 = \frac{u_l^4}{\sum_{l=1}^{n} u_l^4} \tag{12}$$

- **Layer5** This layer we will get a suitable scaling for the defuzzified output. Between in and out relation is written by

$$\Delta R_i = \sum_{l=1}^{N} u_l^5 \omega_l \tag{13}$$

4) Trainning on The Neuro-Fuzzy Model Using PSO: The purpose of training algorithms is to let cost function minimized with adjust free parameters of the network. The cost function in this work is represented by

$$\mathrm{E} = \frac{1}{2} e_k^2 \tag{14}$$

where $e_k = S_k - \hat{C}_k = DOM_k$

There are several ways to train ANFIS system; the most conventional way is the backward propagation approach. However, this approach easily gets stuck in local optima and therefore is not suitable for a high non-linear model. To overcome this drawback, the PSO algorithm is employed for a more effective training of ANFIS system. By employing the PSO training algorithm, adjusting the weighing factor of bot antecedent and the consequent of the network pass by specific number of training iterations, led error cost function less than the desired threshold value.

We use one input to ANFIS system, namely the degree of mismatch DOM, and one output, △R, the adjustable factor of measurement covariance matrix R. Using 5 membership functions for each of our inputs give us 10 parameters (5×2) since each function has 2 parameters associated with it. A total of 5 output rules are used, with each output a singleton (using the Sugeno model), and functions as the weights of the neural network. Hence, we have 10+5=15 parameters to be optimized. That is, our neural network consists of one

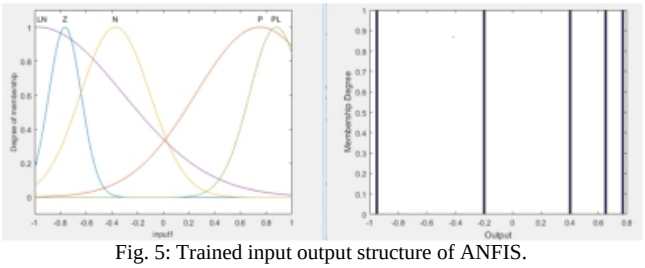


Fig. 5: Trained input output structure of ANFIS.

Label		LN	N	Z	P	PL
Input	Center	-0.976	-0.373	-0.7647	0.752	0.878
	Width	0.6649	0.27	0.126	0.5016	0.2087
Output	Weight	0.95	0.4	8	1.3	0.77

Fig. 6: Trained ANFIS membership functions for input (DOM) and the weight for output (∆R).

input neuron, 5 hidden-layer neurons, and one output neuron. In PSO had set 15-dimensional particle, the scale factors for antecedent parameters (*mLN, δLN, mN, δN, mZ, δZ, mP, δP, mLP, δLP*) and the scale factors for consequent parameters (*W*1, *W*2, *W*3, *W*4, *W*5), totally were fifteen parameters. During the training time, both the parameters of antecedent and consequent are adjusted at same time by the corresponding scale factor.

The general working procedure of the PSO learning algorithm is described as follows:

- Initialization: Each particle is initialized with each random vector of the search space.
- Updating Fitness Position: Based on the best local fitness location and global fitness location, the velocity and the position of a particle is adjusted accordingly.
- Convergence: The process repeats iteratively until the stopping condition is met.

Figs. 3-6 demonstrate the influence of the learning algorithm to the ANFIS structure. Fig. 3 illustrates the initial input-output structure of ANFIS with its corresponding parameters as depicted in Fig. 4. The input membership function type is designed as Gaussian shape and the output type is Singleton. Similarly, Figs. 5 and 6 illustrate the trained input-output structure of ANFIS and their corresponding parameters.

III. Simulation and Real Implementation

We evaluate our approach in both simulation experiment and practical implementation.

A. Experimental Results

For the simulation experiment, we evaluate our algorithm by testing it on two different cases with different number of landmarks. Figs. 7 and 8 show the robot trajectory and landmark positions. The blue dots signify observed landmarks while the red dots signify all of the known landmarks in the environment. The red dot line represents the estimated robot pose over time while the black line denotes the ground truth. The initial robot position is set at $[x\ y\ \theta]$ = [190 25 0]. The moving speed of robot is set at

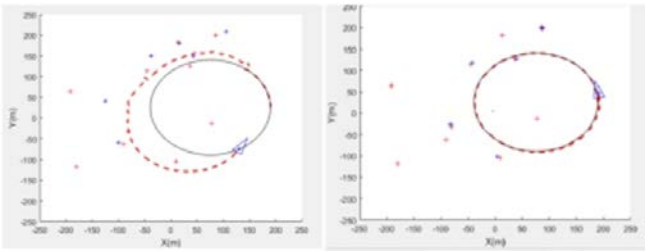


Fig. 7: Conventional EKF based SLAM (left) and ANEKF based SLAM (right) with 10 landmarks in the environment.

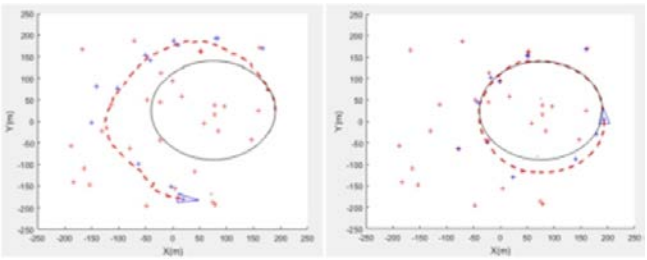


Fig. 8: Conventional EKF based SLAM (left) and ANEKF based SLAM (right) with 30 landmarks in the environment.

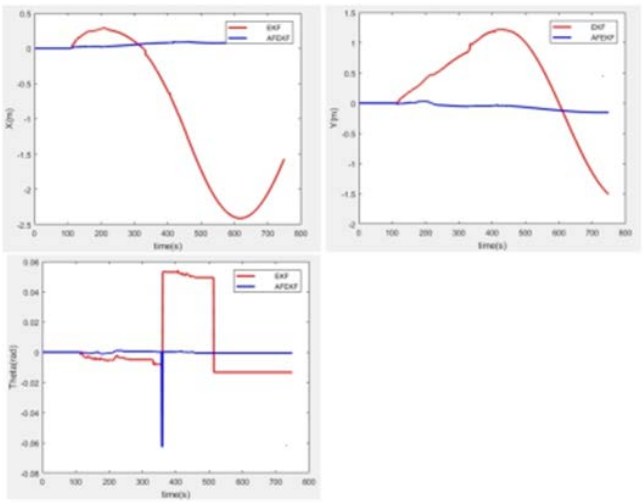


Fig. 9: The estimation error in 10 landmarks environment.

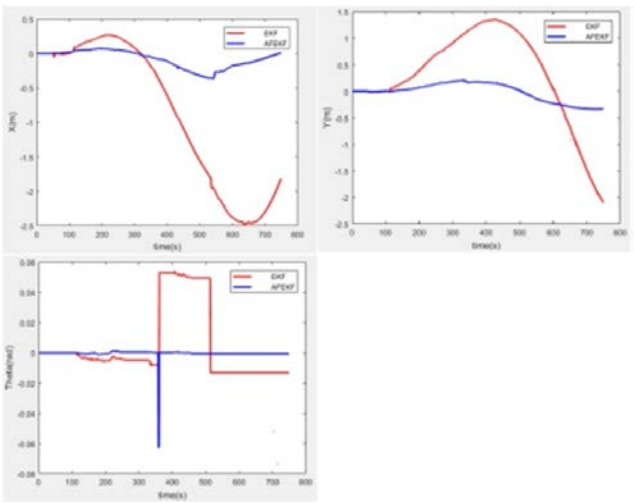


Fig. 10: The estimation error in 30 landmarks environment.

1m/s with the fixed steering angle at 0.5 deg. The equipment of robot using range- bearing sensor with 60 degrees of front field of view and the maximum measure range of 100 meters. To simulate the real implementation,

Fig. 11: The Implemented Environment.

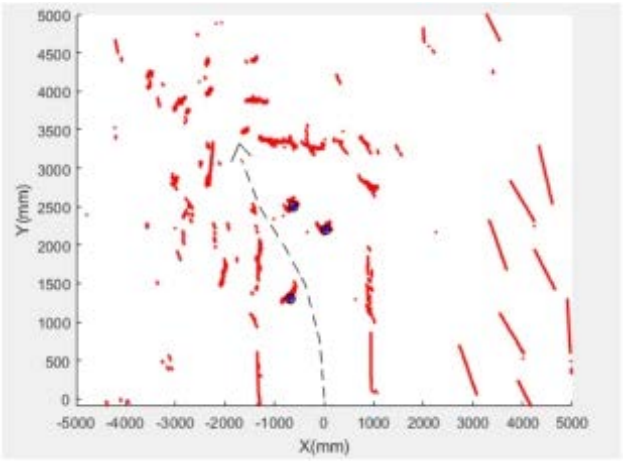


Fig. 12: The complete map build by ARIA mobile robot.

control noise and measurement noise are set as random distributed with zero mean and variance at 1m.

The proposed method is carried out in two different cases, the first case with 10 landmarks and the second case contains 30 landmarks. In simulation side of our research, using the conventional EKF based SLAM compared with ANFEKF. In the conventional EKF based SLAM, the matrices Q_k and R_k are set fixed throughout the experiment while in the ANFEKF based SLAM, Q_k is set at the same value but R_k is adjusted by ANFIS system as an adaptive parameter.

Figs. 9 and 10 show the estimation error of the robot state of both approaches over 750-run times. As observed from the figures that, with 30 landmarks, the estimation error of ANFEKF is close to zero, which is much less than that of the conventional method. Because results show that ANFEKF more closed to zero, it performance is greater than the conventional EKF based SLAM.

B. Real Implementation

In this section, the real implementation on an Aria mobile robot platform is carried out to illustrate our ANFEKF based SLAM capability. Using encoders with wheels of the robot because of measuring the rotation with the internal sensors. If used encoder individually, it would lead to the error accumulation while robot travelled far distance.

Therefore, the integration with external sensors is necessary to relocate the robot position more accurate and reliable. In our approach, the on-board sonar sensors are regarded to external sensors, using sensory reading fused with the odometry data(from internal sensor) to estimate the robot position.

The environment for the real implementation is set up as shown in Fig. 11. The Aria robot starts from the left most corner (blue rectangle) and needs to reach its destination at the right most corner (blue rectangle) while simultaneously localizing itself and building the map on the move. There are three landmarks (red circle) shown in the figure. The robot utilizes odometry data from the wheel-encoder to localize its pose and integrate with observed landmarks to relocate the robot position more accurate and reliable.

Fig. 12 represents the completed map when the mo- bile robot reaches the target. The black dash line de- notes the robot pose throughout the run time; the black circle denotes the observed landmarks while the red dot represents the environment map. For more information on the experiments, please refer to the following link:

https://www.youtube.com/watch?v=0cI5wV3pfLY

IV. Conclusions

In this paper, we present the development of an adaptive neuro fuzzy inference system for SLAM based EKF. The conventional EKF is useful with the assumption that the priori knowledge is well known. However, in the practical applications, this knowledge is biased and may cause the sig- nificant degradation of the conventional EKF performance. Our ANFEKF is developed with the aim of dealing with the incorrect knowledge problem by adapting the R matrix. The main advantage of the proposed approach is the consistency in comparison with the conventional approaches. Two cases of simulation experiments are carried out together with the real implementation on a mobile robot to study and evaluate the performance of our work. The results show that SLAM based ANFEKF performs more accurate than that of the conventional EKF.

Acknowledgement

The support of this work in part by the National Science Council of Taiwan under Grant NSC-102-2221-E-194-019 is gratefully acknowledged.

References

[1] L. Ojeda and J. Borenstein, "Personal dead-reckoning system for gps- denied environments," in *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*. IEEE, 2007, pp. 1–6.

[2] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE transactions on robotics and automation*, vol. 13, no. 2, pp. 251– 263, 1997.

[3] L. Jetto, S. Longhi, and D. Vitali, "Localization of a wheeled mobile robot by sensor data fusion based on a fuzzy logic adapted kalman filter," *Control Engineering Practice*, vol. 7, no. 6, pp. 763–771, 1999.

[4] J. Sasiadek, Q. Wang, and M. Zeremba, "Fuzzy adaptive kalman filter- ing for ins/gps data fusion," in *Intelligent Control, 2000. Proceedings of the 2000 IEEE International Symposium on*. IEEE, 2000, pp. 181–186.

[5] A. Chatterjee and F. Matsuno, "Improving ekf-based solutions for slam problems in mobile robots employing neuro-fuzzy supervision," in *Intelligent Systems, 2006 3rd International IEEE Conference on*. IEEE, 2006, pp. 683–689.

[6] R. Havangi, M. A. Nekoui, and M. Teshnehlab, "Adaptive neuro-fuzzy extended kaiman filtering for robot localization," in *Power Electronics and Motion Control Conference (EPE/PEMC), 2010 14th International*. IEEE, 2010, pp. T5–130.

[7] X. Jiang, T. Li, and Y. Yu, "A novel slam algorithm with adaptive kalman filter," in *Advanced Robotics and Mechatronics (ICARM), International Conference on*. IEEE, 2016, pp. 107–111.

[8] D.-J. Jwo and S.-C. Chang, "Application of optimization technique for gps navigation kalman filter adaptation," *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, pp. 227–234, 2008.

[9] R. Mehra, "On the identification of variances and adaptive kalman filtering," *IEEE Transactions on automatic control*, vol. 15, no. 2, pp. 175–184, 1970.

[10] A. Chatterjee and F. Matsuno, "A neuro-fuzzy assisted extended kalman filter-based approach for simultaneous localization and map-ping (slam) problems," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 5, pp. 984–997, 2007.

[11] R. Fitzgerald, "Divergence of the kalman filter," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 736–747, 1971.

[12] M. S. Grewal and A. P. Andrews, "Kalman filtering. theory and practice prentice hall," *Englewood Cliffs, NJ*, 1993.

[13] T. H. Cong, Y. J. Kim, and M.-T. Lim, "Hybrid extended kalman filter-based localization with a highly accurate odometry model of a mobile robot," in *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*. IEEE, 2008, pp. 738–743.

[14] P. Koprinkova-Hristova and K. Alexiev, "Neuro-fuzzy tuning of kalman filter," in *Intelligent Systems (IS), 2016 IEEE 8th International Conference on*. IEEE, 2016, pp. 651–657.

[15] A. Chatterjee and F. Matsuno, "A geese pso tuned fuzzy supervisor for ekf based solutions of simultaneous localization and mapping (slam) problems in mobile robots," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5542–5548, 2010.

[16] H. Wang, W. Liu, F. Zhang, S. X. Yang, and L. Zhang, "A ga-fuzzy logic based extended kalman filter for mobile robot localization," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on*. IEEE, 2015, pp. 319–323.

Cong Hung Do receives a master degree from Department of Electrical Engineering, National Chung Cheng University, Taiwan, in 2017. He graduated from Technology and Education University, Vietnam, and majored in mechatronics. His research interests include robotics, mechatronics, computer vision, and specifically motion planning with neural-fuzzy techniques. He is currently working on manufacturing systems and optimization problems in Vietnam.

Huei-Yung Lin received his BS degree in Applied Mathematics from National Chiao Tung University, Taiwan, and his MS and PhD degrees in electrical and computer engineering from the State University of New York at Stony Brook. In 2002 he joined the Department of Electrical Engineering, National Chung Cheng University, Taiwan, and currently is a full professor. His current research interests include computer vision, robotics, pattern recognition, and image processing. He serves as an organizing committee member of several robotics related IEEE international conference. He is also a founding member and deputy secretary general of Taiwan Society of Robotics.

Yi-Chun Huang is currently a master student at Department of Electrical Engineering, National Chung Cheng University, Taiwan. Her research interests include multi-robots exploration, autonomous cleaning robot, mobile robot navigation in dynamic environment, simultaneous localization and mapping (SLAM), and complete coverage path planning (CCPP). She is working on complete coverage with efficient cost function on multi-robots, and creating a new strategy for multi-robots to find optimal paths for intelligent movement.

Self-Balance Control of a Two-Wheeled Robot Using Polynomial Fuzzy Systems

Gwo-Ruey Yu, *Member, IEEE* and Hsiang-Ting Huang

***Abstract*—This study proposes a Takagi-Sugeno (T-S) fuzzy control and a polynomial fuzzy governance to achieve the self-balance regulation of a two-wheeled robot. In the polynomial fuzzy model, the state can exist in the system matrix, whereas the T-S fuzzy model can only support the constant in the state matrix. Thus, the polynomial fuzzy system has a greater control gain range than the T-S fuzzy system does. This means that the polynomial fuzzy system has a greater chance of finding the optimal control gain. The polynomial fuzzy system is therefore not only capable of addressing the shortcomings of the T-S fuzzy system, but also of increasing the efficiency of the fuzzy controller. The paper then uses linear matrix inequality (LMI) stability conditions and the sum of square (SOS) way in the robot model to obtain the parallel distributed compensation (PDC) control gain. Finally, computer simulations show that the performance of the proposed polynomial fuzzy control system is much better than that of the existing T-S fuzzy control system.**

Keywords—Two-wheeled robot, polynomial fuzzy systems, self-balance control.

I. INTRODUCTION

Robots are used in a vast variety of applications, and constitute a safer or more efficient substitute for human labor. Today the operations of many processing and production industries such as the computer, motor vehicle and household appliance industries, are already almost completely automated. As populations continue to grow, robots may play an even greater part in our daily lives. Robots can help in almost any task, ranging from help with housework to public transportation. Many countries are now spending more money on this field in order to achieve a more complete and convenient lifestyle. The development and research of intelligent robots has therefore received significant attention recently. An inverted pendulum, which is a nonlinear unstable system in nature, is an ideal experiment platform for control theory education, and can be used as a benchmark for testing control strategies [1-3]. Many abstract control concepts, such as system stability, system controllability, system convergence speed, system disturbance rejection ability, etc., can be observed and tested with the inverted pendulum.

Recently, more and more research has begun to focus on two-wheeled robots which are a derivative of the inverted pendulum. The most well-known example is the Segway. Segway is an electrically powered vehicle that is able to control its equilibrium. This self-balancing ability is based on the theory of dynamic stabilization. With its built-in Solid-State Gyroscopes, the Segway is able to recognize the pose in which it has been set. Then, its high-speed central micro processing unit delivers instructions on how best to maintain balance. In literature [4], a partial feedback linearization control method was used to regulate a two-wheeled robot. However, the method is limited to a specific region, and has less robust. In literature [5] and [6] third coordinates were taken into consideration to deduce a 3-D dynamic equation for a two-wheeled robot, and remote control for a two-wheeled robot was developed. However, these studies did not consider the full nonlinear system. Transient response is difficult to predict, especially when the robot is not moving on a smooth surface. In literature [7], the sliding method was used to control a two-wheeled robot. In literature [8] an adaptive neural network regulation for the stability of a two-wheeled robot was developed. However, there were too many parameters in this method to define, and the algorithm was too complicated. Based on the above advantages and disadvantages, this paper proposes an improved fuzzy regulation manner.

In control theory, the analysis of the control and stability of nonlinear systems has always been a complicated, difficult problem. However, since Takagi and Sugeno described the relationship between the input and output of the T-S fuzzy model [9-10], the description of nonlinear systems, the design of control facilities and even the analysis of stability have all been simplified. Over the past few years, many successful T-S fuzzy regulation applications have been proposed. In literature [11], T-S fuzzy control facility was applied to achieve synchronization and secure communication in a chaotic system. In literature [12] it was used to derive the stability of multiple time delays, and was applied to a passive tuned mass damper. In literature [13], network control was combined with T-S fuzzy system in order to refine time delay problem in the network. Thus, using T-S fuzzy system to describe nonlinear systems, and the design of T-S fuzzy controller is an established research field. However in terms of finding gains, the stability conditions that T-S fuzzy system takes are too rigorous, and a constant matrix can only be allowed in the consequent part of the T-S fuzzy structure. Therefore, the literatures [14-15] proposed solutions to this problem; [14] used a regional membership function shape independent analysis way to make the LMI-based stabilization criteria more loosened, while [15] derived the

Gwo-Ruey Yu is with Department of Electrical Engineering and Advanced Institute of Manufacturing with High-tech Innovations, National Chung Cheng University, Chia-Yi, Taiwan (e-mail: ieewoyu@ccu.edu.tw).

Hsing-Ting Huang is with Graduate Institute of Opto-Mechatronics, National Chung Cheng University, Chia-Yi, Taiwan.

discrete relaxed stability conditions using the piecewise Lyapunov function and a switching T-S fuzzy model.

From the above T-S fuzzy control systems, the polynomial fuzzy control system was developed [16-18]. Polynomial fuzzy system allows state to exist in the consequent part of the system, and find better opportunities to control gains [19]. The literature [18] introduced the Taylor-series-based polynomial fuzzy models. The literature [22] used polynomial output-feedback control to discuss the stability of nonlinear systems. According to previous literatures, polynomial fuzzy systems are not only able to address the disadvantages of T-S fuzzy systems, but are also able to ensure that the control facility is more effective and general. Thus, this paper will use a polynomial fuzzy control system and polynomial modeling. The main contributions of this paper can be summarized as follows: 1) The polynomial fuzzy controller is first time designed for self-balance regulation of a two-wheeled robot on the basis of polynomial fuzzy model. 2) As a result, the control gain range limit is significantly decreased, and the stability criteria are more relaxed. With fewer conditions to consider, there is a better chance of finding a good gain in order to address the problems of the two-wheeled robot, such as an accurate nonlinear system not being considered during modeling, the algorithm being too complicated, etc. 3) In the simulation results, we can clearly show that our polynomial fuzzy control system exhibits better performance than that of the existing T-S fuzzy control system.

This paper presents the polynomial fuzzy controller of a two-wheeled robot and is arranged as follows. Section 2 considers the dynamics of the two-wheeled robot. Section 3 gives the presented sketch for the polynomial fuzzy model, controllers and stability conditions using a SOS approach. Section 4 shows the simulation result of T-S fuzzy system and polynomial system. Finally, Section 5 offers the conclusion.

II. Dynamics of Two-wheeled Robot

A. Two-Wheeled Robot Model

Fig. 1 shows the diagrammatic sketch of two-wheeled robot.

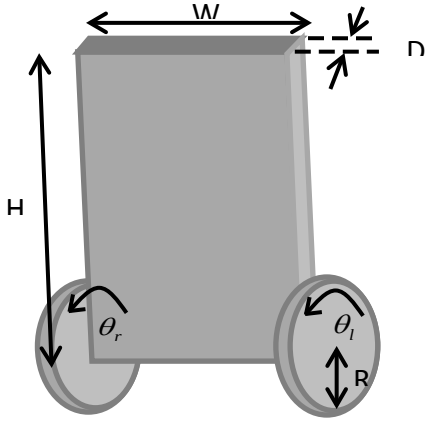


Fig. 1 Two-wheeled robot

Fig. 2 shows the side view of two-wheeled robot. The plane view of the two-wheeled robot is shown in Fig. 3. Table 1 lists the parameters of two-wheeled robot.

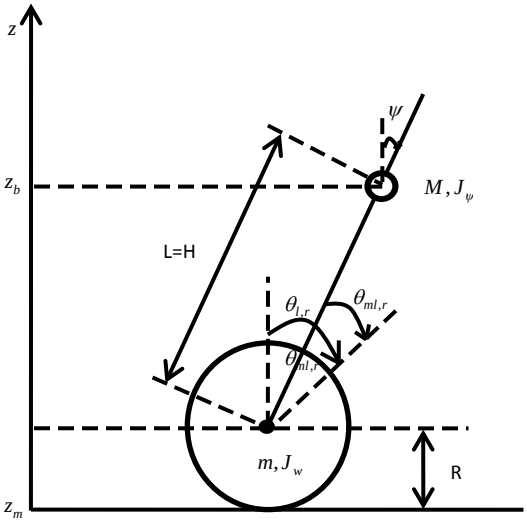


Fig. 2 Side view of two-wheeled robot

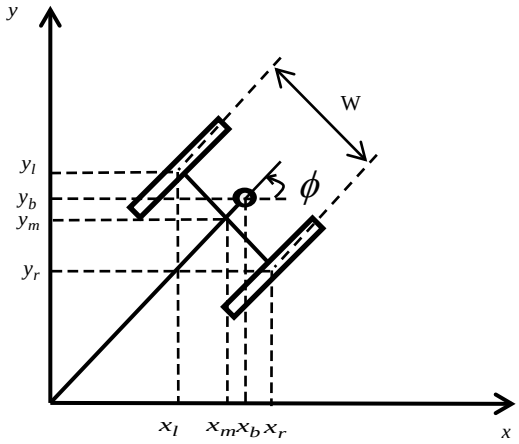


Fig. 3 Plane view of two-wheeled robot

Table 1 the parameters of two-wheeled robot

Gravity accleration	G=9.81 [$m/\sec^2$]
Body weight	M=0.6 [kgm^2]
Body width	W=0.14 [m]
Body depth	D=0.04 [m]
Body height	H=0.144 [m]
Body pitch inertia moment	$J_\psi = ML^2/3$ [kgm^2]
Body yaw inertia moment	$J_\phi = M(W^2 + D^2)/12$ [kgm^2]
Distance of the center of mass from the wheel axle	L=H/2 [m]
Wheel weight	M=0.03 [kg]
Wheel radius	R=0.04 [m]

Wheel inertia moment	$J_w = mR^2/2$ [kgm^2]
DC motor inertia moment	$J_m = 1\times10^{-5}$ [kgm^2]
DC motor resistance	$R_m = 6.69$ [Ω]
DC motor back EMF constant	$K_b = 0.468$ [$V\,\mathrm{sec}/rad$]
DC motor torque cinstant	$K_t = 0.317$ [Nm/A]
Gear ratio	n=1
Friction coefficient between body and DC motor	$f_m = 0.0022$
Friction coefficient between wheel and floor	$f_W = 0$

B. Equation of Motion of Two-Wheeled Robot Model

By using the Lagrangian approach based on the coordinate in Fig. 2 and Fig. 3, we can derive motion equation of two-wheeled robot as follows. We assume K_1 is the translation kinetic energy, K_2 is the rotation kinetic energy, and U is the potential energy, and the equations are as the following:

$$K_1 = \frac{1}{2}m\left(\dot{x}_l^{\,2} + \dot{y}_l^{\,2} + \dot{z}_l^{\,2}\right) + \frac{1}{2}m\left(\dot{x}_r^{\,2} + \dot{y}_r^{\,2} + \dot{z}_r^{\,2}\right) + \frac{1}{2}M\left(\dot{x}_b^{\,2} + \dot{y}_b^{\,2} + \dot{z}_b^{\,2}\right) \tag{1}$$

$$K_2 = \frac{1}{2}J_w\dot{\theta}_l^2 + \frac{1}{2}J_w\dot{\theta}_r^2 + \frac{1}{2}J_\psi\dot{\psi}^2 + \frac{1}{2}J_\phi\dot{\phi}^2 + \frac{1}{2}n^2J_m(\dot{\theta}_l - \dot{\psi})^2 + \frac{1}{2}n^2J_m(\dot{\theta}_r - \dot{\psi})^2 \tag{2}$$

$$U = mgz_l + mgz_r + mgz_b \tag{3}$$

where the fifth and sixth of K_2 are the rotation kinetic energy of an armature in the left and right of the DC motor. Then, we choose θ to be the average of the left and right wheel, ψ to be the body pitch angle and ϕ to be the body yaw angle as the generalized coordinates. The Lagrangian L can be represented as follows:

$$L = K_1 + K_2 - U \tag{4}$$

Therefore, the Lagrang equations are as follows:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\theta}}\right) - \frac{\partial L}{\partial\theta} = F_\theta \tag{5}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\psi}}\right) - \frac{\partial L}{\partial\psi} = F_\psi \tag{6}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\phi}}\right) - \frac{\partial L}{\partial\phi} = F_\phi \tag{7}$$

Using (4), (5)-(7) can be derived as follows:

$$\left[(2m+M)R^2 + 2J_w + 2n^2J_m\right]\ddot{\theta} + \left(MLR\cos\psi - 2n^2J_m\right)\ddot{\psi} - MLR\dot{\psi}^2\sin\psi = F_\theta \tag{8}$$

$$\left(MLR\cos\psi - 2n^2J_m\right)\ddot{\theta} - \left(ML^2 + J_\psi + 2n^2J_m\right)\ddot{\psi} - MgL\sin\psi_\psi - ML^2\dot{\phi}^2\sin\psi\cos\psi = F \tag{9}$$

$$\left[\frac{1}{2}mW^2 + J_\phi + \frac{W^2}{2R^2}(J_w + n^2J_m) + ML^2\sin^2\psi\right]\ddot{\phi} + 2ML^2\dot{\psi}\dot{\phi}\sin\psi\cos\psi = F_\phi \tag{10}$$

The dynamic equation of the DC motor is as follows:

$$L_m\dot{i}_{l,r} = v_{l,r} + K_b(\dot{\psi} - \dot{\theta}_{l,r}) - R_m i_{l,r} \tag{11}$$

The motor inductance is approximate to zero. Therefore, the current can be described as follows:

$$i_{l,r} = \frac{v_{l,r} + K_b(\dot{\psi} - \dot{\theta}_{l,r})}{R_m} \tag{12}$$

Using (12), the force can be represented as follows:

$$F_\theta = \alpha(v_l + v_r) - 2(\beta + f_w)\dot{\theta} + 2\beta\dot{\psi} \tag{13}$$

$$F_\psi = -\alpha(v_l + v_r) + 2\beta\dot{\theta} - 2\beta\dot{\psi} \tag{14}$$

$$F_\phi = \frac{W}{2R}\alpha(v_l - v_r) - \frac{W^2}{2R^2}(\beta + f_w)\dot{\phi} \tag{15}$$

where $\alpha = \dfrac{nK_t}{R_m}$, $\beta = \dfrac{nK_tK_b}{R_m}$.

III. SOS-Based Stability Analysis

This section presents the polynomial fuzzy model and controller using the SOS approach. LMI-based design approaches have reported many successful examples, and have recently developed in prevalence. However, there are still many challenges and issues that need to be addressed. Specifically, in the procedures of finding gains, the stability criteria that the T-S fuzzy system uses are very rigorous, and only a linear constant matrix can be allowed in the consequent part of the T-S fuzzy structure. Therefore, a novel group of fuzzy model has been grown to solve these problems. Polynomial fuzzy model is able to use fewer commands than T-S fuzzy model. In polynomial fuzzy systems, the polynomial fuzzy modeling framework is more widespread and effective than that of T-S fuzzy modeling.

A. Polynomial Fuzzy Model

Model Rule *i:*

IF $z_1(t)$ is M_{i1} and … and $z_p(t)$ is M_{ip}

THEN $\dot{x}(t) = \mathbf{A}_i(x(t))\hat{x}(x(t)) + \mathbf{B}_i(x(t))u(t)$, (16)

where $\mathbf{A}_i(x(t)) \in R^{n\times n}$ and $\mathbf{B}_i(x(t)) \in R^{n\times n}$ are polynomial matrices in $x(t)$. $\hat{x}(x(t))$ is a column vector whose entries

are all monomials in $x(t)$. The defuzzification of the polynomial fuzzy model (16) can be constructed shown below:

$$\dot{x}(t) = \frac{\sum_{i=1}^{r} w(z(t))\{\mathbf{A}_i(x(t))\hat{x}(x(t)) + \mathbf{B}_i(x(t))u(t)\}}{\sum_{i=1}^{r} w(z(t))}$$

$$= \sum_{i=1}^{r} h_i(z(t))\{\mathbf{A}_i(x(t))\hat{x}(x(t)) + \mathbf{B}_i(x(t))u(t)\} \qquad (17)$$

where

$z(t) = [z_1(t) \ldots z_p(t)]$, $\sum_{i=1}^{r} h_i(z(t)) = 1$, i=1,…,r for all t.

If $\hat{x}(x(t)) = x(t)$, both $\mathbf{A}_i(x(t))$ and $\mathbf{B}_i(x(t))$ are matrices with constant elements, then (17) can be simplified to a linear state-space equation. Therefore, the T-S fuzzy model is a particular situation of the polynomial fuzzy model.

B. Polynomial Fuzzy Modeling of Two-Wheeled Robot

In this section, we present the polynomial fuzzy model of the two-wheeled robot. Let ψ be the premise variables of the two-wheeled robot and the state equation of two–wheeled robot can be represented as follows:

$$N\begin{bmatrix}\ddot{\theta}\\ \ddot{\psi}\end{bmatrix} + O\begin{bmatrix}\dot{\theta}\\ \dot{\psi}\end{bmatrix} + P\begin{bmatrix}\theta\\ \psi\end{bmatrix} = Q\begin{bmatrix}v_l\\ v_r\end{bmatrix} \qquad (18)$$

where

$$N = \begin{bmatrix}(2m+M)R^2 + 2J_w + 2n^2J_m & MLR\cos\psi - 2n^2J_m\\ MLR\cos\psi - 2n^2J_m & ML^2 + J_\psi + 2n^2J_m\end{bmatrix},$$

$$O = \begin{bmatrix}2(\beta + f_w) & -2\beta + MLR\dot{\psi}\sin\psi\\ -2\beta & 2\beta\end{bmatrix},$$

$$P = \begin{bmatrix}0 & 0\\ 0 & \dfrac{-MgL\sin\psi - ML^2\dot{\phi}^2\sin\psi\cos\psi}{\psi}\end{bmatrix},$$

$$Q = \begin{bmatrix}\alpha & \alpha\\ -\alpha & -\alpha\end{bmatrix}.$$

Therefore, the state equation can be expanded and 4×4 matrix and the polynomial fuzzy model rules of the two-wheeled robot can be presented as follows:

Model Rule1:

IF ψ approach to $-\frac{\pi}{2}$,

Then $\dot{x}(t) = \mathbf{A}_1(x(t))x(t) + \mathbf{B}_1(x(t))u(t)$.

Model Rule2:

IF ψ approach to 0,

Then $\dot{x}(t) = \mathbf{A}_2(x(t))x(t) + \mathbf{B}_2(x(t))u(t)$.

Model Rule3:

IF ψ approach to $\frac{\pi}{2}$,

Then $\dot{x}(t) = \mathbf{A}_3(x(t))x(t) + \mathbf{B}_3(x(t))u(t)$.

where

$$x(t) = \begin{bmatrix}\theta\\ \psi\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}, \text{ and } u(t) = \begin{bmatrix}v_l\\ v_r\end{bmatrix}$$

The subsystem matrices of SOS approach are as follows:

$$A_1 = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & -73.8167 & -63.3686 & 63.3686 - 0.364*\dot{\psi}\\ 0 & 118.5285 & 32.1068 & -32.1068 - 0.5844*\dot{\psi}\end{bmatrix},$$

$$B_1 = \begin{bmatrix}0 & 0\\ 0 & 0\\ 61.5912 & 61.5912\\ -31.2062 & -31.2062\end{bmatrix}$$

$$A_2 = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & -409.7184 & -162.1273 & 162.1273\\ 0 & 269.6273 & 78.1496 & -78.1496\end{bmatrix},$$

$$B_2 = \begin{bmatrix}0 & 0\\ 0 & 0\\ 157.5798 & 157.5798\\ -75.9576 & -75.9576\end{bmatrix}$$

$$A_3 = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & -73.8167 & -63.3686 & 63.3686 - 0.364*\dot{\psi}\\ 0 & 118.5285 & 32.1068 & -32.1068 - 0.5844*\dot{\psi}\end{bmatrix},$$

$$B_3 = \begin{bmatrix}0 & 0\\ 0 & 0\\ 61.5912 & 61.5912\\ -31.2062 & -31.2062\end{bmatrix}$$

The membership functions are shown as the following:

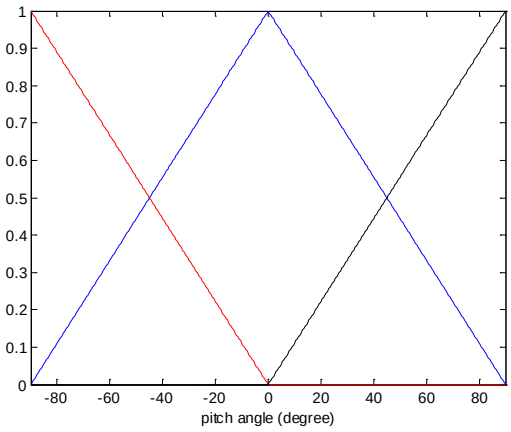


Fig. 4 Membership function of the degree of the pitch angle.

C. Polynomial Fuzzy Controller

As T-S fuzzy regulator, polynomial fuzzy regulator is arranged via PDC manner. The membership function is the same as polynomial fuzzy model. The consequent of the representation is the only different part. The polynomial fuzzy regulator is shown below:

Control Rule i :

IF $z_1(t)$ is M_{i1} and … and $z_p(t)$ is M_{ip}

THEN $u(t) = -\mathbf{F}_i(x(t))\hat{x}(x(t))$, $i = 1,2,...,r.$ (19)

The defuzzification process is shown as follows:

$$u(t) = -\sum_{i=1}^{r} h_i(z(t))\mathbf{F}_i(x(t))\hat{x}(x(t)) \tag{20}$$

From (17) and (20), the overall fuzzy model can be depicted as:

$$\dot{x}(t) = \sum_{i=1}^{r}\sum_{j=1}^{r} h_i(z(t))h_j(z(t)) \times\{\mathbf{A}_i(x(t)) - \mathbf{B}_i(x(t))\mathbf{F}_j(x(t))\}\hat{x}(t) \tag{21}$$

D. SOS-based Stability Criteria with Decay rate

Theorem 1: Let $\mathbf{D}(x)$ be a diagonal positive polynomial matrix. The feedback regulation system (21) is steady if a symmetric polynomial matrix $\mathbf{K}(\tilde{x}) \in \Re^{N\times N}$ and a polynomial matrix $\mathbf{M}_i(x) \in \Re^{m\times N}$ exist, such that the SOSs (22) and (23) are feasible, where for all x :

$$s^T(\mathbf{K}(\tilde{x}) - \sigma(x)\mathbf{I})s \text{ is SOS} \tag{22}$$

$$-s^T\left(\begin{bmatrix} \Gamma(x) & \mathbf{K}(\tilde{x})\mathbf{D}^T(x) \\ \mathbf{D}(x)\mathbf{K}(\tilde{x}) & -\mathbf{K}(\tilde{x}) \end{bmatrix} - \delta_{ij}(x)I\right)s \text{ is SOS},$$

$$i \le j \tag{23}$$

where

$$\Gamma(x) = \mathbf{H}(x)\mathbf{A}_i(x)\mathbf{K}(\tilde{x}) - \mathbf{H}(x)\mathbf{B}_i(x)\mathbf{M}_j(x) + \mathbf{K}(\tilde{x})\mathbf{A}_i^T(x)\mathbf{H}^T(x) - \mathbf{M}_j^T(x)\mathbf{B}_i^T(x)\mathbf{H}^T(x) - \sum_{k\in K}\frac{\partial\mathbf{K}(\tilde{x})}{\partial x_k}\mathbf{A}_i^k(x)\hat{x}(x) - \sum_{k\in K}\frac{\partial\mathbf{K}(\tilde{x})}{\partial x_k}\mathbf{A}_j^k(x)\hat{x}(x),$$

$s \in \Re^N$ is vector that is autonomous of x. $\mathbf{H}(x) \in \Re^{N\times N}$ is a polynomial matrix of which the (i,j)th element is clarified as

$$\mathbf{H}^{ij}(x) = \frac{\partial\hat{x}_i(x)}{\partial x_j} \tag{24}$$

A feedback control gain $\mathbf{F}_i(x)$ can be acquired from $\mathbf{K}(\tilde{x})$ and $\mathbf{M}_i(x)$ as

$$\mathbf{F}_i(x) = \mathbf{M}_i(x)\mathbf{K}^{-1}(\tilde{x}) \tag{25}$$

Proof:
Ponder a nominee of Lyapunov function

$$\mathbf{V}(x) = \hat{x}^T(x)\mathbf{K}^{-1}(\tilde{x})\hat{x}(x) \tag{26}$$

where $\mathbf{K}^{-1}(\tilde{x}) \in R^{N\times N}$ is a symmetric polynomial matrix.

$\boldsymbol{V}(x)$ is a positive definite function of x and the criterion (22) means that either $\mathbf{K}^{-1}(\tilde{x})$ or $\mathbf{K}(\tilde{x})$ are positive definite for each x, and Therefore, the stability criterion is presented by

$$\dot{\mathbf{V}}(x) = \hat{x}^T(x)\mathbf{K}^{-1}(\tilde{x})\mathbf{H}(x)\dot{x}(x) + \dot{x}^T(x)\mathbf{H}^T(x)\mathbf{K}^{-1}(\tilde{x})\hat{x}(x) + \hat{x}^T(x)\left(\sum_{k=1}^{n}\frac{\partial\mathbf{K}^{-1}(\tilde{x})}{\partial x_k}\dot{x}_k\right)\hat{x}(x) \tag{27}$$

Since $\mathbf{B}_i^k(x) = 0$ for $k \in \mathbf{K}$, we obtain

$$\dot{x}_k = \sum_{i=1}^{r} h_i(z)\mathbf{A}_i^k(x)\hat{x}(x) \text{ for } k \in \mathbf{K} \tag{28}$$

On the other hand

$$\frac{\partial\mathbf{K}^{-1}(\tilde{x})}{\partial x_k} = 0 \text{ for } k \notin \mathbf{K} \tag{29}$$

From (21), (28) and (29), (27) becomes

$$\dot{\mathbf{V}}(x) = \sum_{i=1}^{r}\sum_{j=1}^{r} h_i(z(t))h_j(z(t))\hat{x}^T(x) \times\left\{\boldsymbol{\Omega}_{ij}(x) + \sum_{k\in\mathbf{K}}\frac{\partial K^{-1}(\tilde{x})}{\partial x_k}\mathbf{A}_i^k(x)\hat{x}(x)\right\}\hat{x}(x) \tag{30}$$

where $\boldsymbol{\Omega}_{ij}$ is $\mathbf{K}^{-1}(\tilde{x})\mathbf{H}(x)\left[\mathbf{A}_i(x) - \mathbf{B}_i(x)\mathbf{F}_j(x)\right] + \left[\mathbf{A}(x) - \mathbf{B}_i(x)\mathbf{F}_j(x)\right]^T\mathbf{H}^T(x)\mathbf{K}^{-1}(\tilde{x})$.

By applying Schur complement to matrix (23) and multiplying left and right sides by $\mathbf{K}^{-1}(\tilde{x})$. We obtain

$$-\{\mathbf{\Omega}_{ij}(x)+\mathbf{\Omega}_{ji}(x)-\sum_{k\in K}\mathbf{K}^{-1}(x)\frac{\partial\mathbf{K}(\tilde{x})}{\partial x_k}\mathbf{A}_i^k(x)\hat{x}(x)\mathbf{K}^{-1}(x)$$
$$-\sum_{k\in K}\mathbf{K}^{-1}(x)\frac{\partial\mathbf{K}(\tilde{x})}{\partial x_k}\mathbf{A}_j^k(x)\hat{x}(x)\mathbf{K}^{-1}(x)$$
$$-\mathbf{D}(x)\mathbf{K}^{-1}\mathbf{D}(x)\}\geq 0 \quad (31)$$

When the inverse of $\mathbf{K}(x)$ exists, we obtain $\mathbf{K}^{-1}(\tilde{x})\mathbf{K}(\tilde{x})=\mathbf{I}$, and making differentiation on both sides of respective x_k yields

$$\frac{\partial\mathbf{K}^{-1}(\tilde{x})}{\partial x_k}\mathbf{K}(\tilde{x})+\mathbf{K}^{-1}(\tilde{x})\frac{\partial\mathbf{K}(\tilde{x})}{\partial x_k}=0$$

The last expression can be rewritten as

$$\mathbf{K}^{-1}(\tilde{x})\frac{\partial\mathbf{K}^{-1}(\tilde{x})}{\partial x_k}\mathbf{K}^{-1}(\tilde{x})=-\frac{\partial\mathbf{K}^{-1}(\tilde{x})}{\partial x_k} \quad (32)$$

Substituting (32) into (31), we obtain

$$-\{\mathbf{\Omega}_{ij}(x)+\mathbf{\Omega}_{ji}(x)-\sum_{k\in K}\frac{\partial\mathbf{K}^{-1}(\tilde{x})}{\partial x_k}\mathbf{A}_i^k(x)\hat{x}(x)$$
$$-\sum_{k\in K}\frac{\partial\mathbf{K}^{-1}(\tilde{x})}{\partial x_k}\mathbf{A}_j^k(x)\hat{x}(x)$$
$$-\mathbf{D}(x)\mathbf{K}^{-1}(x)\mathbf{D}(x)\}\geq 0 \quad (33)$$

From (33), it follows that

$$\dot{\mathbf{V}}(x)\leq-\hat{x}^T(x)\mathbf{D}(x)\mathbf{K}^{-1}(x)\mathbf{D}(x)\hat{x}\ (x)$$
$$\leq-\beta\mathbf{V}(x)$$

where $\beta=-\lambda_{\min}(D(x)K^{-1}(x)D(x))/\lambda_{\max}(K^{-1}(x))$.

Furthermore, if $\delta_{ij}(x)>0$ for all *x*, then $\dot{V}(x)<0$ when $x\neq 0$.

IV. Results

To illustrate the advantages of the presented theorem, the LMI-based approach is utilized to the two–wheeled robot for comparison. This section is composed of four parts. Case 1 shows the simulation of the small pitch angle stability between the LMI approach and the SOS approach. Case 2 shows the simulation of the large pitch angle stability between the LMI approach and the SOS approach. Case 3 shows the different pitch angles via SOS approach. Case 4 shows the different initial values via SOS approach and LMI approach. The subsystem matrices of T-S fuzzy model are shown below:

The subsystem matrices of LMI approach

$$A_1=\begin{bmatrix}0&0&1&0\\0&0&0&1\\0&1.1521&-43.1688&43.1688\\0&64.7497&11.4917&-11.4917\end{bmatrix},$$

$$B_1=\begin{bmatrix}0&0\\0&0\\41.9580&41.9580\\-11.1694&-11.1694\end{bmatrix},$$

$$A_2=\begin{bmatrix}0&0&1&0\\0&0&0&1\\0&-409.7184&-162.1273&162.1273\\0&269.6273&78.1496&-78.1496\end{bmatrix},$$

$$B_2=\begin{bmatrix}0&0\\0&0\\157.5798&157.5798\\-75.9576&-75.9576\end{bmatrix},$$

$$A_3=\begin{bmatrix}0&0&1&0\\0&0&0&1\\0&1.1521&-43.1688&43.1688\\0&64.7497&11.4917&-11.4917\end{bmatrix},$$

$$B_3=\begin{bmatrix}0&0\\0&0\\41.9580&41.9580\\-11.1694&-11.1694\end{bmatrix}$$

Case 1

We use stability conditions (22) and (23) to acquire the polynomial fuzzy gains and the initial values are $\begin{bmatrix}0&5&0&0\end{bmatrix}$ degree. The fuzzy gains and the eigenvalues of Lyapunov matrix ***P*** are shown below:

Zero order polynomial fuzzy controller gains

$$F_1=\begin{bmatrix}-0.2844\times10^3&-3.1624\times10^3&-0.1326\times10^3&-0.3457\times10^3\\-0.2844\times10^3&-3.1624\times10^3&-0.1326\times10^3&-0.3457\times10^3\end{bmatrix},$$
$$F_2=\begin{bmatrix}-0.3958\times10^2&-4.4505\times10^2&-0.1892\times10^2&-0.4821\times10^2\\-0.3958\times10^2&-4.4505\times10^2&-0.1892\times10^2&-0.4821\times10^2\end{bmatrix},$$
$$F_3=\begin{bmatrix}-0.4169\times10^3&-4.6419\times10^3&-0.1944\times10^3&-0.5077\times10^3\\-0.4169\times10^3&-4.6419\times10^3&-0.1944\times10^3&-0.5077\times10^3\end{bmatrix},$$

The positive matrix ***P*** and the corresponding eigenvalues

$$P=\begin{bmatrix}0.0542\times10^7&0.0206\times10^7&0.4946\times10^7&0.0446\times10^7\\0.0206\times10^7&0.0079\times10^7&0.1888\times10^7&0.0170\times10^7\\0.4946\times10^7&0.1888\times10^7&4.5161\times10^7&0.4071\times10^7\\0.0446\times10^7&0.0170\times10^7&0.4071\times10^7&0.0369\times10^7\end{bmatrix}$$

$$eig(P)=\begin{bmatrix}8.3364\\40.0047\\254.7996\\4.6149\times10^7\end{bmatrix}$$

By using the LMI-based stability conditions [20], the T-S fuzzy regulation gains and the eigenvalues of Lyapunov matrix $\boldsymbol{P}$ are obtained shown below

T-S fuzzy regulator gains

$$F_1 = \begin{bmatrix} -17.5523 & -247.6300 & -7.8616 & -26.6587 \\ -17.5523 & -247.6300 & -7.8616 & -26.6587 \end{bmatrix},$$

$$F_2 = \begin{bmatrix} -36.6191 & -512.6513 & -15.8221 & -60.0243 \\ -36.6191 & -512.6513 & -15.8221 & -60.0243 \end{bmatrix},$$

$$F_3 = \begin{bmatrix} -17.5523 & -247.6300 & -7.8616 & -26.6587 \\ -17.5523 & -247.6300 & -7.8616 & -26.6587 \end{bmatrix}$$

The positive matrix $\boldsymbol{P}$ and the corresponding eigenvalues

$$P = \begin{bmatrix} 0.0198\times10^{10} & 0.2601\times10^{10} & 0.0080\times10^{10} & 0.0307\times10^{10} \\ 0.2601\times10^{10} & 3.5567\times10^{10} & 0.1081\times10^{10} & 0.4197\times10^{10} \\ 0.0080\times10^{10} & 0.1081\times10^{10} & 0.0033\times10^{10} & 0.0128\times10^{10} \\ 0.0307\times10^{10} & 0.4197\times10^{10} & 0.0128\times10^{10} & 0.0496\times10^{10} \end{bmatrix}$$

$$eig(P) = \begin{bmatrix} 8.3364 \\ 40.0047 \\ 8.7996\times10^{6} \\ 3.6286\times10^{10} \end{bmatrix}.$$

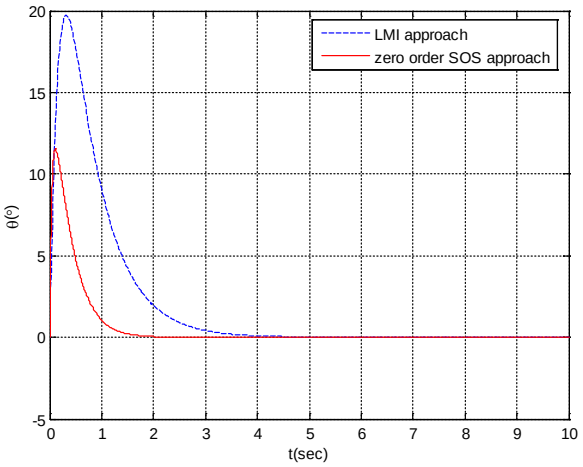


Fig. 5 The time response of θ via SOS and LMI.

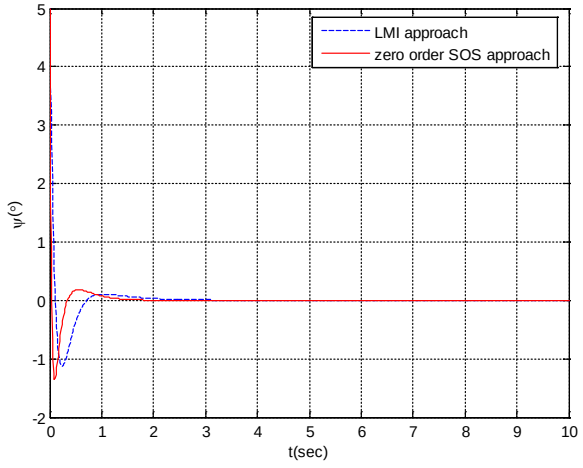


Fig. 6 The time response of ψ via SOS and LMI.

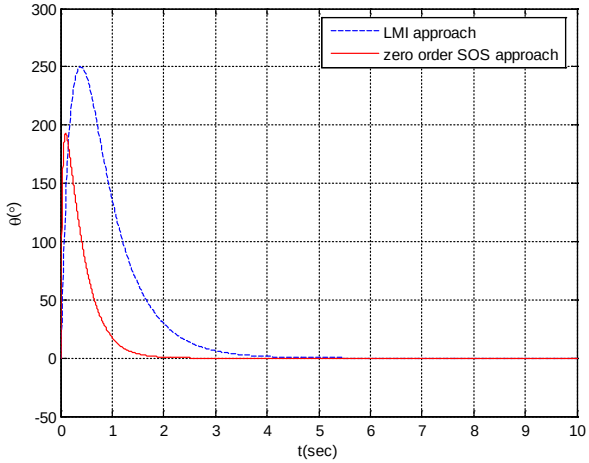


Fig. 7 The time response of θ in case 2.

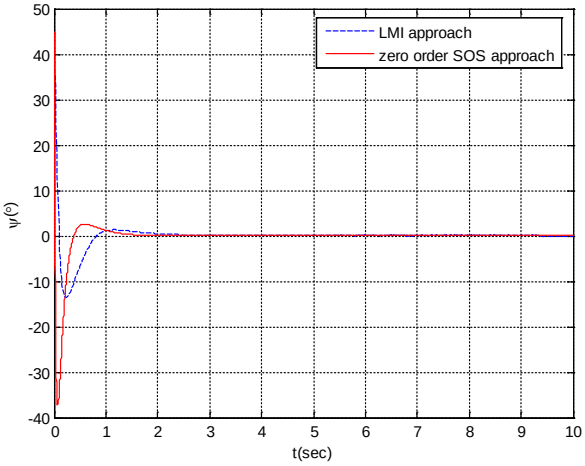


Fig. 8 The time response of ψ in case 2.

Fig. 5 and Fig. 6 display the waveform of system states via LMI and SOS way with a small angle of inclination, respectively. In this case, we can see that the convergence speed of the proposed SOS way is obviously quickly than that of the conventional LMI way. This proves that using polynomial fuzzy controller to control a small angle of inclination is more efficacious than using T-S fuzzy controller. Polynomial fuzzy systems can reduce the control rules and the stability conditions are also more relaxed. Therefore, we have more chance to find the optimize control gains.

Case 2

We use stability conditions (22) and (23) to acquire the polynomial fuzzy regulator gains and the initial values are $\begin{bmatrix} 0 & 45 & 0 & 0 \end{bmatrix}$ degree.

Fig. 7 and Fig. 8 show the waveform of system states via LMI and SOS approach with a large angle of inclination, respectively. In this case, we can see that the convergence speed of the SOS way is obviously higher than that of the LMI way. This proves that with either small or large angles of inclination, using polynomial fuzzy controller is more efficacious than using T-S fuzzy controller.

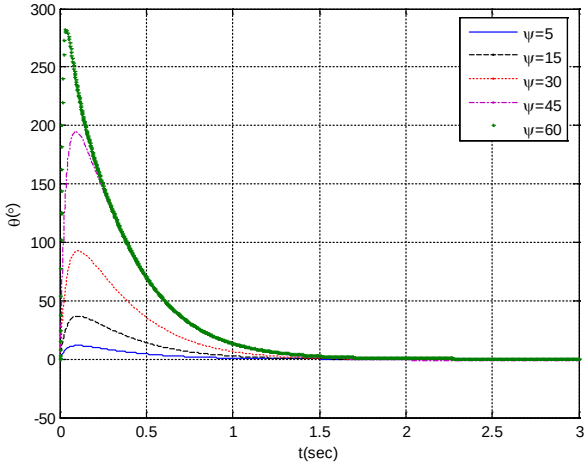


Fig. 9 The time response of θ in case 3.

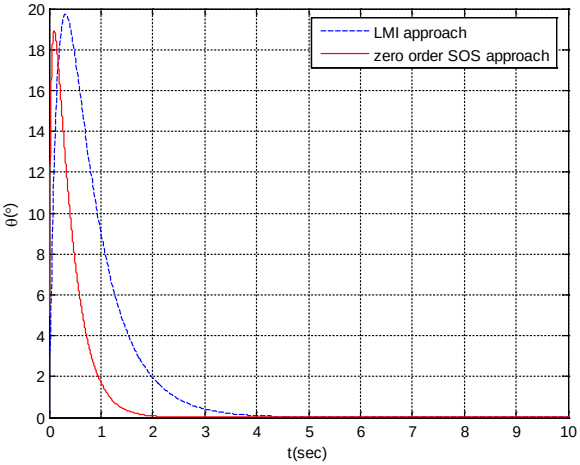


Fig. 11 The time response of θ in case 4.

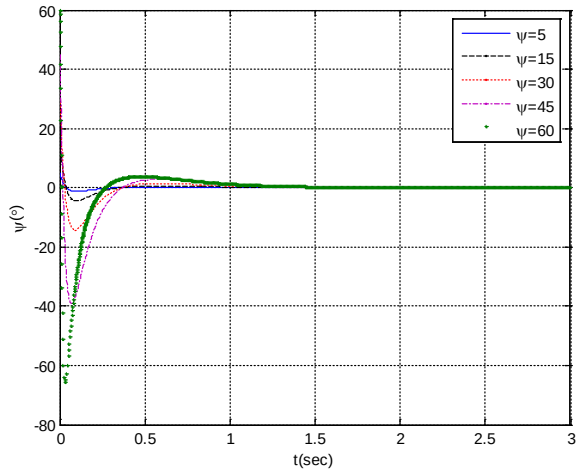


Fig. 10 The time response of ψ in case 3.

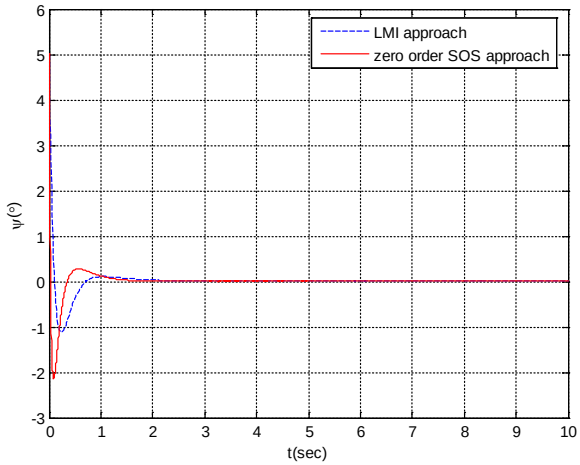


Fig. 12 The time response of ψ in case 4.

Case 3

We use stability conditions (22) and (23) to acquire the polynomial fuzzy regulator gains and the initial values are different degrees.

Fig. 9 and Fig. 10 show the waveform of system states with different initial values. In this case, we can see that no matter how large the angle of inclination is, the polynomial fuzzy controller has great control ability, and maintains system stability.

Case 4

We use stability conditions (22) and (23) to obtain the polynomial fuzzy controller gains and the initial values are $\begin{bmatrix}5 & 5 & 5 & 5\end{bmatrix}$ degree.

Fig. 11 and Fig. 12 show the waveform of system states. In this case, the polynomial fuzzy regulator has great control ability, and maintains system stability. The performance of the polynomial fuzzy regulator is better than that of the T-S fuzzy regulator. The polynomial fuzzy systems reduce control rules and the stability conditions are also relaxed such that the better control gains are obtained.

V. Conclusion

This paper presents a polynomial fuzzy control with decay rate for a two-wheeled robot. The polynomial fuzzy system not only overcame the disadvantages of the T-S fuzzy system, but also made the control facility more effective and general. Thus, the polynomial fuzzy control system was used to improve the problems facing a two-wheeled robot, and as a means to further discuss self-balance research. Using polynomial fuzzy system meant that the control gain limit range was significantly decreased, and the stability condition was also more relaxed. Therefore, we had a better chance of finding the optimal control gains. In the results, we see that no matter how big the angle of the inclination is, in all cases, the polynomial fuzzy controller had great control ability and maintained system stability; the polynomial fuzzy control system was superior in performance to the T-S fuzzy control system.

References

[1] Sangtae Kim and Sang Joo Kwon, “Nonlinear optimal control design for underactuated two-wheeled inverted pendulum mobile platform,”

IEEE/ASME Trans. Mechatronics., vol. 22, no. 6, pp.2803-2808, Dec. 2017.

[2] Kazuto Yokoyama and Masaki Takahashi, "Dynamics-based nonlinear acceleration control with energy shaping for a mobile inverted pendulum with a slider mechanism," *IEEE Trans. Control System Technology*, vol. 24, no. 1, pp.40-55, Jan. 2016.

[3] M. S. Park, and D. K. Chwa, "Swing-up and stabilization control of inverted-pendulum systems via coupled sliding-mode control method," *IEEE Trans. Ind. Electron.*, vol. 53, no. 2, pp.631-639, Sep. 2006.

[4] K. Pathak, J. Franch, and S. K. Agrawal, "Velocity and position controlof a wheel inverted pendulum by partial feedback linearization," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 505–513, Jun. 2005.

[5] Y. Kim, S. H. Kim, and Y. K. Kwak, "Dynamic analysis of a nonholo-nomic two-wheeled inverted pendulum robot,"*J. Intell. Robot. Syst.*,vol. 44, no. 1, pp. 25–46, Sep. 2005.

[6] F. Grasser, A. D'Arrigo, S. Colombi, and A. Rufer, "JOE: A mobile,inverted pendulum," *IEEE Trans. Ind. Electron.*, vol. 49, no. 1, pp. 107–114, Feb. 2002.

[7] J. Huang, Z. H. Guan, T. Matsuno, T. Fukuda and K. Sekiyama, "Sliding-mode velocity control of mobile-wheeled inverted-pendulum systems," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 750–758, Aug. 2010.

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

[9] W. J. Wang and W. W. Lin, "Decentralized PDC for large-scale T–S fuzzy systems,'' *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 6, pp. 779-786, Dec. 2005.

[10] T. S. Li and S. H. Tsai, "T-S Fuzzy bilinear model and fuzzy controller design for a class of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 3, pp. 494-506, Jun. 2007.

[11] K. Y. Lian, T. S. Chiang, C. S. Chiu and P. Liu, "Synpaper of fuzzy model-based designs to synchronization and secure communications for chaotic systems,'' *IEEE Trans. Syst. II*, vol. 31, no. 1, pp. 66-83, Feb. 2001.

[12] F. H. Hsiao, C.W Chen, Y. W Liang, S. D. Xu and W. L. Chiang, "T-S Fuzzy controllers for nonlinear interconnected systems with multiple time delays," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 9, pp.1883-1893, Sep. 2005.

[13] J. Zhang, P. Shi and Y. Q. X, "Fuzzy delay compensation control for T-S fuzzy systems over network," *IEEE Trans. Cyber*, vol. 43, no. 1, pp.259-268, Feb. 2013.

[14] M. Narimani and H. K. Lam, "Relaxed LMI-based stability conditions for Takagi–Sugeno fuzzy control systems using regional-membership-function-shape -dependent analysis approach," *IEEE Trans. Fuzzy Syst*, vol. 17, no. 5, pp.1221-1228, Oct 2009.

[15] W. J. Wang, Y. J. Chen, and C. H. Sun "Relaxed stabilization criteria for discrete-time T–S fuzzy control systems based on a switching fuzzy model and piecewise Lyapunov function," *IEEE Trans. Fuzzy Syst*, vol. 37, no. 3, pp.551-559, Jun 2007.

[16] W. H. Li and W. Q. Wang, "Guaranteed cost control for polynomial discrete fuzzy time delay systems by sum-of-squares approach," *Proc. IEEE International Conference on Information and Computing Science*, pp. 178-181, Liverpool, England, 2012.

[17] H.K. Lam and L.D. Seneviratne, "Stability analysis of polynomial fuzzy-model-based control systems under perfect/imperfect premise matching," *IET Control Theory & Appl.*, vol. 5, no. 15, pp. 1689-1697, Oct. 2011.

[18] A. Sala and C. Ari˜no, "Polynomial fuzzy models for nonlinear control: a Taylor-series approach," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 6, pp. 1284-1295, Dec. 2009.

[19] K. Tanaka, H. Yoshida, H. Ohtake, and H. O. Wang, "A sum-of-squares approach to modeling and control of nonlinear dynamical systems with polynomial fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp.911-922, Aug. 2009.

[20] K. Tanaka, and H. O. Wang, "*Fuzzy control systems design and analysis: a linear matrix inequality approach*," New York: Wiley, 2004.

[21] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo, *SOSTOOLS: Sum of Squares Optimization Toolbox for MATLAB Version 2.00*, California Inst. Technol., Pasadena, 2004.

[22] H.K. Lam, "Stabilization of nonlinear systems using sampled-data output-feedback fuzzy controller based on polynomial-fuzzy-model-based control approach," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 258–267, Feb. 2012.

Gwo-Ruey Yu received the Ph.D. degree in Electrical Engineering from the University of Southern California, Los Angeles, in 1997. He is currently a Professor of Electrical Engineering Department and the Director of Elegant Power Application Research Center, National Chung Cheng University, Taiwan. Dr. Yu is respectively the recipients of the Best Paper Award of IEEE 2017 International Automatic Control Conference, the Advisor Award of Robotic Society of Taiwan in 2017, the Best Paper Award in Application of IEEE 2016 International Conference on Fuzzy Theory and It's Applications, the Outstanding Paper Award of IEEE 2016 International Automatic Control Conference, the Honorable Mention Paper Award of IEEE 2016 International Conference on Advanced Robotics and Intelligent Systems. His research interests include intelligent robots, automatic control based on artificial intelligence, and renewable energy systems.

Hsiang-Ting Huang received the B.S. degree from National Kaohsiung Normal University, Kao-Hsiung, Taiwan, in 2010, and the M.S. degree in Graduate Institute of Opto-Mechatronics, National Chung Cheng University, Chia-Yi, Taiwan in 2013, respectively. She received the First Prize of the Best Paper Award of International Conference on Fuzzy Theory and Its Applications, in 2012. Her research interests include intelligent robots, fuzzy systems, and intelligent control.

Contouring Control of a 5-DOF Robot Arm for Machining

Woraphrut Kornmaneesang, Thanana Nuchkrua, and Shyh-Leh Chen

***Abstract*— This work is to design the contouring controller of a 5-DOF robot arm for machining. The kinematics is employed to generate the reference commands for each joint of the robot arm from the desired circular path. Then, the controller for contouring the desired path is designed based on the method of equivalent errors. For comparison, a distributed PID controller with feedforward function is also designed. The simulation and experimental results verify the effectiveness of the proposed contouring controller.**

I. Introduction

Due to the flexibility and large working space, robot manipulators have extensively played important roles in dealing with the automation tasks in manufacturing, such as carrying a large or heavy payload, painting, and assembly operations [1-4]. In fact, another important task in manufacturing, i.e., machining, can also be performed by a robot. This work represents a preliminary study of a long-term project, where a complete robotic machining system consisting of a dual-arm robot will be established. In this study, only a single robot arm is considered. In a machining task, controlling the robot arm to follow a desired path is the ultimate goal for the product quality in manufacturing [3, 4]. Thus, in order to achieve this aim, the error between the desired path and the actual path, called contour error, is the control objective to be minimized.

To deal with the contouring control problem, the cross-coupled control (CCC) was firstly proposed by Koren, and recognized as the control structure for a biaxial system [5]. Furthermore, Chiu and Tomizuka introduced the task coordinate frame approach (TCF), which focuses on the normal error dynamics in the task coordinate frame [6]. The diversity of dynamics among all axes is necessary to be considered in order to reduce the contour error. In this respect, Yeh and Hsu proposed a perfectly matched feedback control [7]. Considering the contouring control for five-axis machine tools, Yang and Altintas presented a contour error component compensation control method [8], in which the estimated contour errors are fed back to the desired commands of each axis with an appropriate proportional gain. On the other hand, the method of equivalent errors proposed by Chen and Wu transforms the system dynamics into the coordinates of equivalent errors [9]. In other words, the contouring problem is converted into the problem of stabilization. Thus, the design of integrated contouring controller becomes easier. This method can be easily implemented to the motion system with more than three axes.

This work will consider a robotic system with five degrees of freedom (DOF). The method of equivalent errors will be followed to design a contouring controller. For comparison, a distributed controller will also be designed, which aims at minimizing the tracking error of each joint response. The distributed controller considers each joint dynamics separately, and a proportional-integral-derivative controller (PID controller) with feedforward controller is designed. It will be shown that due to the strong nonlinearity in the robotics dynamics, the contouring performance of a distributed controller is limited. The controller designed by the method of equivalent errors can more effectively reduce the contouring error.

The rest of this paper is organized as follows. The kinematics and dynamics of the single robot arm system are briefly reviewed in Section II. Section III provides the contour error definition. Next, Section IV describes the design of the distributed controller for the contouring control. Then, the simulation and experimental results are presented in Section V. Finally, conclusions are drawn in Section VI.

II. System Description

As mentioned above, the complete robotic machining system is a dual-arm system. In this study, only one arm is considered and its contouring control problem is discussed. In order to well design the robot arm controller, the kinematics and dynamic equations of the robot arm system are the two key factors. The kinematics is necessary in order to generate the command references for servo drives from the desired path. On the other hand, the dynamic equations are required for the design of the contouring controller. Consequently, they are going to be discussed in this section.

A. Kinematics

Usually, a set of the desired displacements and orientations ξ of the end-effector with respects to the Cartesian coordinates are treated as the desired path for the manipulators. However, the controllers need to directly actuate the generalized or machine coordinates, called joint

This work was supported in part by the Ministry of Science and Technology, Taiwan, ROC, under Grant MOST 106-2218-E-194-011, MOST 107-2634-F-194-001, and MOST 106-2622-E-194-003 -CC2.

Woraphrut Kornmaneesang is with the Advanced Institute of Manufacturing with High-Tech Innovation, National Chung Cheng University, Taiwan (email: woraphrut.korn@gmail.com)

Thanana Nuchkrua was with the Advanced Institute of Manufacturing with High-Tech Innovation, National Chung Cheng University, Taiwan (email: thanana.nuch@yahoo.com)

Shyh-Leh Chen is with the Mechanical Engineering Department and the Advanced Institute of Manufacturing with High-Tech Innovation, National Chung Cheng University, Taiwan (email: imslc@ccu.edu.tw)

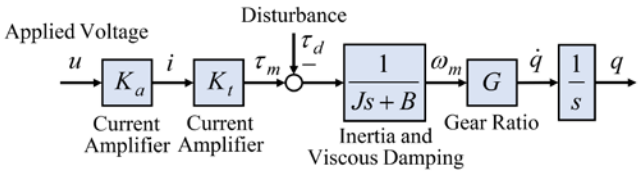


Figure 1. Block diagram of the single servo drive system

space q. Thus, the relationship between the displacements and orientations of the end-effector and the joint space is considered. It can be described by kinematics. Forward kinematics is the transformation from the joint space into the displacements and orientations of the end-effector as:

$$\xi = T(q), \tag{1}$$

where $T:\Re^n \Rightarrow \Re^n$ is the forward kinematic transformation, and *n* is the number of degrees of freedom. On the other hand, the joint space can be obtained from the inverse kinematics as:

$$q = T^{-1}(\xi). \tag{2}$$

B. Dynamics of a single robot arm system

Since the single robot arm is a serial manipulator, its dynamic equations can be derived by the Euler-Lagrange equations described in [10] and can be expressed as:

$$M(q)\ddot{q} + V(q,\dot{q}) + g(q) = \tau \tag{3}$$

where M is the inertia matrix of manipulators, V is a vector related to the centrifugal and Coriolis forces, g is a vector related to gravitational force, and τ is a vector of applied torques.

Each joint in the 5-DOF robot arm is driven by a servo motor, whose dynamics is shown in Fig. 1, and can be written in the form of differential equation as:

$$\frac{J}{GK_aK_t}\ddot{q}(t) + \frac{B}{GK_aK_t}\dot{q}(t) = u(t) - \frac{\tau_d(t)}{K_aK_t} \tag{4}$$

where J and B are the inertia and viscous damping coefficient, respectively; K_a and K_t are the current amplifier and torque constant gain, respectively; G is the gear ratio; u and τ_d are respectively the control input voltage and external disturbance including friction, backlash, etc. Since there are five servo motors, the subscript $i = 1,2,...,5$ is introduced. Also, let us define:

$$m_i = \frac{J_i}{G_iK_{ai}K_{ti}},\ c_i = \frac{B_i}{G_iK_{ai}K_{ti}},\ d_i(t) = \frac{\tau_{di}(t)}{K_{ai}K_{ti}} \tag{5}$$

Then, the dynamics of the i-th servo motor dynamics can be written as:

$$m_i\ddot{q}_i(t) + c_i\dot{q}_i(t) = u_i(t) - d_i(t) \tag{6}$$

III. CONTOURING CONTROLLER BY EQUIVALENT ERRORS

In case of the contouring control problem, the control objective is no longer only the tracking error e_t, but also the contour error ε_c which is defined as:

$$\varepsilon_c = \left\| \xi_a - \xi_n \right\|, \tag{7}$$

where ξ_a and ξ_n are the current actual point and its nearest point on the desired path, respectively, while the tracking error e_t is defined to be the error between the desired point ξ_d and the current actual point ξ_a, as illustrated in Fig. 2. For contouring control problem, it is the contour error that is to be minimized. Hence, a contouring controller will be designed in this work. The contouring controller will be designed using the method of equivalent errors [9]. For details about the contouring controller, please refer to [9].

The equivalent errors are defined by

$$\begin{bmatrix} \varepsilon(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} p(q) \\ \dot{q}_d^T(q - q_d) \end{bmatrix} \tag{7}$$

where $\varepsilon = p(q)$ is the equivalent contour error, and $e = \dot{q}_d^T(q - q_d)$ is the tangential error. Note here that $p:\Re^n \Rightarrow \Re^{n-1}$ is the function defining the desired path in the joint space. Then, by a twice differentiating the equivalent errors with respect to time, the dynamics of equivalent errors can be obtained as:

$$\begin{bmatrix} \ddot{\varepsilon} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} \Omega_1(q,\dot{q},t) \\ \Omega_2(q,\dot{q},t) \end{bmatrix} + \Gamma(q,\dot{q},t)u \tag{8}$$

where

$$\Omega_1(q,\dot{q},t) = h(q,\dot{q}) + \nabla p(q)(-M^{-1}(V+G))$$

$$\Omega_2(q,\dot{q},t) = \ddot{q}_d^T(q - q_d) + \ddot{q}_d^T(2\dot{q} - 3\dot{q}_d) + \dot{q}_d^T(-M^{-1}(V+G))$$

$$\Gamma(q,\dot{q},t) = \begin{bmatrix} \nabla p(q) \\ \dot{q}_d^T \end{bmatrix} M^{-1}$$

where $\nabla p(q) = \partial p(q)/\partial q$ and

$$h(q) = \begin{bmatrix} h_1 & h_2 & \cdots & h_{n-1} \end{bmatrix}$$

$$h_k = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 p_k(q)}{\partial q_i \partial q_j}\dot{q}_i\dot{q}_j, \quad 1 \le k \le n-1$$

Now, the problem of contouring control is converted into the stabilization of equivalent errors. For this purpose, we can employ the technique of feedback linearization [27], i.e., the control input is taken as:

$$u = \Gamma^{-1}(-\Omega + v) \tag{9}$$

where v is the new control input that is designed after linearizing the system. There are many control approaches to design the new control input. Here, the integral sliding mode control (ISMC) is used for robustness, i.e.,

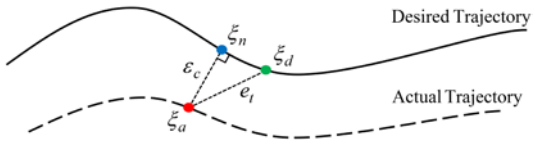


Figure 2. Contour and tracking errors

$$v = -\left[b_1\xi + b_2\eta + \rho \,\mathrm{sat}\left(\frac{\sigma}{\delta}\right)\right] \tag{10}$$

where $\eta = \begin{bmatrix}\varepsilon^T & e\end{bmatrix}^T$ and $\xi = \begin{bmatrix}\dot{\varepsilon}^T & \dot{e}\end{bmatrix}^T$ are equivalent errors and their time derivatives, $\sigma = \xi + b_1\eta + b_2\int \eta\, dt = 0$ represents the sliding manifold, and b_1, b_2, ρ, δ are positive control gains.

However, it is much easier to design a tracking controller that can minimize the tracking error. Many approaches of designing the tracking controller can be found in the literature. In this work, a distributed tracking controller is also designed for comparison with the contouring controller designed by the method of equivalent errors.

IV. Distributed Controller Design

In this work, the distributed controller will consist of two parts: feedback control and feedforward control. The feedback control is to guarantee the stability and transient performance of the overall system. On the other hand, the feedforward control is to reduce the effect of servo lags and improve the steady state errors.

A. Feedback Controller Design

To improve the frequency response of the controlled system, a cascade control configuration of position and velocity loops is exploited and illustrated in Fig. 3. A PID-type controller is adopted here due to its simplicity of implementation. In other words, a PI control will be implemented in the velocity loop, and a P control will be applied in the position loop, which is referred to as the P-PI control configuration [12].

Systematically designing the P-PI controller requires a transfer function of each servo drive, the input and output of which are the applied voltage u and the angular velocity $\dot{q}$, respectively. It can be derived from (6) as:

$$\frac{sQ_i(s)}{U_i(s)} = G_i(s) = \frac{1}{m_i s + c_i}. \tag{11}$$

The PI controller for the angular velocity control loop can be written as:

$$C_{\omega i}(s) = \frac{k_{Pi}s + k_{Ii}}{s}, \tag{12}$$

and the P controller for the angle control loop can be expressed as:

$$C_{\theta i}(s) = K_i, \tag{13}$$

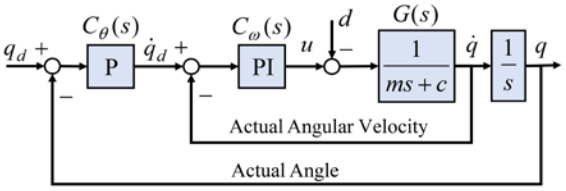


Figure 3. Block diagram of the P-PI feedback control structure

where k_P and k_I are the proportional and integral gains for the angular velocity control loop and K is the proportional gain for the position control loop, respectively.

To guarantee that the P-PI control structure works well, there exist two necessary conditions: one is that all poles of the controlled system must strictly locate in the left-hand side of the s-plane in order to assure the stability, and the other is that the bandwidth of the velocity loop must be larger than the position loop at least 3 times. Likewise, the bandwidth of the controlled system must cover the frequency of the reference at least three times. To accomplish the above considerations, the PID controller design techniques, such as root locus techniques [13], and frequency response techniques [14], are employed.

B. Feedforward Controller Design

Normally, an addition of feedforward control is used to enhance the control performance such as the tracking error. In view of the motion control, the dynamics of the servo drives induces itself servo lags that directly cause the contour and tracking error. Hence, in order to improve the servo lags, the feedforward controllers are considered to be integrated into both the position and velocity loops of the control structure illustrated in Fig. 4, from which the transfer function of the controlled system can be derived as:

$$\frac{Q_i(s)}{Q_{di}(s)} = \frac{\alpha_i v_i s^3 + \alpha_i k_{Pi} s^2 + (\alpha_i k_{Ii} + K_i k_{Pi})s + K_i k_{Pi}}{m_i s^3 + (c_i + k_{Pi})s^2 + (k_{Ii} + K_i k_{Pi})s + K_i k_{Pi}} \tag{14}$$

where α and v are the feedforward gains for the position and velocity loops, respectively. Usually, as compared with the proportional gain k_P, the viscous damping c is so tiny that the transfer function can be simplified as:

$$\frac{Q_i(s)}{Q_{di}(s)} = \frac{\alpha_i v_i s^3 + \alpha_i k_{Pi} s^2 + (\alpha_i k_{Ii} + K_i k_{Pi})s + K_i k_{Pi}}{m_i s^3 + k_{Pi} s^2 + (k_{Ii} + K_i k_{Pi})s + K_i k_{Pi}} \tag{15}$$

It can be obviously seen that the orders of the pole and zero are equal. Consequently, the transfer function is allowed to be unity by letting α and v equal to 1 and m, respectively.

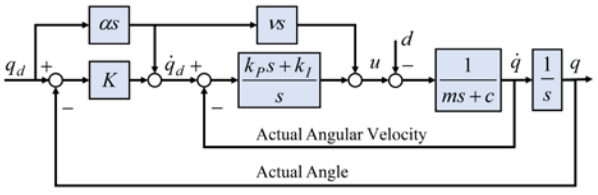


Figure 4. Block diagram of the integrated control structure

Figure 5. A 5-DOF single robot arm

V. Simulation and Experimental Results

In this section, to validate the capability of the control approaches, a 5-DOF robot arm is studied numerically and experimentally. The overall system is shown in Fig. 5 and the system parameters are shown in Table I and II. Two different control approaches are employed to contouring control in order to follow a desired path: the first one is the method of equivalent errors and the other one is the PID controller with the feedforward controller. Then, a tilted circular path is generated to evaluate the contouring performance of the control approaches.

According to the contouring control problem, since the desired path respects the coordinate of the end-effector, the machine coordinate $[q_1 \;\; q_2 \;\; q_3 \;\; q_4 \;\; q_5]^T$ and the workpiece coordinate $[P_x \;\; P_y \;\; P_z \;\; O_x \;\; O_y \;\; O_z]^T$ systems are considered in kinematics. From Fig. 6 and by Denavit-Hartenberg (DH) parameters method [16], the forward kinematics can be obtained as

$$\begin{aligned}
P_x &= 0.3C_1S_{234} + 0.1S_1S_{234} + l_4 + l_5 + l_w \\
&\quad + l_2S_{34} + l_3C_4 - (l_b + l_1)C_{234} \\
P_y &= -0.3\left(S_1S_5 + C_1C_{234}C_5\right) + 0.1\left(-C_1S_5 + S_1C_{234}C_5\right) \\
&\quad + C_5\left(l_2C_{34} - l_3S_4 + (l_b + l_1)S_{234}\right) + (l_{z1} + l_{z3} + l_{z4})S_5 \\
P_z &= -0.3\left(-S_1C_5 + C_1C_{234}S_5\right) + 0.1\left(-C_1C_5 + S_1C_{234}S_5\right) \\
&\quad + S_5\left(l_2C_{34} - l_3S_4 + (l_b + l_1)S_{234}\right) - (l_{z1} + l_{z3} + l_{z4})C_5 \\
O_x &= S_1S_{234} \\
O_y &= -C_1S_5 + S_1C_{234}C_5 \\
O_z &= C_1C_5 + S_1C_{234}S_5
\end{aligned} \tag{16}$$

where $l_1, l_2, ..., l_5$ are the length of each link of the robot arm. l_b and l_w are the length of the base and half the length of workpiece. $[P_x \;\; P_y \;\; P_z]^T$ and $[O_x \;\; O_y \;\; O_z]^T$ represent the displacements and orientations of the workpiece coordinate, respectively. To simplify the cosine and sine functions, let us denote: $C_1 = \cos(q_1)$, $C_2 = \cos(q_2)$, …, $C_5 = \cos(q_5)$, $C_{34} = \cos(q_3 + q_4)$, $C_{234} = \cos(q_2 + q_3 + q_4)$, $S_1 = \sin(q_1)$, $S_2 = \sin(q_2)$, …, $S_5 = \sin(q_5)$, $S_{34} = \sin(q_3 + q_4)$, and

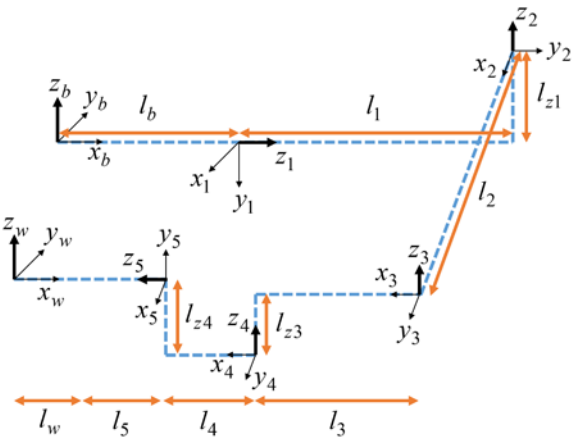


Figure 6. A coordinate system of the 5-DOF robot arm

$S_{234} = \sin(q_2 + q_3 + q_4)$. Then, the inverse kinematics can be obtained from (16) and is omitted here for brevity.

For the following simulation and experimental results, the control parameters are shown in Table III. The desired path is a circular path of radius 0.5 m with a tilted angle of $\pi/4$ rad. In the simulation, the trajectory paths and the actual contour error, which are executed by the controller designed by the method of equivalent errors and the distributed controller, are shown in Figs. 7 and 8. Then, Figs. 9 and 10 present the experimental results.

TABLE I. Parameters for Single Servo Drives

Joint	$m\,(V \cdot s^2/rad)$	$c\,(V \cdot s/rad)$
1	0.1180	1.1184
2	0.1131	0.9165
3	0.0080	0.0965
4	0.0014	0.0731
5	0.0012	0.0131

TABLE II. Parameters of The Length of Links

Length of links (m)									
l_b	l_1	l_2	l_3	l_4	l_5	l_w	l_{z1}	l_{z2}	l_{z3}
0.3	0.1	0.37	0.2	0.1	0.05	0.05	0.13	-0.009	0.05

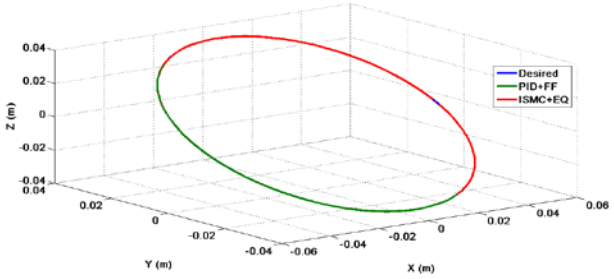


Figure 7. Trajectory paths for the robot arm in simulation

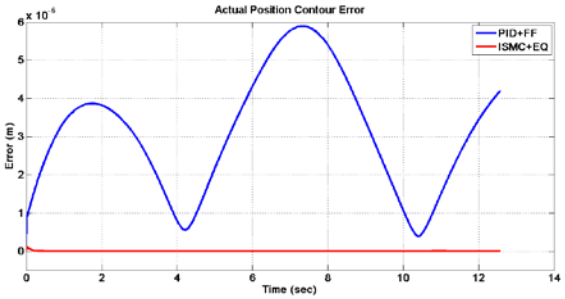


Figure 8. Actual contour errors for the robot arm in simulation

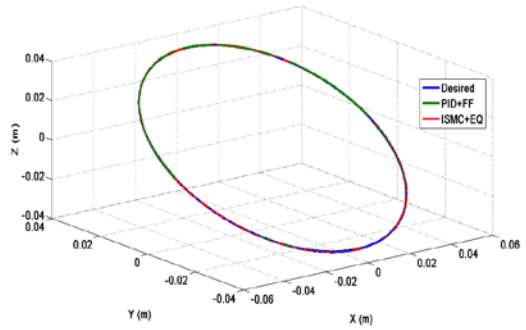


Figure 9. Trajectory path for the robot arm in experiment

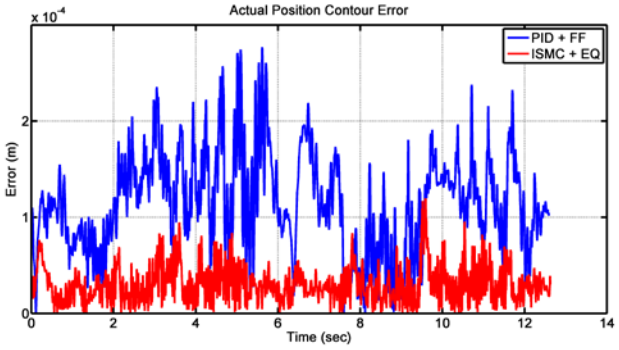


Figure 10. Actual contour error for the robot arm in experiment

TABLE III. NUMBERICAL RESULTS FOR THE 5-DOF ROBOT ARM

Control Approach	Control parameters
FF-enhanced PID	$K_1 = diag(400, 300, 250, 200, 150)$; $K_2 = diag(250, 200, 180, 150, 100)$; $K_P = diag(250, 200, 150, 75, 50)$; $V_{ff} = diag(0.2, 0.5, 0.1, 0.01, 0.001)$; $\alpha = 1$
ISMC	$b_1 = 300$; $b_2 = 30{,}000$; $\rho = diag(10, 30, 250, 800, 100)$; $\delta = diag(1, 2.5, 10, 20, 10)$

As shown in Figs. 7 and 8, it is clear that the method of equivalent errors can provide much better performance than the distributed control. The average contour error is $0.0015\ \mu m$ for the proposed method compared to $3.09\ \mu m$ for the distributed controller. From the experimental results shown in Figs. 9 and 10, the proposed method also outperform the distributed controller, although it is not as significant in the simulations. The average contour error for the proposed controller and distributed controller are $30.25\ \mu m$ and $113.72\ \mu m$, respectively. The contour error by the proposed method is approximately one quarter of the error by the conventional controller. The results clearly verify the effectiveness of the proposed method.

VI. CONCLUSIONS

The contouring control problem of a 5-DOF robot arm has been studied in this work. To study the limitation of the distributed controller, the method of equivalent errors is utilized to reveal the robustness against the strong nonlinearity of the robot arm system. Both approaches employ the kinematic transformation to generate the joint space reference from the desired path. The results compared the outcomes of the controller designed by the method of equivalent errors and the distributed controller. As expected, in both simulation and experimental results, the performance of the distributed controller is limited by the strong nonlinearity in the robotic system, while the controller designed by the method of equivalent errors can more effectively reduce the contour error.

REFERENCES

[1] A. Chatraei and V. Záda, "Global Optimal Feedback-Linearizing Control of Robot Manipulators", *Asian Journal of Control*, vol. 15, no. 4, pp. 1178-1187, 2012.

[2] S. Malagari and B. Driessen, "Globally exponential controller/observer for tracking in robots without velocity measurement", *Asian Journal of Control*, vol. 14, no. 2, pp. 309-319, 2010.

[3] V. Feliu, F. Castillo, V. Jaramillo and G. Partida, "A Robust Controller for A 3-DOF Flexible Robot with a Time Variant Payload", *Asian Journal of Control*, vol. 15, no. 4, pp. 971-987, 2012.

[4] J. Moreno-Valenzuela and V. Santibáñez, "Robust Saturated PI Joint Velocity Control for Robot Manipulators", *Asian Journal of Control*, vol. 15, no. 1, pp. 64-79, 2012.

[5] Y. Koren, "Cross-Coupled Biaxial Computer Control for Manufacturing Systems", *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 4, p. 265, 1980.

[6] G. Chiu and M. Tomizuka, "Contouring control of machine tool feed drive systems: a task coordinate frame approach", *IEEE Transactions on Control Systems Technology*, vol. 9, no. 1, pp. 130-139, 2001.

[7] S. Yeh and P. Hsu, "Perfectly Matched Feedback Control and Its Integrated Design for Multiaxis Motion Systems", *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, no. 3, p. 547, 2004.

[8] J. Yang and Y. Altintas, "A generalized on-line estimation and control of five-axis contouring errors of CNC machine tools", *International Journal of Machine Tools and Manufacture*, vol. 88, pp. 9-23, 2015.

[9] S. Chen and C. Chou, "Contouring Control of Robot Manipulators Based on Equivalent Errors", *Asian Journal of Control*, vol. 16, no. 3, pp. 904-914, 2013.

[10] M. M. Fateh, "Dynamic Modeling of Robot Manipulators in D-H Frames," *World Applied Sciences Journal*, vol. 6, no. 1, pp. 39-44, 2009.

[11] L. L. Ou, W. D. Zhang and L. Yu, "Low-Order Stabilization of LTI Systems With Time Delay," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 774-787, April 2009.

[12] N. S. Nise, "Steady-State Errors", in Control Systems Engineering, 6th ed. New York: John Wiley & Sons, Inc. 2004, ch. 7, pp. 339-386.

[13] A. Krall, "The Root Locus Method: A Survey", *SIAM Review*, vol. 12, no. 1, pp. 64-72, 1970.

[14] N. S. Nise, "Design via Frequency Response", in Control Systems Engineering, 6th ed. New York: John Wiley & Sons, Inc. 2004, ch. 11, pp. 625-662.

[15] G. Casciola and S. Morigi, "REPARAMETRIZATION OF NURBS CURVES", *International Journal of Shape Modeling*, vol. 02, no. 0203, pp. 103-116, 1996.
[16] W. Mohammed Jasim, "Solution of Inverse Kinematics for SCARA Manipulator Using Adaptive Neuro-Fuzzy Network", *International Journal on Soft Computing*, vol. 2, no. 4, pp. 59-66, 2011.
[17] T. Nuchkrua, S. Y. Chang and S. L. Chen, "Contouring control of 5-DOF manipulator robot arm based on equivalent errors," *2015 International Automatic Control Conference (CACS)*, pp. 48-53, 2015.

Woraphrut Kornmaneesang was born on May 31, 1993, in Bangkok, Thailand. He received B.E. degree from King Mongkut's University of Technology Thonburi, Thailand, 2015, in Control System and Instrumentation Engineering. Currently, he is pursuing his M.S. degree in Advanced Institute of Manufacturing with High-tech Innovations, National Chung Cheng University, Taiwan. His research interests include contouring control of multi-axis motion systems and modern control.

Thanana Nuchrua He received M.Eng. degree in mechatronics engineering (with excellent thesis) from Asian Institute of Technology. In year of 2014, he completed Ph.D. degree in Engineering and Technology (mechanical engineering) from Sirindhorn International Institute of Technology (SIIT), Thammasat University, Thailand. He is currently visiting scholar. Prior to this, he was a postdoctoral scholar in National Chung Cheng University, Taiwan, for two years. His research concerns with seeking a control framework with emphasis on the model-based control for the nonlinear control problem subjected to robotics, manufacturing, energy storage devices, rehabilitation engineering inspired by biology. He obtained the recipient of Postgraduate Award, SIIT PhD Scholarship Award for Outstanding Students. He is an active reviewer for many international conferences and journals. 2015-now, some his work published in journal of Bionic Engineering has been a list of the 25 most highly cited papers.

Shyh-Leh Chen was born on October 25, 1964, in Keelung, Taiwan. He received B.S and M.S. degrees from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1987 and 1989, respectively, both in power mechanical engineering. He received a Ph.D. degree in mechanical engineering from Michigan State University, East Lansing, Michigan, USA, in 1996.

Since 1996, he has been with National Chung Cheng University (CCU), Chiayi, Taiwan, where he is currently a Professor in the Department of Mechanical Engineering. He was the Deputy Director of Advanced Institute of Manufacturing with High-tech Innovations (AIM-HI), CCU, from 2011 to 2016. He also served as the Director of Advanced Machine Tools Research Center, CCU, from 2010 to 2013. His research interests include nonlinear dynamics and control, wavelet analysis, with application to motion control of multi-axis systems, active magnetic bearings, and ship stabilization.

Prof. Chen has served as the program chairs/invited session chairs of several international conferences, including 9th World Congress on Intelligent Control and Automation (WCICA 2011), 2013 CACS International Automatic Control Conference (CACS 2013), the 5th International Conference on Advanced Manufacturing (ICAM 2014), and the tenth annual IEEE International Conference on Automation Science and Engineering (IEEE CASE 2014). He received the Automatic Control Award for Young Scholars and Outstanding Automatic Control Award from Chinese Automatic Control Society in 2003 and 2015, respectively, Outstanding Teaching Award from CCU in 2006, Delta Award from IEEE Tainan section in 2017, and Distinguished Research Award from CCU in 2018. He was the recipient of the Best Paper Award of The 2011 IAENG International Conference on Control and Automation, 2013 Conference on Precision Machinery and Manufacturing Technology (PMMT 2013), 2015 CACS International Automatic Control Conference (CACS 2015), and 2017 CACS International Automatic Control Conference (CACS 2017).

Hierarchical Sliding-Mode Formation Control Using ORIT2FNN for Uncertain Networked Multiple Ball-Riding Robots

Feng-Chun Tai, Ching-Chih Tsai *Fellow, IEEE*, Cheng-Kai Chan

***Abstract*—This paper presents a distributed consensus cooperative formation control using hierarchical sliding-mode control and output-feedback recurrent interval type-2 fuzzy neural networks (ORIT2FNN) for a group of uncertain inverse-Atlas-spherical driven ball-riding robots (IAS-BRRs). The dynamic of each IAS-BRR in both sagittal and coronal planes can be modeled using the Euler-Lagrangian equation as two decoupled fourth-order underactuated models, and the whole multi-robot system is described by a directed graph. Based on the hierarchical sliding-mode control and Lyapunov stability theory, an intelligent adaptive, decentralized, cooperative consensus-based formation control method is proposed to accomplish formation control, where the ORIT2FNN is used to approximate the system uncertainties online. Simulations are performed to demonstrate that the proposed control method is useful and beneficial.**

***Index Terms*—Ball-riding robot, consensus control, cooperative formation control, output-feedback recurrent interval type-2 fuzzy neural networks (ORIT2FNN).**

I. Introduction

Cooperative control for homogeneous multi-agents, multi-robots or multi-vehicles has received much attention for past and present decades [1,2]. Homogeneous cooperative control techniques have been successfully applied to several engineering applications in [3-5]. Up to now, researchers have proposed several formation control methods [4,5]; most of them using the consensus cooperative algorithms by assuming the controlled systems being modelled by more than first-order systems models [3-9] with advantage of superior network flexibility. For example, Chen *et al.* [9] presented a fuzzy observer-based consensus tracking approach for nonlinear second-order multi-agent systems.

The ball-riding robots have been investigated by many researchers [12-15]. From the control point of view, motion control of the ball-riding robots can be treated as a fourth-order underactuated control problem, which has been well studied in [16-18]. Tsai *et. al.* [15] presented a dynamic model of ball-riding robot with an LQR control method for station keeping and point stabilization. Lin and Mon [16] proposed a hierarchical decoupling sliding-mode control approach to addressing more general under-actuated systems. Wang *et. al.* [17] presented two sliding-mode design methods for underactuated mechanical systems. Qin *et. al.* [18] established a hierarchical sliding-mode control approach to stabilizing the SIMO underactuated systems with nonlinearity. Inspired by these studies, the aggregated hierarchical sliding-mode control method in [17] would be useful in finding formation control laws of the ball-riding robots.

Backstepping control techniques have been regarded as one of important control methodologies for nonlinear systems, especially for those systems with nonlinear strict feedback forms. Moreover, adaptive backstepping has also been applied to not only achieve station keeping and trajectory tracking of two types of single ball-riding robots [21-22], but also carry out decentralized consensus tracking of a class of second-order multi-agent systems in [9, 23]. That is, the combination of backstepping techniques and sliding-mode control brings outstanding performance, which has been documented in [24] due to the finite time convergence property.

RIT2FNN have been used for identification and approximation of nonlinear system dynamic. It have also been demonstrated powerful and effective in proposing an intelligent backstepping slide-mode control or direct adaptive RIT2FNN-based control method for single ball-riding robot in [21] and [22], respectively. Tsai *et. al.* [23] used an IT2FNN with output feedback, abbreviated as output feedback RIT2FNN (ORIT2FNN), to learn uncertain parts of heterogeneous omnidirectional mobile robots. Inspired by [21-23, 25], ORIT2FNN would be a promising tool in constructing a novel formation controller for a team of heterogeneous ball-riding robots.

The goal of this paper is to develop the distributed cooperative formation control laws of a group of uncertain multiple ball-riding robots using ORIT2FNN, adaptive backstepping approach, and hierarchical sliding-mode control, and to verify the proposed control method is useful and beneficial via simulations. In comparison with the state of the art in cooperative formation control, the presented contents are novel in establishing an intelligent adaptive cooperative formation control law for a team of two types of ball-riding robots modeled by fourth-order underactuated dynamic system models in both sagittal or coronal plane, with the assumption that they moves on a flat terrain.

The rest of this paper is constructed by follows. Section II briefly recalls how to model each ball-riding robot in formation and the overall multi-robot system, and then formulates the control problem. Section III describes how to

Feng-Chun Tai, Ching-Chih Tsai, and Cheng-Kai Chanare with the Department of Electrical Engineering, National Chung Hsing University, Taichung 40227, Taiwan.
(Corresponding author Ching-Chih Tsai, email: cctsai@nchu.edu.tw)
(email: d099064008@mail.nchu.edu.tw, bossmaster_chan@hotmail.com)

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.

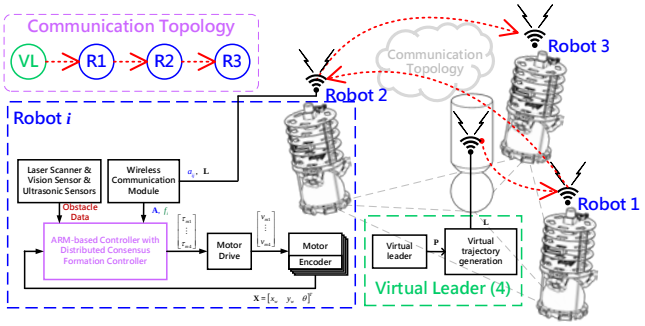


Fig. 1. A homogeneous multi-ball-riding robot system with *n* IAS-BRRs.

design the adaptive cooperative formation control law using ORIT2FNN, adaptive backstepping and hierarchical sliding-mode control. In Section IV, two simulations on formation control of a team of multiple ball-riding robots are performed to show the effectiveness and merit of the proposed control method. Section V concludes the paper.

II. Problem Formulation

This section first introduces the modeling technique of the ball-riding robot in the world frame; a multi-robot system consist of the mentioned ball-riding robot derived by the graph theory; and finally describes the control problem. The mentioned ball-riding robot is considered as a mobile robot equipped with one bawling-like ball and an inverse Atlas spherical motion platform driven by three omnidirectional wheels, as shown in Fig. 1.

2.1 Modeling a Ball-Riding Robot in the World Frame

The dynamic model of the ball-riding robot can be considered as two independent inverted pendulum system in both sagittal and coronal planes since the system state are independent and decoupled, where the system model is first described by [16-17]. Besides, the system models in both sagittal and coronal planes are completely identical except the subscript. Fig. 2 and Table I shows the simple planar model diagrams and the definition of symbol separately. From Fig. 2 and Table I, the system state of each IAS-BRR are defined as:

$$\dot{x} = v_x, \dot{v}_x = a_x = u_x, \dot{y} = v_y, \dot{v}_y = a_y = u_y \tag{1}$$

where the pair (x,y) denotes the position of the contact point between the ball and the terrain; v_x and v_y respectively stand for the velocities of the robot in both sagittal and coronal planes; α_x and α_y are respectively desired acceleration commands in both x and y axes; u_x and u_y are the acceleration control commands in both x and y axes. From [20], once the velocities of the robot have been obtained from the integral of the designed control commands, the angular velocities of three real driving motors are given by

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

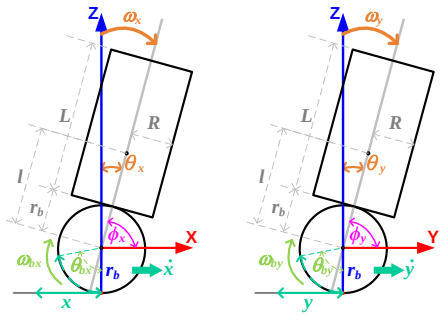


Fig. 2. A simple planar model of a ball-riding robot in the either sagittal or coronal plane.

TABLE I. Definition of Symbol.

Symbol(Unit)	Symbol Description	Typical value
m_B (kg)	Mass of the body	3.2 kg
m_b (kg)	Mass of the ball	0.65 kg
l (m)	Length of the body	0.15 m
r_b (m)	Radius of the ball	0.11 m
I_b (kg-m^2)	Moment of inertia of the ball	$0.005\,[kg\cdot m^2]$
I_B (kg-m^2)	Moment of inertia of the body	$0.032\,[kg\cdot m^2]$
r_w (m)	Radius of each omnidirectional wheel	0.05 m

where r_w is the radius of each omnidirectional wheel, and K_z is a real and positive constant specified by the designer, and the rotational angular rate, ω_z, of the robot with respect to z axis. On the other hand, the dynamic behavior of the whole robot can be simply described by the two-dimensional mobile inverted pendulum. In deriving the dynamic equations for inclination, let θ_x and θ_y (unit: rad) respectively represent the inclination angles of the robot in both saggital and coronal planes, ϕ_x and ϕ_y respectively denotes shaft angles of the ball with respect to the inclination of the robot in both sagittal and coronal planes, as indicated in Fig.1 (c). Using Lagrangian mechanics and the three assumptions:

1) There is no slip between the spherical wheel and the floor;
2) The motion in the median sagittal and coronal plane are decoupled;
3) The equations of motion in these two planes are identical.

one obtains the following equations of motion of θ_x and θ_y.by:

$$\begin{aligned} M_{x11}\ddot{\theta}_x - \beta\dot{\theta}_x^{\,2}\sin(\theta_x) - \frac{\beta g}{r_b}\sin\theta_x &= -M_{x12}\ddot{\phi}_x = -\frac{M_{x12}}{r_b}u_x \\ M_{y11}\ddot{\theta}_y - \beta\dot{\theta}_y^{\,2}\sin(\theta_y) - \frac{\beta g}{r_b}\sin\theta_y &= -M_{y12}\ddot{\phi}_y = -\frac{M_{x12}}{r_b}u_y \end{aligned} \tag{3}$$

where $M_{x11} = \alpha + \gamma + 2\beta\cos(\theta_x)$; $M_{x12} = \alpha + \beta\cos(\theta_x)$; $M_{y11} = \alpha + \gamma + 2\beta\cos(\theta_y)$; $M_{y12} = \alpha + \beta\cos(\theta_y)$; $\alpha = I_b + (m_B + m_b)r_b^{\,2}$; $\beta = m_B r_b l$; $\gamma = m_B l^2 + I_B$.

Consider the uncertainty term, we decompose the three system parameters into m_B=$m_B{}^0$+Δm_B, l=l^0+Δl and I_B=$I_B{}^0$+ΔI_B where Δm_B, Δl and ΔI_B denote the uncertainties of M(q), C(q,$\dot{q}$) , G(q) and D($\dot{q}$) , respectively. By using these three perturbed terms, defining the four state variables, $x_{1i}=\theta_{xi}$, $x_{2i}=x_i$, $v_{1i}=\dot{\theta}_{xi}$, $v_{2i}=\dot{x}_i$ and the control u_i=u_{xi} for the i^{th} ball-riding robot in the sagittal plane, one obtains the

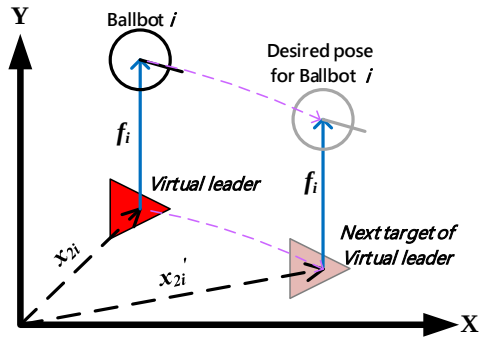


Fig. 2. Illustration of the current and goal pose of ball-riding robot *i*.

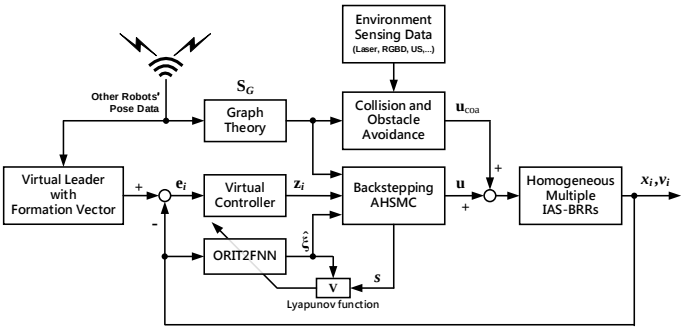


Fig. 3. The structure of the proposed algorithm.

following sagittal-plane dynamic model of ball-riding robot *i*, for $i=1,\ldots,n$,

$$\begin{aligned} \dot{x}_{1i} &= v_{1i},\ \dot{v}_{1i} = b_{1i}(x_{1i},x_{2i},v_{1i},v_{2i})u_i + g_{1i}(x_{1i},x_{2i},v_{1i},v_{2i}) + \xi_{1i}, \\ \dot{x}_{2i} &= v_{2i},\ \dot{v}_{2i} = b_{2i}(x_{1i},x_{2i},v_{1i},v_{2i})u_i + g_{2i}(x_{1i},x_{2i},v_{1i},v_{2i}) + \xi_{2i} \end{aligned} \quad (4)$$

where $b_{1i}=-M_{x12}/(r_b M_{x11})$, $g_{1i}=(\beta(v_{1i})^2\sin(x_{1i})+\beta g\sin(x_{1i})/r_b)/M_{x11}$, $b_{2i}=1$, $g_{2i}=0$.

In (4), $x_{1i}, v_{1i}, x_{2i}, v_{2i} \in R^1$ are represented as the inclination angle, and tilt rate, translational position, translational position velocity, of the i^{th} ball-riding robot in the sagittal plane, respectively, and they are all directly measured and bounded; ξ_{1i} and ξ_{2i} caused by the three system parameters are two continuous and bounded disturbances satisfying $|\xi_{1i}| \le h_{1\xi\max} < \infty$ and $|\xi_{2i}| \le h_{2\xi\max} < \infty$. Note that, by using the similar procedure, we have the completely identical coronal-plane dynamic model of ball-riding robot *i*, for $i=1,\ldots,n$. Moreover, the system model (4) represent the fourth-order underactuated state equations of the i^{th} ball-riding robot in in the either sagittal or coronal plane.

2.2 Modeling a Multi-Ball-riding robot System

Generally speaking, a multi-ball-riding robot system has been considered as a group of ball-riding robots, where a leader exchanges information with other IAS-BRRs via communication architecture. Suppose that interconnection topology of *n* ball-riding robots is a directed graph *G*, and *n* ball-riding robots can be regarded as *n* nodes. The relevant weighted adjacency matrix is denoted as $\boldsymbol{A} = [a_{ij}]$ and $a_{ij} \ge 0$, $\forall\ i, j \in \{1, 2, \ldots, n\}$. Moreover we assume $a_{ii} = 0$. The Laplacian matrix *L* of the directed graph *G* is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D}=diag(d_1,d_2,\ldots,d_n)$, and $d_i = \sum_{j=1}^{n} a_{ij}$. The interconnection topology of the overall multi-ball-riding robot system is also a directed graph $\bar{G}$, and its Laplacian matrix $\bar{\boldsymbol{L}}$ is given by $\bar{\boldsymbol{L}} = \bar{\boldsymbol{D}} - \bar{\boldsymbol{A}}$. In order to achieve this formation control objective, three assumptions about the communication topology are made as follows:

(i) The graph $\bar{G}$ with its relevant Laplacian matrix $\bar{\boldsymbol{L}}$ is directed and has a spanning tree with root being the virtual leader, the $(n+1)^{th}$ ball-riding robot;
(ii) At least one follower robot accesses the information from the leader
(iii) The dynamics of the leader must be independent from any ball-riding robot.

According to Assumption (i), the Laplacian matrix $\bar{\boldsymbol{L}}$ is symmetric and has only one zero eigenvalue and all other nonzero eigenvalues with positive real part. On basis of Assumptions (i)-(iii), there exists a diagonal matrix $\boldsymbol{B} = diag\{a_{1(n+1)}, \ldots, a_{n(n+1)}\}$ where all diagonal entries are nonnegative and it has at least one positive diagonal entry such that the summation of both matrices, i.e., $\boldsymbol{S}_G = \boldsymbol{L} + \boldsymbol{B}$, is invertible and has all the nonzero eigenvalues with positive real parts; therefore, there exist a symmetric and positive -definite matrix $\boldsymbol{Q}$, and a symmetric and diagonal matrix $\boldsymbol{P}$, such that $\boldsymbol{P}\boldsymbol{S}_G + \boldsymbol{S}_G{}^T\boldsymbol{P} = \boldsymbol{Q}$. Moreover, the $\boldsymbol{P}$ can be found by $\boldsymbol{P} = diag\{p_1, \ldots, p_n\}$, where $[1/p_1, \ldots, 1/p_n]^T = \boldsymbol{S}^{-1}[1, \ldots, 1]^T$.

2.3 Problem Statement

The control objective of the formation control is briefly described as follows: each of the ball-riding robots follows virtual leader, and the pose asymptotically converges to the desired relative pose between the i^{th} ball-riding robot and their leader. This is formulated as $\lim_{t\to\infty} x_{1i}(t) = \lim_{t\to\infty} v_{1i}(t) = 0$,

$$\lim_{t\to\infty}\left(x_{2i}(t) - x_{2i}^*(t)\right) = \lim_{t\to\infty}\left(v_{2i}(t) - v_{2i}^*(t)\right) = 0, \qquad i=1,2,\ldots,n,$$

where $x_{2i}^*(t) = x_L(t) + f_i(t)$ and $v_{2i}^*(t) = \dot{x}_L(t) + \dot{f}_i(t)$ are respectively denoted by the desired pose and velocity of the i^{th} robot, x_{1i} is the position of the i^{th} robot, $x_L(t)$ is the pose of the leader, and $f_i(t)$ is the desired relative pose vector between the i^{th} ball-riding robot and leader. Note that the desired relative pose for the i^{th} ball-riding robot should be different from it for any other ball-riding robot, and $f_{n+1}(t) = 0$ and $\dot{f}_{n+1}(t) = 0$. Moreover, we further assume that $x_L(t)$ and $f_i(t)$, $i=1,..,n$, are twice differentiable; $f_i(t)$, $\dot{f}_i(t)$ and $\ddot{f}_i(t)$, $i=1,..,n$, are known for the i^{th} robot, and all the state variables of the i^{th} ball-riding robot are directly measured.

III. Proposed Method

This section will derive a consensus-based cooperative formation control law using ORIT2FNN, as shown in Fig. 3. In doing so, the ORIT2FNN will be briefly described in order to on-line learn the nonlinear and state-dependent functions, ξ_{1i} and ξ_{2i}, in (11), and then the intelligent adaptive consensus-based cooperative formation controller is synthesized to not only let a group of ball-riding robot s cooperatively move in formation, but also control each ball-riding robot to generate the geometric configuration of the formation.

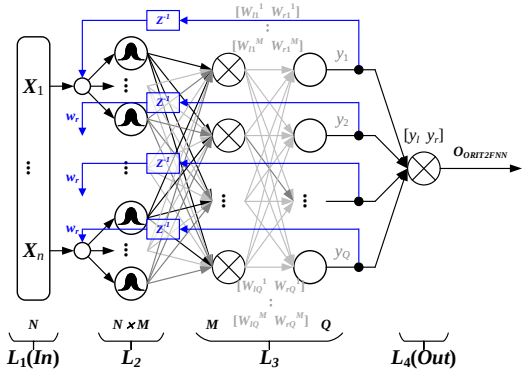


Fig. 4. The structure of the proposed ORIT2FNN.

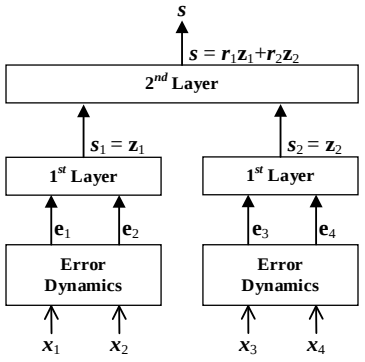


Fig. 5. Schematic of aggregated hierarchical sliding-mode control (AHSMC) with adaptive backstepping control.

3.1 Online ORIT2FNN Function Learning

The structure of the used ORIT2FNN as shown in Fig. 4. Including four layers: the input, the membership function, the rule and the output layers. By adding feedback connection through delay from the output layer, the recurrent feedback is embedded in the network [18-20]. Basically, the output of the proposed ORIT2FNN can be expressed by the following matrix-vector form.

$$\boldsymbol{y}(x) = \boldsymbol{W}^T \boldsymbol{Y}(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{m}, \boldsymbol{\omega}) \in \mathbb{R}^1 \tag{5}$$

where $\boldsymbol{W}^T = [W_L\ W_R] = [W_{r1} \cdots W_{rn},\ W_{l1} \cdots W_{ln}]$, $\boldsymbol{Y} = [\boldsymbol{Y}_L\ \boldsymbol{Y}_R]^T$, and $\boldsymbol{y}(x) = [y_1 \quad \cdots \quad y_Q]^T$. For page limitation, the above notations are defined in [18-20].

3.2 Intelligent Distributed Formation Control

This section is devoted to designing each distributed consensus cooperative control law u_i such that $\lim_{t\to\infty}(x_{1i} - x_{1i}^*) = 0$ and $\lim_{t\to\infty}(v_{1i} - v_{1i}^*) = 0$, i=1,…,n, In doing so, the overall formation system based on the decoupled system model (4) can be rewritten in a vector-matrix form

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{v}_1, \quad \dot{\mathbf{v}}_1 = \mathbf{b}_1(\mathbf{x}_1, \mathbf{v}_1)\mathbf{u} + \mathbf{g}_1(\mathbf{x}_1, \mathbf{v}_1) + \boldsymbol{\xi}_1 \\ \dot{\mathbf{x}}_2 &= \mathbf{v}_2, \quad \dot{\mathbf{v}}_2 = \mathbf{b}_2(\mathbf{x}_2, \mathbf{v}_2)\mathbf{u} + \mathbf{g}_2(\mathbf{x}_2, \mathbf{v}_2) + \boldsymbol{\xi}_2 \end{aligned} \tag{6}$$

where

$\mathbf{x}_1 \triangleq (x_{11}, x_{12}, ..., x_{1n})^T$, $\mathbf{v}_1 \triangleq (v_{11}, v_{12}, ..., v_{1n})^T$,
$\mathbf{x}_2 \triangleq (x_{21}, x_{22}, ..., x_{2n})^T$, $\mathbf{v}_2 \triangleq (v_{21}, v_{22}, ..., v_{2n})^T$, $\mathbf{u} \triangleq (u_1, u_2, ..., u_n)^T$,
$\mathbf{g}_1 \triangleq (g_{11}, ..., g_{1n})^T$, $\mathbf{g}_2 \triangleq (g_{21}, ..., g_{2n})^T$, $\mathbf{b}_1 \triangleq diag(b_{11}, ..., b_{1n})^T$,
$\mathbf{b}_2 \triangleq diag(b_{21}, ..., b_{2n})^T$, $\boldsymbol{\xi}_1 \triangleq (\xi_{11}, \xi_{12}, ..., \xi_{1n})^T$, $\boldsymbol{\xi}_2 \triangleq (\xi_{21}, \xi_{22}, ..., \xi_{2n})^T$.

Note that both matrices $\mathbf{b}_1$ and $\mathbf{b}_2$ are diagonal and all their diagonal entries are nonzero. The proposed cooperative formation control using ORIT2FNN is synthesized as shown in the subsequent three steps.

Step1: Let us first define the tracking error vector of the first order subsystems as $\mathbf{e}_1 = \mathbf{x}_1 - \mathbf{x}_1^*$ where $\mathbf{x}_1^* \triangleq (0, 0, ..., 0)^T$. Then it follows that

$$\dot{\mathbf{e}}_1 = \dot{\mathbf{x}}_1 - \dot{\mathbf{x}}_1^* = \mathbf{v}_1 - \mathbf{v}_1^* = \mathbf{v}_1 \tag{7}$$

where $\boldsymbol{v}_1^* \triangleq (0, 0, ..., 0)^T \in R^n$. To stabilize the $\boldsymbol{e}_1$ dynamics, let us consider v_2 as another virtual control vector let

$$\mathbf{v}_1 = \boldsymbol{\Phi}_1(\mathbf{e}_1) = -\mathbf{K}\mathbf{e}_1 \tag{8}$$

in which the matrix $\boldsymbol{K}$ is the diagonal and positive-definite matrix. With the virtual controls, the dynamics of tracking errors becomes $\dot{\boldsymbol{e}}_1 = -\boldsymbol{K}\boldsymbol{e}_1$. It is easy to show the tracking error vector $\boldsymbol{e}_1$ approach zero as time goes to infinity by choosing the following positive-definite Lyapunov function V_{11} as

$$V_1 = \mathbf{e}_1^{\mathrm{T}}\mathbf{e}_1 / 2 \tag{9}$$

where leads to $\dot{V}_1 = -\mathbf{e}_1^{\mathrm{T}}\mathbf{K}\mathbf{e}_1 < 0$.

Next, let us define the formation state error of the i^{th} system in formation be given by, for i = 1, .., n,

$$e_{2i} = \sum_{j=1}^{n+1} a_{ij}(x_{2i} - x_{2j}) \tag{10}$$

which can be rewritten in a vector-matrix form,

$$\mathbf{e}_2 = \boldsymbol{S}_G(\mathbf{x}_2 - \mathbf{x}_2^*) \tag{11}$$

where $\boldsymbol{x}_2^* \triangleq (x_{21}^*, x_{22}^*, ..., x_{2n}^*)^T$, and the matrix $\boldsymbol{S}_G$ is reduced from the graph Laplacian $\bar{\boldsymbol{L}}$ and given by

$$\boldsymbol{S}_G = \begin{bmatrix} \sum_{j=1}^{n+1} a_{1j} & -a_{12} & \cdots & -a_{1n} \\ -a_{2n} & \sum_{j=1}^{n+1} a_{2j} & \cdots & -a_{2n} \\ \cdots & \cdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & \sum_{j=1}^{n+1} a_{nj} \end{bmatrix} \tag{12}$$

Note that the matrix $\mathbf{S}_G$ is symmetric, and has all the nonzero eigenvalues with positive real parts, moreover, it is invertible. Furthermore, there exist a symmetric and *P.D.* matrix $\boldsymbol{Q}$, and a symmetric and diagonal matrix $\boldsymbol{P}$ such that $\boldsymbol{P}\boldsymbol{S}_G + \boldsymbol{S}_G{}^T\boldsymbol{P} = \boldsymbol{Q}$

Hence, the dynamics of the formation state error vector is

$$\dot{\mathbf{e}}_2 = \boldsymbol{S}_G(\dot{\mathbf{x}}_2 - \dot{\mathbf{x}}_2^*) = \boldsymbol{S}_G(\mathbf{v}_2 - \dot{\mathbf{x}}_2^*) \tag{13}$$

where $\mathbf{v}_2^* \triangleq \dot{\mathbf{x}}_2^* \in R^n$. To stabilize the $\boldsymbol{e}_2$ dynamics, we consider the virtual control vector $\boldsymbol{v}_2$ as

$$\mathbf{v}_2 = \boldsymbol{\Phi}_2(\mathbf{e}_2) = \dot{\mathbf{f}} + \boldsymbol{S}_G^{-1} \otimes \mathbf{I}_1(\mathbf{b} \otimes \dot{\mathbf{x}}_L) - c_0\mathbf{e}_2 \tag{14}$$

where the vector $\boldsymbol{b}$ is given by $\boldsymbol{b}$ = $[a_{1(n+1)}, \ldots, a_{n(n+1)}]^T$ and

c_0>0. With the virtual control vector in (14), the dynamics of formation state error vector becomes

$$\begin{aligned}\dot{\mathbf{e}}_2 &= \boldsymbol{S}_G(\mathbf{v}_2 - \dot{\mathbf{x}}_2^*) \\ &= \boldsymbol{S}_G\left(\dot{\mathbf{f}} + \boldsymbol{S}_G^{-1}\otimes\mathbf{I}_1(\mathbf{b}\otimes\dot{\mathbf{x}}_L) - c_0\mathbf{e}_2 - \dot{\mathbf{x}}_2^*\right) \\ &= -c_0\boldsymbol{S}_G\otimes\mathbf{I}_1\mathbf{e}_2\end{aligned} \tag{15}$$

Furthermore, we choose the following positive-definite Lyapunov function V_2 as

$$V_2 = \mathbf{e}_2^T\mathbf{P}\mathbf{e}_2/2 \tag{16}$$

Taking the time derivative of (16) along the trajectories of (15), one obtains

$$\dot{V}_2 = -c_0\mathbf{e}_2{}^T(\mathbf{S}^T\mathbf{P}+\mathbf{P}\mathbf{S})\mathbf{e}_2 = -c_0\mathbf{e}_2{}^T\mathbf{Q}\mathbf{e}_2 < 0 \tag{17}$$

which shows that $\boldsymbol{e}_2 \to 0$ as $t_2 \to \infty$ exponentially.

Step 2: Define the backstepping errors as

$$\begin{aligned}\mathbf{z}_1 &= \mathbf{v}_1 - \boldsymbol{\Phi}_1(\mathbf{e}_1) = \mathbf{v}_1 - (-\mathbf{K}\mathbf{e}_1) = \mathbf{v}_1 + \mathbf{K}\mathbf{e}_1 \\ \mathbf{z}_2 &= \mathbf{v}_2 - \boldsymbol{\Phi}_2(\mathbf{e}_2) = \mathbf{v}_2 - \left(\dot{\boldsymbol{f}} + \boldsymbol{S}_G^{-1}\otimes\mathbf{I}_1(\mathbf{b}\otimes\dot{\mathbf{x}}_L) - c_0\mathbf{e}_2\right) \\ &= \mathbf{v}_2 - \dot{\boldsymbol{f}} - \boldsymbol{S}_G^{-1}\otimes\mathbf{I}_1(\mathbf{b}\otimes\dot{\mathbf{x}}_L) + c_0\mathbf{e}_2\end{aligned} \tag{18}$$

With the defined backstepping errors , the system model (6) can be rewritten by

$$\begin{aligned}\dot{\mathbf{e}}_1 &= \mathbf{z}_1 - \mathbf{K}\mathbf{e}_1 \\ \dot{\mathbf{z}}_1 &= \mathbf{b}_1\mathbf{u} + \mathbf{g}_1 + \mathbf{K}\dot{\mathbf{e}}_1 + \boldsymbol{\xi}_1 \\ \dot{\mathbf{e}}_2 &= \boldsymbol{S}_G\mathbf{z}_2 - c_0\boldsymbol{S}_G\mathbf{e}_2 \\ \dot{\mathbf{z}}_2 &= \mathbf{b}_2\mathbf{u} + \mathbf{g}_2 - \ddot{\boldsymbol{f}} - \boldsymbol{S}_G^{-1}\otimes\mathbf{I}_1(\mathbf{b}\otimes\ddot{\mathbf{x}}_L) + c_0\dot{\mathbf{e}}_2 + \boldsymbol{\xi}_2\end{aligned} \tag{19}$$

To stabilize all the error vectors $\boldsymbol{e}_1$, $\boldsymbol{z}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{z}_2$, we use the hierarchical sliding-mode control method in [15] to find the control vector $\boldsymbol{u}$, as shown in Fig. 5, and define the two sliding surfaces without $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ as

$$\boldsymbol{s} = \boldsymbol{s}_1 + \boldsymbol{\Lambda}\boldsymbol{s}_2 = \mathbf{z}_1 + \boldsymbol{\Lambda}\mathbf{z}_2 \tag{20}$$

where both matrices $\boldsymbol{\Lambda}$=*diag*{λ_1, λ_2,…, λ_n} is diagonal, positive-definite, and real-valued. Taking the time derivative of the second-layer sliding surface, one obtains

$$\begin{aligned}\dot{\boldsymbol{s}} &= \dot{\mathbf{z}}_1 + \boldsymbol{\Lambda}\dot{\mathbf{z}}_2 \\ &= (\mathbf{b}_1 + \boldsymbol{\Lambda}\mathbf{b}_2)\mathbf{u} + (\mathbf{g}_1 + \boldsymbol{\Lambda}\mathbf{g}_2) + \mathbf{K}\dot{\mathbf{e}}_1 \\ &\quad + \boldsymbol{\Lambda}\left(-\left(\ddot{\boldsymbol{f}} + \mathbf{S}_G^{-1}(\mathbf{b}\otimes\ddot{\mathbf{x}}_L)\right) + c_0\dot{\mathbf{e}}_2\right) + \boldsymbol{\xi}\end{aligned} \tag{21}$$

where $\boldsymbol{\xi} = \boldsymbol{\xi}_1 + \Lambda\boldsymbol{\xi}_2 \triangleq (\xi_1, \xi_2, ..., \xi_n)^T$.

In the sequel, the mentioned ORIT2FNN approximator is used to approximate the nonlinear term ξ_i, so one can let the ideal approximation result be

$$\xi_i = W_i^{*T}\varphi_i^*(x_i,\ c_i^*,\ \omega_i^*) + \varepsilon_{if}^* \tag{22}$$

where W_i^* is the optimal weight vector, φ_i^* is the optimal ORIT2FNN vector, ε_{if}^* is a small and bounded error vector. It is assumed that the norms of the optimal weights $\|W_i\|$, $\|\omega_i\|$ and $\|c_i\|$ are bounded. Then the actual ORIT2FNN estimate, $\hat{\xi}_i$, can be represented by

$$\hat{\xi}_i = \hat{W}_i^T\hat{\varphi}_i(x_i,\ \hat{c}_i,\ \hat{\omega}_i) \tag{23}$$

where $\hat{W}_i$, $\hat{\varphi}_i$, $\hat{c}_i$, $\hat{\omega}_i$ are estimates of W_i^*, φ_i^*, c_i^*, ω_i^* in the network. By defining $\tilde{W}_i = W_i^* - \hat{W}_i$, $\varphi_i = \varphi_i^* - \hat{\varphi}_i$, $\tilde{\omega}_i = \omega_i^* - \hat{\omega}_i$, $\tilde{c}_i = c_i^* - \hat{c}_i$, and expanding $\tilde{\varphi}_i = \varphi_i^* - \hat{\varphi}_i$ using Taylor series, i.e.,

$$\tilde{\varphi}_i = \frac{\partial\tilde{\varphi}_i}{\partial\hat{\omega}_i}\tilde{\omega}_i + \frac{\partial\tilde{\varphi}_i}{\partial\hat{c}_i}\tilde{c}_i + H_i = A_i\tilde{\omega}_i + B_i\tilde{c}_i + H_i \tag{24}$$

which can be rewritten by

$$\begin{aligned}\xi_i &= \left(\tilde{W}_i + \hat{W}_i\right)^T\left(\tilde{\varphi}_i + \hat{\varphi}_i\right) + \varepsilon_{if}^* \\ &= \hat{\xi}_i + \hat{W}_i^T A_i\tilde{\omega}_i + \hat{W}_i^T B_i\tilde{c}_i + \tilde{W}_i^T\hat{\varphi}_i + h_i\end{aligned} \tag{25}$$

where $h_i = \hat{W}_i^T H_i + \tilde{W}_i^T\tilde{\varphi}_i + \varepsilon_{if}^*$ and $\|h_i\|_\infty < h_{i\max} < \infty$. Thus, the estimate error of $\tilde{\xi}_i$ is obtained from

$$\begin{aligned}\tilde{\xi}_i &\triangleq \xi_i - \hat{\xi}_i \\ &= \tilde{\omega}_i^T A_i^T\hat{W}_i + \tilde{c}_i^T B_i^T\hat{W}_i + \tilde{W}_i^T\hat{\varphi}_i + h_i\end{aligned} \tag{26}$$

Step 3: Next, in order to achieve the formation control and self-balancing, we choose $\hat{\boldsymbol{\xi}} \triangleq \left(\hat{\xi}_1,\ \hat{\xi}_2,\ ...,\ \hat{\xi}_n\right)^T$, which satisfy $\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi} - \hat{\boldsymbol{\xi}}$, from (21), the following decentralized sliding-mode consensus control law by following:

$$\begin{aligned}\mathbf{u} = &\frac{-\mathbf{g}_1 - \boldsymbol{\Lambda}\mathbf{g}_2 - \mathbf{K}\dot{\mathbf{e}}_1 + \boldsymbol{\Lambda}\ddot{\boldsymbol{f}} + \boldsymbol{\Lambda}\boldsymbol{S}_G^{-1}(\mathbf{b}\otimes\ddot{\mathbf{x}}_L)}{\mathbf{b}_1 + \boldsymbol{\Lambda}\mathbf{b}_2} \\ &+ \frac{-\boldsymbol{\Lambda}c_0\dot{\mathbf{e}}_2 - h_1\boldsymbol{s} - h_2\,\mathrm{sgn}(\boldsymbol{s}) - \hat{\boldsymbol{\xi}}}{\mathbf{b}_1 + \boldsymbol{\Lambda}\mathbf{b}_2}\end{aligned} \tag{27}$$

Such that

$$\dot{\boldsymbol{s}} = -h_1\boldsymbol{s} - h_2\,\mathrm{sgn}(\boldsymbol{s}) + \tilde{\boldsymbol{\xi}} \tag{28}$$

where $h_1 > 0 \in R^1$ and $h_2 \in R^1$ are both positive design parameters such that $h_2 > h_{\max} + h_{\xi\max}$, where $h_{\max} = \max\{h_{i\max},\ \ldots,\ h_{n\max}\}$. One can find the parameter updating rules for the ORIT2FNN function approximator and make sure that system is stable via the Lyapunov theorem by selecting the Lyapunov function V_3 in the following.

$$\begin{aligned}V_3 = &\frac{1}{2}\boldsymbol{s}^T\boldsymbol{s} \\ &+ \sum_{i=1}^n\frac{1}{2}\frac{1}{\eta_{W_i}}\tilde{W}_i^T\tilde{W}_i + \sum_{i=1}^n\frac{1}{2}\frac{1}{\eta_{\omega_i}}\tilde{\omega}_i^T\tilde{\omega}_i + \sum_{i=1}^n\frac{1}{2}\frac{1}{\eta_{c_i}}\tilde{c}_i^T\tilde{c}_i\end{aligned} \tag{29}$$

where $\eta_{W_{ji}} > 0$, $\eta_{\omega_{ji}} > 0$, $\eta_{c_{ji}} > 0$, $i = 1, 2, \ldots, n$, are the parameter updating gains. Taking the time derivative of V_3 along the trajectories of (28), one obtains

$$\begin{aligned}\dot{V}_3 \le &-h_1\boldsymbol{s}^T\boldsymbol{s} - \left(h_2 - h_{\max} - h_{\xi\max}\right)\|\boldsymbol{s}\|_1 + \sum_{i=1}^n\tilde{W}_i^T\left(\frac{1}{\eta_{W_i}}\dot{\tilde{W}}_i + \hat{\varphi}_i s_i\right) \\ &+ \sum_{i=1}^n\tilde{\omega}_i^T\left(\frac{1}{\eta_{\omega_{1i}}}\dot{\tilde{\omega}} + A_i^T\hat{W}_i s_i\right) + \sum_{i=1}^n\tilde{c}_i^T\left(\frac{1}{\eta_{c_i}}\dot{\tilde{c}}_i + B_i^T\hat{W}_i s_i\right)\end{aligned} \tag{30}$$

where $\|\boldsymbol{s}\|_1 = \sum_{i=1}^n|\boldsymbol{s}_i|$. Let the parameter updating rules are chosen by the following, for i=1, …, n,

$$\dot{\hat{\mathbf{W}}}_i = \eta_{W_i}\hat{\boldsymbol{\varphi}}_i \boldsymbol{s}_i \ , \ \dot{\hat{\boldsymbol{\omega}}}_i = \eta_{\omega_i}\mathbf{A}_i^T\hat{\mathbf{W}}_i\boldsymbol{s}_i \ , \ \dot{\hat{\mathbf{c}}}_i = \eta_{c_i}\mathbf{B}_i^T\hat{\mathbf{W}}_i\boldsymbol{s}_i \quad (31)$$

Then it turns out

$$\dot{V}_3 \le -h_1\boldsymbol{s}^T\boldsymbol{s} - (h_2 - h_{\max} - h_{\xi\max})|\boldsymbol{s}| \le 0 \quad (32)$$

which is negative semi-definite. From the sliding-mode control theory and the Barbalat's lemma, it implies that $\boldsymbol{s} \to 0$ asymptotically. On the other hand, we can show both first-layer sliding surfaces converge to zero in finite time, namely that $\boldsymbol{s}_1 = \mathbf{z}_1 \to 0$ and $\boldsymbol{s}_2 = \mathbf{z}_2 \to 0$ in finite time. In doing so, we first prove that $\|\boldsymbol{s}_1\|_2 \in L_2$, $\|\boldsymbol{s}_2\|_2 \in L_2$, i.e.

$$\int_0^\infty \|\boldsymbol{s}_1\|_2^2 \, d\tau < \infty \ , \ \int_0^\infty \|\boldsymbol{s}_2\|_2^2 \, d\tau < \infty \quad (33)$$

From (32), we have

$$\begin{aligned} V(t) &= V(0) - \int_0^\infty \left(h_1\boldsymbol{s}^T\boldsymbol{s} + (h_2 - h_{\max} - h_{\xi\max})|\boldsymbol{s}|\right)d\tau \\ &\le V(0) < \infty \end{aligned} \quad (34)$$

since $h_2 > h_{\max} + h_{\xi\max}$, we can find that $\int_0^\infty \boldsymbol{s}^T\boldsymbol{s}d\tau < \infty$, which leads to

$$\begin{aligned} \int_0^\infty \boldsymbol{s}^T\boldsymbol{s}\, d\tau &= \int_0^\infty (\boldsymbol{s}_1 + \boldsymbol{\Lambda}\boldsymbol{s}_2)^T(\boldsymbol{s}_1 + \boldsymbol{\Lambda}\boldsymbol{s}_2)\, d\tau \\ &= \int_0^\infty (\boldsymbol{s}_1^T + \boldsymbol{s}_2^T\boldsymbol{\Lambda}^T)(\boldsymbol{s}_1 + \boldsymbol{\Lambda}\boldsymbol{s}_2)\, d\tau \\ &= \int_0^\infty (\boldsymbol{s}_1^T\boldsymbol{s}_1 + \boldsymbol{s}_1^T\boldsymbol{\Lambda}\boldsymbol{s}_2 + \boldsymbol{s}_2^T\boldsymbol{\Lambda}^T\boldsymbol{s}_1 + \boldsymbol{s}_2^T\boldsymbol{\Lambda}^T\boldsymbol{\Lambda}\boldsymbol{s}_2)\, d\tau \\ &= \int_0^\infty (\boldsymbol{s}_1^T\boldsymbol{s}_1 + 2\boldsymbol{s}_1^T\boldsymbol{\Lambda}\boldsymbol{s}_2 + \boldsymbol{s}_2^T\boldsymbol{\Lambda}^T\boldsymbol{\Lambda}\boldsymbol{s}_2)\, d\tau < \infty \end{aligned} \quad (35)$$

From the inequality of quadratic equation $2a^Tb \le a^Ta + b^Tb$ that holds for any vectors, it is obvious that

$$\int_0^\infty 2\boldsymbol{s}_1^T\boldsymbol{\Lambda}\boldsymbol{s}_2 \, d\tau \le \int_0^\infty (\boldsymbol{s}_1^T\boldsymbol{s}_1 + \boldsymbol{s}_2^T\boldsymbol{\Lambda}^T\boldsymbol{\Lambda}\boldsymbol{s}_2)\, d\tau \quad (36)$$

which leads to

$$\int_0^\infty 4\boldsymbol{s}_1^T\boldsymbol{\Lambda}\boldsymbol{s}_2 \, d\tau \le \int_0^\infty \boldsymbol{s}^T\boldsymbol{s}\, d\tau < \infty \Rightarrow \int_0^\infty \boldsymbol{s}_1^T\boldsymbol{\Lambda}\boldsymbol{s}_2 \, d\tau < \infty \quad (37)$$

We take (35) and (37) with the inequality $\boldsymbol{s}_1^T\boldsymbol{s}_1 \ge 0$ and $\boldsymbol{s}_2^T\boldsymbol{\Lambda}^T\boldsymbol{\Lambda}\boldsymbol{s}_2 \ge 0$, which imply that

$$\begin{aligned} &\int_0^\infty (\boldsymbol{s}_1^T\boldsymbol{s}_1 + \boldsymbol{s}_2^T\boldsymbol{\Lambda}^T\boldsymbol{\Lambda}\boldsymbol{s}_2)d\tau < \infty \\ &\Rightarrow \int_0^\infty \boldsymbol{s}_1^T\boldsymbol{s}_1 d\tau < \infty \ , \ \int_0^\infty \boldsymbol{s}_2^T\boldsymbol{\Lambda}^T\boldsymbol{\Lambda}\boldsymbol{s}_2 d\tau < \infty \end{aligned}$$

Now we discuss the condition of $\boldsymbol{s}_1\cdot\boldsymbol{s}_2 < 0$. From the derivation of the hierarchical sliding-mode control, we can find that λ_i doesn't influence the stability of the second-layer sliding surface. Hence, we can choose $\lambda_i = -\lambda_{0i}$ when $\boldsymbol{s}_1\cdot\boldsymbol{s}_2 < 0$. Therefore $\lambda_i\, \boldsymbol{s}_1\cdot\boldsymbol{s}_2 = -\lambda_{0i}\, \boldsymbol{s}_1\cdot\boldsymbol{s}_2 > 0$. We can also obtain the conclusion of (33). In summary, so long as we choose

$$\lambda_i = \begin{cases} \lambda_{0i} \ , & \boldsymbol{s}_1\cdot\boldsymbol{s}_2 \ge 0 \\ -\lambda_{0i} \ , & \boldsymbol{s}_1\cdot\boldsymbol{s}_2 < 0 \end{cases} \quad (\lambda_{0i} > 0) \quad \text{for } i = 1, 2.$$

From (34) we can obtain $\|\boldsymbol{s}\|_2 \in L_\infty$. At the same time, from (32) we can also obtain $\|\dot{\boldsymbol{s}}\|_2 \in L_\infty$. Because the choice of λ_i guarantees $\lambda_i\boldsymbol{s}_1\cdot\boldsymbol{s}_2 \ge 0$, we obtain $\|\boldsymbol{s}_1\|_2 \in L_\infty$, $\|\boldsymbol{s}_2\|_2 \in L_\infty$, i.e.

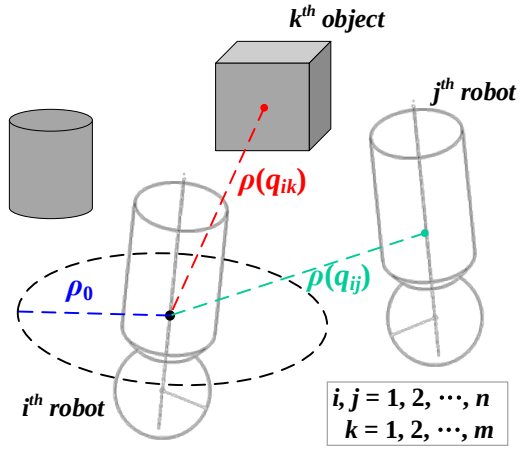


Fig. 6. Illustration of the collision and obstacle avoidance strategies for the multi-ball-riding robot system.

$$\sup_{t\ge 0}\|\boldsymbol{s}_1\| = \|\boldsymbol{s}_1\|_\infty < \infty, \ \sup_{t\ge 0}\|\boldsymbol{s}_2\| = \|\boldsymbol{s}_2\|_\infty < \infty \quad (38)$$

From the equation $\boldsymbol{s}_1$=$\boldsymbol{z}_1$, it follows that

$$\dot{\boldsymbol{s}}_1 = \mathbf{b}_1\mathbf{u} + \mathbf{g}_1 + \mathbf{K}\dot{\mathbf{e}}_1 + \boldsymbol{\xi}_1 \quad (39)$$

Since all variables in the right-hand side of (39) are bounded. Hence, $\|\dot{\boldsymbol{s}}_1\|$ is bounded, i.e., $\|\dot{\boldsymbol{s}}_1\|_2 \in L_\infty$. From $\boldsymbol{s}_2$=$\boldsymbol{z}_2$, we can obtain that $\|\dot{\boldsymbol{s}}_2\|_2 \in L_\infty$. Because we've proved $\|\boldsymbol{s}_1\|_2 \in L_2$ and $\|\boldsymbol{s}_2\|_2 \in L_\infty$, according to Barbalat's lemma, we can have $\lim_{t\to\infty}\|\boldsymbol{s}_1\| = 0$ and $\lim_{t\to\infty}\|\boldsymbol{s}_2\| = 0$, i.e., the first-layer sliding surfaces, $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, converge to zero asymptotically, which proved that $\|\boldsymbol{s}_1\|_2 \in L_\infty$, $\|\dot{\boldsymbol{s}}_1\|_2 \in L_\infty$, $\|\boldsymbol{s}_2\|_2 \in L_\infty$, $\|\dot{\boldsymbol{s}}_2\|_2 \in L_\infty$

Since we've make sure that both first-layer sliding surfaces converge to zero in finite time, then it implies from (32) that both tracking errors $\mathbf{e}_1$ and $\mathbf{e}_2$ also approach zero as time goes to infinity. Consequently, we have $x_{1i} \to x_{1l}^* = 0$, $v_{1i} \to v_{1l}^* = 0$, $x_{2i} \to x_{2l}^*$, and $v_{2i} \to v_{2l}^*$ as $t \to \infty$. With the aforementioned design procedures and consensus analysis, the main result of the section is summarized as in the following theorem.

Theorem 1: *Given the underactuated fourth-order nonlinear ball-riding robot* (*IAS-BRR*) *model* (4), *network structure, and the assumption therein, the ORIT2FNN-based distributed consensus formation control law in* (27) *together with the parameter adaptive laws in* (31) *is designed such that* $x_{1i} \to x_{1l}^* = 0$, $x_{2i} \to x_{2i}^*$, $v_{1i} \to v_{1l}^* = 0$ *and* $v_{2i} \to v_{2i}^*$ *as* $t \to \infty$, *i=1, 2, …, n, where* x_{2i}^* *and* v_{2i}^* *are respectively the position and velocity given by the virtual leader, namely that all the n follower ballbot converge to their positions and velocities specified by the leader ballbot, and all the ballbots are maintained at their zero tilt angles.*

Remark 1: *From the decentralized formation control law in* (27), *it is easy to find the consensus-based cooperative formation control law for the* i^{th} *ball-riding robot* (*IAS-BRR*) *in formation in any either plane as follows:*

$$u_i = \frac{-g_{1i} - \lambda_i g_{2i} - \sum_{j=1}^{n} K_{ij}\dot{e}_{1j} + \lambda_i \dddot{f} + \lambda_i \sum_{j=1}^{n} S_{ij} b_j \ddot{\mathbf{x}}_L}{b_{1i} + \lambda_i b_{2i}} + \frac{-\lambda_i c_0 \dot{e}_{2j} - h_1 s_i - h_2 \operatorname{sgn}(\boldsymbol{s}_i) - \hat{\xi}_i}{b_{1i} + \lambda_i b_{2i}} \quad (40)$$

where

$$\begin{aligned} s_i &= z_{1i} + \lambda_i z_{2i} \\ &= v_{1i} + \sum_{j=1}^{n} K_{ij} e_{1j} + \lambda_i \left(v_{2i} + c_0 e_{2j} - \sum_{j=1}^{n} \overline{l}_{ij} b_j \ddot{\mathbf{x}}_L \right) \end{aligned} \quad (41)$$

where S_{ij} is the (*i*, *j*) *entry of the matrix* $\mathbf{S}_G$*. Note that the entry S_{ij} is not zero in the path of the communication topology; otherwise, S_{ij} is zero. Thus, with Assumptions* (i)-(iii)*, the control law in* (40) *is distributed and decentralized.*

By repeating the same proof process, the asymptotical stability of the complete formation system with the parameter updating rules using ORIT2FNNs are similarly obtained by choosing the two identical ORIT2FNNs and Lyapunov functions which are summed into one final Lyapunov function. This shows that all the *n* follower robots converge to their desired poses and pose velocities given by the leader ballbot. The main result is stated in the subsequent Theorem.

Theorem 2: *Given the decoupled fourth-order nonlinear multi-ball-riding robot system model in both sagittal and coronal planes in* (3)*, network structure and assumptions therein, the cooperative consensus-based formation control law* (27) *together with the same sets of parameter adaptive laws in* (31) *is designed in any either plane such that all the n follower robots converge to their desired pose and speeds given by the virtual leader, and the tilt angles of all the robots are maintained at zero.*

Since the control law design is completed, the distributed formation control law integrating with the collision and obstacle avoidance method modified from [7, 26] with a unified approach of the potential field function to achieving both collision and obstacle avoidance functions for the multi-ball-riding robot system, as shown in Fig. 6.

The overall repulsive potential $U(q_i)$ for the aims of collision and obstacle avoidance is defined as follows:

$$U(q_i) = U_{ca}(q_i) + U_{oa}(q_i) \quad (42)$$

where the repulsive potential of collision avoidance $U_{ca}(q_i)$ and obstacle avoidance $U_{oa}(q_i)$ are defined by

$$U_{ca}(q_i) = \begin{cases} \frac{1}{2}\left(K_{ca} \sum_{j=1, i\neq j}^{n} \left(\frac{1}{\rho(q_{ij})} - \frac{1}{\rho_0}\right)^2 \right. & \text{if } \rho(q_{ij}) \le \rho_0 \\ 0 & \text{if } \rho(q_{ij}) > \rho_0 \end{cases} \quad (43)$$

$$U_{oa}(q_i) = \begin{cases} \frac{1}{2} K_{oa} \sum_{k=1}^{M} \left(\frac{1}{\rho(q_{ik})} - \frac{1}{\rho_0}\right)^2 & \text{if } \rho(q_{ik}) \le \rho_0 \\ 0 & \text{if } \rho(q_{ik}) > \rho_0 \end{cases} \quad (44)$$

where the pair $q_i=(x_i, y_i)$ means the position of the i^{th} ball-riding robot; K_{oa} and K_{ca} are two scaling factors; $\rho(q_{ij})$ and $\rho(q_{ik})$ respectively denote the minimal distances from the i^{th} robot to the j^t robot and the k^{th} obstacle, namely that $\rho(q_{ij}) = \left\| q_i - q_j \right\|_2$ and $\rho(q_{ik}) = \left\| q_i - q_{obstacle}^k \right\|_2$, where $q_{obstacle}^k$ is the position of the k^{th} obstacle. Moreover, ρ_0 is the radius of the safe region, and *m* is the number of the obstacles.

TABLE II. Symbols Definitions.

Symbol and unit	Parameter and variable description	Typical values
m_b [*kg*] m_B [*kg*]	Mass of the [body/ball]	0.65 (Null), 1 (Filled) 3.2
l [m]	Distance between the centers of the ball and CoM of the body	0.15
r_b [*m*]	Radius of the ball	0.11
I_b [$kg{\cdot}m^2$] I_B [$kg{\cdot}m^2$]	Moment of inertia of the [ball/body] with respect to the [surface of the ball/CoM]	$I_b = \frac{2}{5} m_b r_b^2 = 0.005_{(Filled)}$ $I_B = m_B\left(l^2/6 + (l \cdot r_B)/3\right) = 0.032$
θ_x [rad], ϕ_x [rad]	[Tilting/rotating] angles	

The repulsive potential $U(q_i) \ge 0$ and tends to infinity when q_i gets closer to the j^{th} ball-riding robot or the k^{th} obstacle. Hence, the overall repulsive force $\mathbf{u}_{coi}$ is obtained from taking the negative gradient operation of the repulsive potential as in the following equation:

$$\mathbf{u}_{co_i} = -\nabla U(q_i) = \left[-\frac{\partial U(q_i)}{\partial q_i}^T, 0 \right]^T = \mathbf{u}_{ca_i} + \mathbf{u}_{oa_i} \in R^3 \quad (45)$$

where

$$\mathbf{u}_{ca_i} = \begin{cases} K_{ca} \sum_{j=1}^{n} \left(\frac{1}{\rho(q_{ij})} - \frac{1}{\rho_0}\right) \frac{1}{\rho^2(q_{ij})} \frac{q_i - q_j}{\rho(q_{ij})} & \text{if } \rho(q_{ij}) \le \rho_0 \\ 0 & \text{if } \rho(q_{ij}) > \rho_0 \end{cases}$$

$$\mathbf{u}_{oa_i} = \begin{cases} K_{oa} \sum_{k=1}^{M} \left(\frac{1}{\rho(q_{ik})} - \frac{1}{\rho_0}\right) \frac{1}{\rho^2(q_{ik})} \frac{q_i - q_{obstacle}^k}{\rho(q_{ik})} & \text{if } \rho(q_{ik}) \le \rho_0 \\ 0 & \text{if } \rho(q_{ik}) > \rho_0 \end{cases}$$

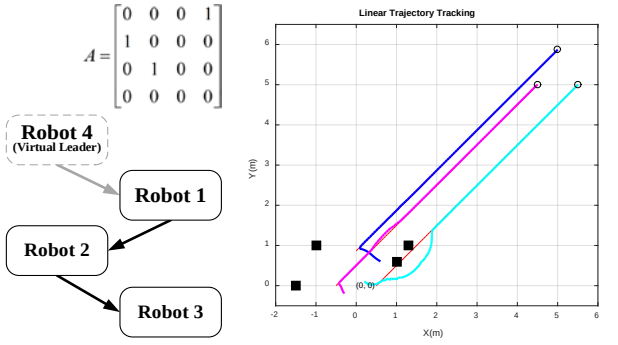
IV. SIMULATIONS AND DISCUSSION

The aims of simulations are to examine the effectiveness and performance of the proposed controller for formation control of three ballbots with uncertain terms. Two simulations are conducted for examining the ability of the proposed controller. Simulations parameters are listed in Table I, and the friction coefficients between the ball and the body and the ball and the ground are respectively set by $\mu_{\phi x}$=0.8 and $\mu_{c\phi x}$=0.8.

At the outset, two ORIT2FNNs with 3 inputs and 9 membership functions is chosen to estimate the uncertain vector functions in both *x* and *y* axes. Moreover, the initial values of the ORIT2FNN parameters are selected by:

w_i=1(i=1,…,6), σ_i=0.8(i=1,…,9),
c_{11} = [-0.09, -0.07], c_{12} = [-0.01, 0.01], c_{13} = [0.07, -0.09],
c_{21} = [-0.09, -0.07], c_{22} = [-0.01, 0.01], c_{23} = [0.07, -0.09],
c_{31} = [-0.09, -0.07], c_{32} = [-0.01, 0.01], c_{33} = [0.07, -0.09].

The first simulation is aimed to control the three IAS-BRRs to maintain the desired right triangular formation whose side lengths are 1m while tracking a straight line. The



(a) **(b)**

Fig. 7. Simulated line tracking results with formation keeping of the multi-robot team: (a) communication topology; (b) tracking responses.

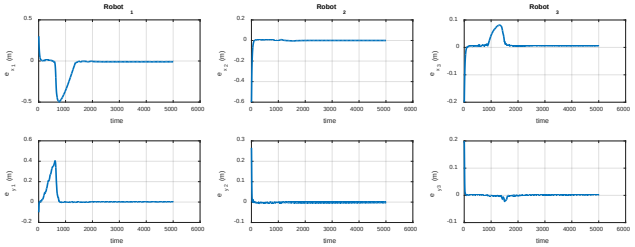


Fig. 8. Tracking errors of the IAS-BRRs.

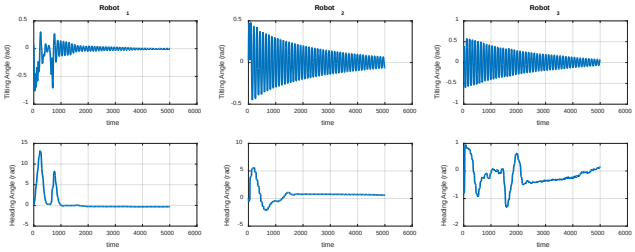
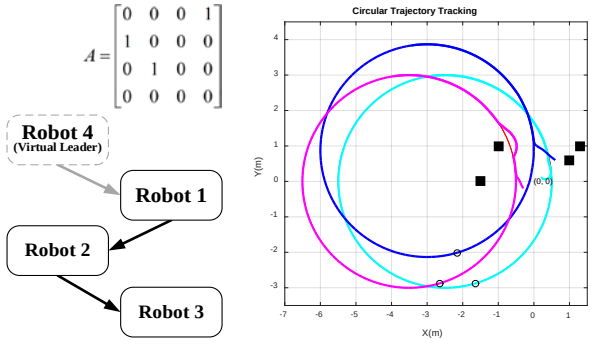


Fig. 9. Tilting and heading angles of the IAS-BRRs.



(a) **(b)**

Fig. 10. Simulated circular tracking results with formation keeping of the multi-robot team: (a) communication topology; (b) tracking responses.

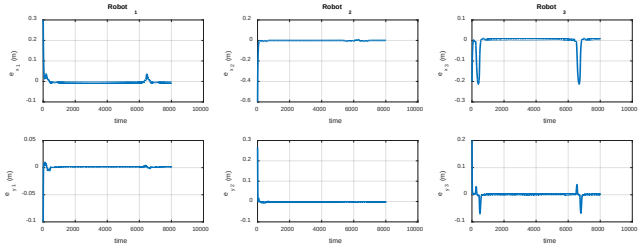


Fig. 11. Tracking errors of the IAS-BRRs.

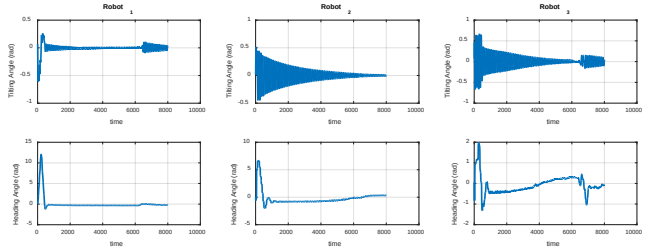


Fig. 12. Tilting and heading angles of the IAS-BRRs.

initial positions of the three IAS-BRRs are (0.2, 0), (0.4, 0.5) and (-0.2, 0) (unit: *m*), respectively, and the virtual leader moves along a line with speeds v=0.1 m/sec being in both axes; the simulation time t=50 seconds. The communication topology and the formation control trajectories of the three IAS-BRRs are displayed in Fig. 7, where the black block stands for the static obstacles. Fig. 8 & 9 depicts the tracking errors and tilting and heading angles of this simulation. The results reveal that the proposed formation controller is effective in achieving the desired formation.

The second simulation is dedicated to test whether the developed distributed formation controller is able to achieve the same triangular formation where the virtual leader moves along a circle with a radius of 3 m and an angular speed, ω=0.2 rad/sec and a simulation time of 80 seconds. Three IAS-BRRs get started at the same initial position as in the first simulation. Fig. 10 displays the simulated circular formation trajectories of the three IAS-BRRs and Fig. 11 & 12 depict the tracking errors, tilting and heading angles of each IAS-BRR. Through these results in Fig. 10-12, the proposed distributed formation controllers is shown effective in controlling all the IAS-BRRs to keep triangular formation with circular trajectory tracking.

From the above-mention simulation results shown in Fig. 7&10, it is obvious to observe that the proposed decentralized consensus control law is shown capable of manipulating all the IAS-BRRs, modeled by a fourth-order underactuated system model, to accomplish formation keeping and desired trajectory tracking in both cases of line and circular movements of the leader. As can be seen in Fig. 9 & 12, all the tilt angles of three IAS-BRRs are almost maintained nearly at the region of the origin, indicating that all the IAS-BRRs moves without falling down, but have significant and long transient responses. The responses can be improved by adjusting some control gains of the proposed controller.

V. Conclusions

This chapter has presented a consensus-based cooperative formation control using hierarchical sliding-mode control and output-feedback recurrent interval type-2 fuzzy neural networks (ORIT2FNN) for a group of homogenous, networked mobile IAS-BBRs with modeling error and exogenous uncertainties. Each IAS-BRR can be modeled by two decoupled, uncertain, and independent underactuated fourth-order dynamic models. Based on the IAS-BRR and multi-robot system models, and the communication topology which is described by a directed graph, we proposed an intelligent adaptive distributed consensus formation control law with ORIT2FNN approximation. Via the Lyapunov stability theory, not only the overall system stability can be

guaranteed, but also the desired formation pattern can be ensured. The effectiveness and merits of the proposed control method have been demonstrated via the simulation results. In conclusion, the proposed consensus formation controller works well with our multiple IAS-BRRs. An important topic for future research would be conducting the experiment and comparing the experiment result to the simulation results in order to verify the usefulness and applicability of the proposed consensus formation controller.

REFERENCES

[1] R. Murray, "Recent Research in Cooperative Control of Multivehicle Systems," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 129, pp. 571-583, 2007.
[2] L. E. Parker, "Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems," *Journal of Physical Agents*, vol. 2, no. 1, pp.1-14, March 2008.
[3] Y. H Chang, C. W. Chang, C. L. Chen, and C. W. Tao, "Fuzzy sliding-mode formation control for multirobot systems: design and implementation," *IEEE Transactions on Systems Man and Cybernetics-Part B-Cybernetics*, 42(2), pp. 444-457, April 2012.
[4] Y. Kuriki and T. Namerikawa, "Formation Control of UAVs with a Fourth-order Flight Dynamics," in *Proc. of IEEE 52nd Annual Conference on Decision and Control (CDC 13)*, pp. 6706-6711, 2013.
[5] Y. Kuriki and T. Namerikawa, "Consensus-based Cooperative Formation Control with Collision Avoidance for a Multi-UAV System," in *Proc. of 2014 American Control Conference* (ACC), Portland, Oregon, USA, pp. 2077-2082, June 4-6, 2014.
[6] Y. Cao and W. Ren, "Optimal Linear-Consensus Algorithms: An LQR Perspective," *IEEE T. On Systems, Man, And Cybernetics - Part B: Cybernetics*, Vol. 40, No. 3, pp. 819-830, 2010.
[7] W. Ren et al., "Information consensus in multivehicle cooperative Control," *IEEE Control Systems magazine*, Vol. 27, No. 2, pp. 71-82, 2007.
[8] Z. Meng, W. Ren et al., "Leaderless and Leader-Following Consensus With Communication and Input Delays Under a Directed Network Topology," *IEEE T. on Systems, Man, and Cybernetics*, Vol. 41, No. 1, pp. 75-88, 2011.
[9] C. L. Philip Chen, C. E. Ren, and T. Du, "Fuzzy observed-based adaptive consensus tracking control for second-order multi-agent systems with heterogeneous nonlinear dynamics," *IEEE Trans. on Fuzzy Syst.*, Vol.24, No.4, pp. 906-915, Aug. 2016.
[10] C. C. Tsai, H. L. Wu, F. C, Tai, Y. S. Chen, , "Adaptive backstepping decentralized formation control using fuzzy wavelet neural networks for uncertain Mecanum-wheeled omnidirectional multi-vehicles," in *Proc. of the 2016 IEEE Intern. Conf. on Industrial Technology*, Taipei, Taiwan, 15~17 March, 2016.
[11] C. Chen, C.-E Zen, and T. Du, "Fuzzy observed-based adaptive consensus tracking control for second-order multi-agent systems with heterogeneous nonlinear dynamics," to appear in *IEEE Transactions on Fuzzy Systems*, 2015.
[12] 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.
[13] 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.
[14] http://rezero.ethz.ch/project_en.html [online] 2011-07.
[15] C. C. Tsai, C. K. Chan, 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.
[16] C. M. Lin and Y. J. Mon, "Decoupling Control by hierarchical fuzzy sliding-mode controller, "*IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 593-598. July 2005.
[17] 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.
[18] D. Qian, J. Yi, and D. Zhao, "Hierarchical sliding mode control for a class of SIMO under-actuated systems, "*Control and Cybernetics*, vol. 37, no.13, pp.159-175. 2008.
[19] Q. Liang and J. M. Mendel, "Interval type-2 logic systems: Theory and design," IEEE Trans. Fuzzy Syst., vol. 8, pp.535-550, 2000.
[20] C. Wang, C. Cheng and T. T. Lee, "Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 34, no. 3, pp. 1472-1477, June 2004.
[21] C. K. Chan, C. C. Tsai, "Intelligent Backstepping Sliding-Mode Control Using Recurrent Interval Type 2 Fuzzy Neural Networks for a Ball Robot with a Four-Motor Inverse-Mouse Ball Drive," in *Proc. of SICE 2012 Annual conference*, Akita, Japan, pp. 1281 – 1286, 20-23 August, 2012.
[22] C. K. Chan ,C. C. Tsai, "Direct Adaptive Recurrent Interval Type 2 Fuzzy Neural Networks Control for a Ball Robot with Four-Motor Inverse-Mouse Ball Drive," *Proc. of 2013 International Conference on Advanced Robotics and Intelligent Systems (ARIS 2013)*, Tainan, Taiwan, 31 May-2 June, 2013.
[23] C. C. Tsai, T. Y. Lin, and F. C. Tai, "Intelligent Distributed Exponential Consensus Formation Control Using ORIT2FNN for Uncertain Networked Heterogeneous Mecanum-Wheeled Omnidirectional Multirobots," in *Proc. of 2017 International Conference on Advanced Robotics and Intelligent Systems (ARIS 2017)*, Taipei Nangang Exhibition Center, Taipei, Taiwan, September 6-8, 2017.
[24] H. K. Khalil, *Nonlinear systems*, 2nd ed., Prentice Hall, 1996.
[25] C. K. Chan, C. C. Tsai, "Direct Adaptive Robust Motion Control for Uncertain Ball-Riding Robots Using Recurrent Interval Type 2 Fuzzy Neural Networks," in *Proc. of SICE 2013 Annual conference*, Nagoya University, Nagoya, Japan, 14-17, September, 2013.
[26] C. C. Tsai, H. L. Wu, F. C. Tai, and Y. S. Chen, "Distributed Consensus Formation Control with Collision and Obstacle Avoidance for Uncertain Networked Omnidirectional Multirobot Systems Using Fuzzy Wavelet Neural Networks," *International Journal of Fuzzy Systems*, Vol. 19, No. 5, pp. 1375-1391, 2017.

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.

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 an IEEE Fellow, an IET Fellow and a CACS Fellow.

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, 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.

Cheng-Kain Chan received the B.S. degree in Electrical Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, ROC, in 1994, and both the M.S. and Ph.D. degrees in Electrical Engineering from National Chung Hsing University, Taichung, Taiwan, ROC, in 2004 and 2013, respectively. His current research interests include mechatronics, nonlinear control, adaptive control and their applications to industrial processes and machines.

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 1 MARCH, 2018

CONTENTS

REGULAR PAPERS

TECHNICAL NOTE