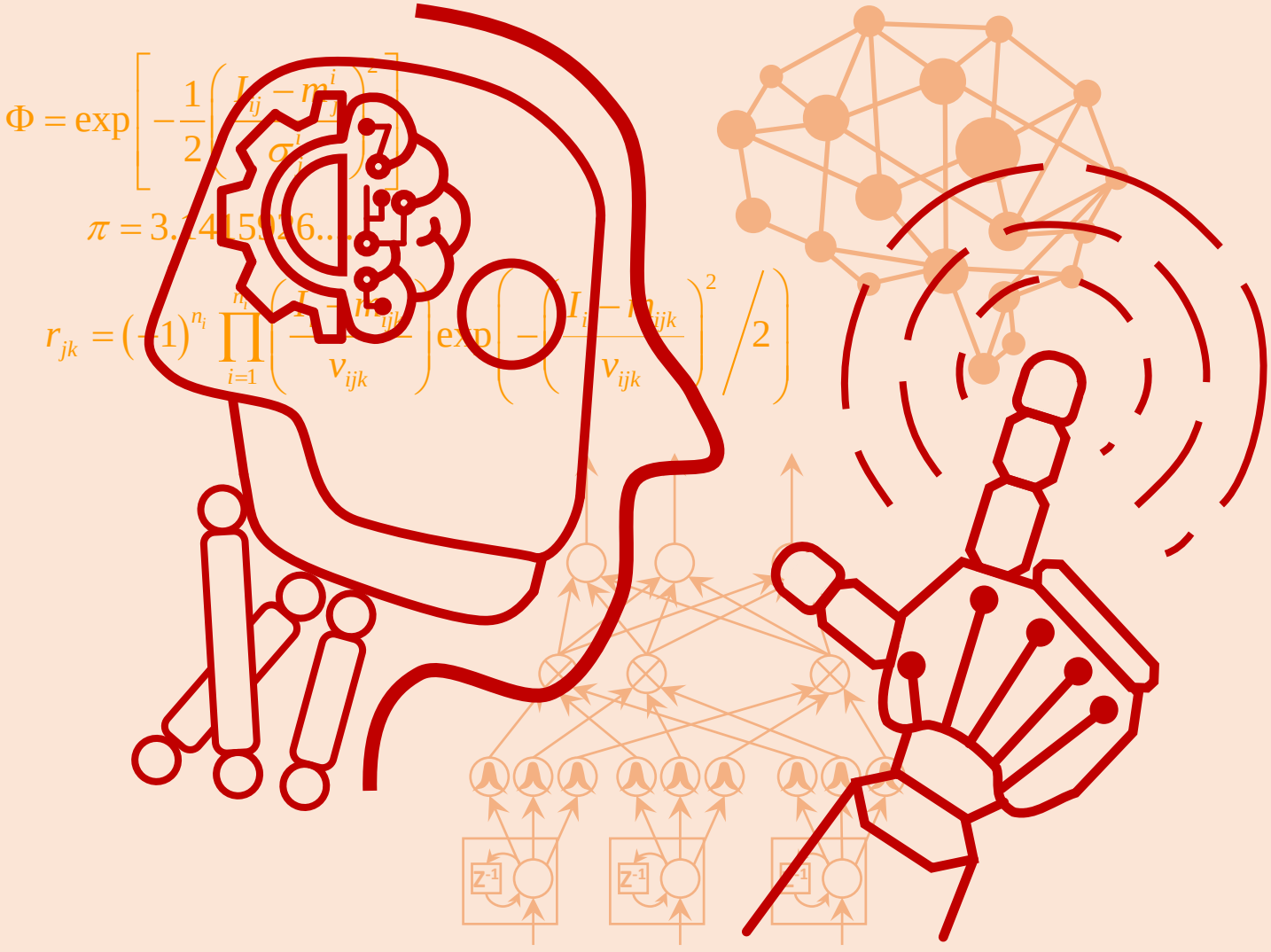
ISSN 2616-8170

RST
Robotics Society of Taiwan

# iRobotics

VOLUME 1
NUMBER 2
JUNE 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.

# Modeling and Implementation of Quadruped Robot with Compliant Mechanisms

Chen-Yu Chan and Yen-Chen Liu, *Member, RST*

***Abstract*—In this paper, a small-sized quadruped robot is introduced by utilizing compliant mechanisms to accomplish walking, turning, and jumping motions. The design of cable-driven legs with a tension spring on the knee joint can provide robust, stable, and energy-efficient gaits from the compliant legs. The kinematic and dynamic models of the compliant legs and jumping mechanisms are presented. Tort-like walking motion and turning movement can be achieved by design of foot trajectory and regulating step length during the walking motion. Moreover, energy-storing mechanism is designed for the quadruped robot to have the ability of jumping with various height and distance. Simulation results with the dynamic model were presented to design the walking and jumping motion of the quadruped robot. The proposed design and system are subsequently validated via experiments to show the efficacy of the quadruped robot in talking, turning, and jumping motions.**

***Index Terms*—Quadruped robot, compliant mechanism, walking and jumping robot, motion control, mechatronics.**

## I. Introduction

Resulting from the increased applicability, the study of design, manufacture, and control of mobile robots has become a significant research topic. Mobile robots can assist human to not only conduct dangerous tasks, but also engage in environment exploration, transporting goods, rescuing mission, and surveillance. Wheeled robots have high moving speed on flat plane that makes wheeled robots more suitable to work for indoor tasks within a flat workspace. However, such robotic systems are not useful and applicable on uneven or un-constructed environment. Therefore, the development of walking robots or legged robots are required in a variety of applications and environments.

In contrary to wheeled mobile robot, walking robots have higher flexibility and ability to move on uneven terrains stably by using leg with multiple degrees-of-freedom [2]–[4]. Legged robots are significantly superior to wheeled robots to move on unknown environments such as battle field, disaster scene, forest, and outer space [5], [6]. Owing to the mobility and ability of moving on uneven terrain, legged robots have been studied extensively [7], [8]. The research topics contain foot trajectory [8], [9], mechanism of legs [10], [11], command of generating trajectory [12], [13], and improving efficiency [6], [14]. In addition to walking motion, running, making turns, and jump have also attracted significant attentions from researchers in robotics and control societies [5], [15]. A hierarchical controller was proposed for MIT Cheetah to achieve a highly dynamic running gait [16]. Moroever, Béier curve and a tunable amplitude sinusoidal wave were considered with a leg-trajectory generator to generate gait pattern for proprioceptive legs. A design concept based on a tendon-bone and co-location architecture was proposed in [17] to generate efficient running by improve ground contact and lift-off performance.

Recently, Cheetah-cub, a compliant quadruped robot, has been developed by using compliant legs to achieve high-speed locomotion on flat terrain [3]. The legs of Cheetah-cub, named Advanced Spring Loaded Pantograph leg (ASLP-leg), have diagonal compression spring and a spring used to replace one rigid linkage on pantograph mechanism, which is driven by cable. Cheetah-cub can speed up to $1.42$m/s and has a Froude number of $1.30$. The control of quadruped robot in trotting gain is proposed in [18] with the use of 5-DOF legs. Researchers in [4] adopt a simplified model of a quadruped robot to estimate robot dynamics and develop an algorithm to vary the robot's running speed. By controlling the impulse profile of legs, the transformed joint torques can stabilize the robot and regulate forward velocity of the robot. Moreover, a variety of actuation has been considered for quadruped robots to achieve walking and running motions [6], [7], [11], [13].

Although legged robots have excellent ability of moving on uneven terrain, it still have difficulty of crossing over large obstacles, which is highly dependent on the size of robot legs. In the natural, jumping motion is an intuitive method for animals to cross over large obstacles. By considering the locomotion efficacy (height per gait) and energy efficiency, jumping can provide the best tradeoff among various locomotion methods such as walking, running, or wheeled locomotion [19]. There are many types of jumping robots that have been presented recently [5], [15], [20]–[23]. A one-motor legged robot has been proposed in [21] to achieve jumping with the ability of adjusting jumping direction. A jumping-crawling robot was proposed by using latex rubber and knee-like joints to utilize large displacement of the sotring energy in the elastic material [22]. Mammal-like mechanical design of the compliant robotic leg has been presented with the elastic tendon for storing energy in jumping motion [5]. The performance of compliant leg design has been tested for impact attenuation for airdrop landings [15]. The robot, designed by Festo, uses harmonic drives and pneumatic pistons for actuating its legs which have elastic tendons [23]. A feedforward control is proposed to generate the desired trajectories with a feedback

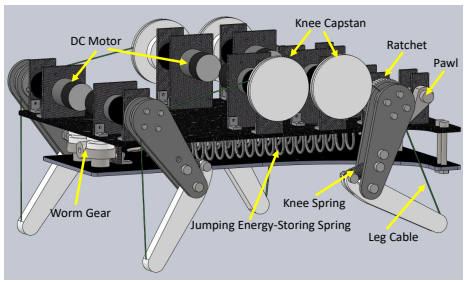
This work was supported in part by the Ministry of Science and Technology, Taiwan, under grants MOST 105-2221-E-006-160-MY3 and MOST 106-2221-E-006-010-MY3, and in part by the Headquarters of University Advancement at the National Cheng Kung University, which is sponsored by the Ministry of Education, Taiwan. The preliminary results of this paper have been presented previously in IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2016 [1].

C.-Y. Chan is with Compal Electronics, Taipei 11492, Taiwan, and Y.-C. Liu is with the Department of Mechanical Engineering, National Cheng Kung University, Tainan 70101, Taiwan, e-mail:{s91402001@gmail.com, yliu@mail.ncku.edu.tw}.

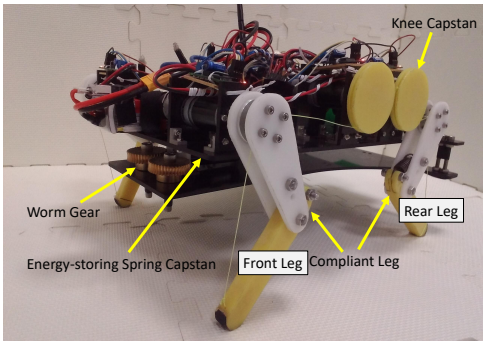
control to stabilize the robot when it is hopping.

Walking, turning, and jumping are three main requirements in the development of quadruped robots. For those robots, equipped with legs more than two degrees-of-freedom, walking and turning motion are easy to achieve. However, more actuators imply more robot weight and more power consumption, which will obstruct robots to perform jumping movement. In this paper, we designed a quadruped robot, which has the ability of walking, turning, and jumping by using two-segments compliant legs and jumping energy storing mechanism. With the proposed walking trajectory, the robot can move straight, and turn by changing the walking step lengths. In addition to compliant legs, the proposed quadruped robot has also equipped with jumping mechanisms that can provide different stretching force for the robot to jump with various heights. Dynamic model of the robot is derived and the model is used to simulate jumping motion of the robot in Simulink®. Moreover, the proposed quadruped robot is built to test the ability of walking and turning with our turning control method. Experimental results of the robot in walking, tuning, and jumping motions are presented and investigated.

This paper is organized as follows. The design and modeling of the proposed quadruped robot are described in Section II. The motion control for walking, turning, and jumping motions are introduced in Section III, which is followed by simulation and experimental results addressed in Section IV. Conclusions and future work of this research are summarized in Section V.




(a) Schematic plot of the quadruped robot




(b) Appearance of the proposed quadruped robot

Fig. 1: Design of the quadruped robot for walking, turning, and jumping motions.

## II. Design and Modeling of Quadruped Robot

For the sake of developing a quadruped robot with the capability of walking and jumping, we introduce a compliant-based design for legs. A two-segment compliant mechanism, driven by DC geared motor and cable, is adopted in the design to perform quadruped motion of the robot. Moreover, the robot is equipped with two energy-storing mechanisms, which are utilized to provide potential energy for the robot to accomplish jumping motion. The schematic plot and the appearance of the proposed quadruped robot are shown in Fig. 1. The detailed design and modeling of the proposed quadruped robot are addressed in this section.

### *A. Compliant Legs*

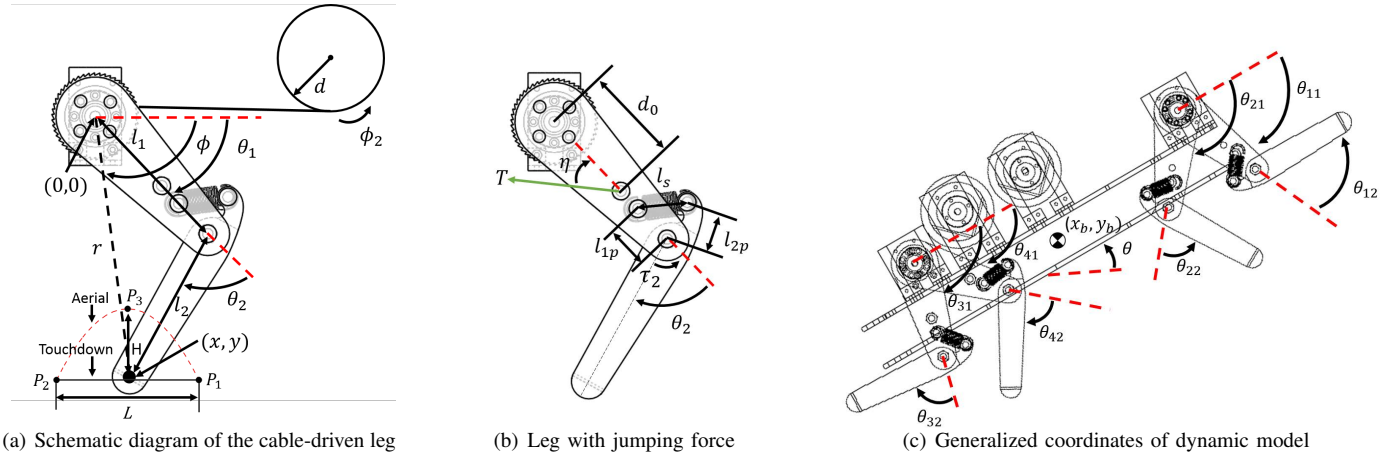
It has been demonstrated that compliant legs can provide robust, stable and energy efficient gait of robot [5], [15], [24]. Moreover, the tension spring on the knee joint can also prevent ground reaction force after the robot touches down from jumping motion. Therefore, the quadruped robot developed in this paper is based on the design of compliant mechanisms. The design of two-segment compliant mechanism for the rear legs is shown in Fig. 2, where $l_1$ and $l_2$ are the length of thigh and calf, $\phi_2$ is the rotational angle of the capstan, $d$ is the radius of the capstan, $d_0$ is the distance from the point of force for jumping spring to the hip motor, $r$ is the distance between hip to the tiptoe.

Both the front and rear legs have two-degree-of-freedoms $(\theta_1, \theta_2)$, where the hip joint is driven by a DC geared motor directly, and the knee joint is controlled by a cable and capstan. A tension spring is installed on the knee joint of the leg to provide compliant motion, as shown in Fig. 2 (a). The knee joint is controlled by a cable-driven mechanism, operated by a motor mounted on the body of the robot. The foot tip is connected to the driving motor via a cable, so that the knee angle, $\theta_2$, can be regulated. Therefore, the position of the foot tip, $(x, y)$, can be fully actuated via hip motor and cable line. In addition to the aforementioned actuators, there is a cable mounted to the middle of the thigh of the rear legs at the point of force $T$ in Fig. 2 (b). This cable is connected to the energy storing and releasing mechanism that is utilized to provide jumping force of the quadruped robot. The force $T$ is generated by jumping energy-storing spring as shown in Fig. 1 (a). More details will be discussed in the section of jumping mechanism.

To control the motion of tiptoes, the inverse kinematics from the foot trajectory $(x, y)$ to the joint angles $\theta_1, \theta_2$, and $\phi$ are necessary. Let's take the rear leg as an example which is shown in Fig. 2 (a). In the following analysis, we assume that the cable from hip to the capstan is parallel to the walking surface, and the origin of the coordinate is located at the shaft of the hip motor. By giving a desired tiptoe position $(x_d, y_d)$, the relationship between the angle of hip joint and the corresponding joint angular positions can be calculated by

$$\phi_d = atan2(y_d, x_d), \tag{1}$$

where $\phi_d$ is the required angle for $(x_d, y_d)$. From the geometry of the leg, we have the desired angle for thigh $\theta_{1d}$ and calf




(a) Schematic diagram of the cable-driven leg (b) Leg with jumping force (c) Generalized coordinates of dynamic model

Fig. 2: Design of the compliant legs with the jumping force from the stretching spring.

$\theta_{2d}$ such that

$$\theta_{1d} = \phi_d + cos^{-1}\left(\frac{r_d^2 + l_1^2 - l_2^2}{2r_d l_1}\right), \tag{2}$$

$$\theta_{2d} = cos^{-1}\left(\frac{l_1^2 + l_2^2 - r_d^2}{2l_1 l_2}\right) - \pi, \tag{3}$$

where $r_d = \sqrt{x_d^2 + y_d^2}$. Subsequently, we obtain the rotational angle of the capstan that

$$\phi_{2d} = \frac{r_0 - r_d}{d}, \tag{4}$$

where $\phi_{2d}$ is the desired angular displacement of knee joint motor, $d$ is radius of knee capstan, and $r_0$ is distance between hip and foot when it is fully extended. Consequently, for a given desired tiptoe position/trajectory $(x_d, y_d)$, the corresponding actuator outputs for thigh, calf, and hip can be generated from the aforementioned equations to provide $\theta_{1d}, \theta_{2d}$, and $\phi_{2d}$.

For hip motors, the requirement of motor torque should take into account the influence of tension spring mounted on the knee. Since the cable goes through the hip of the leg, the thigh angle $\theta_1$ is independent to the length of cable. Thus, the relationship between the cable length to the knee-angle difference can be obtained by

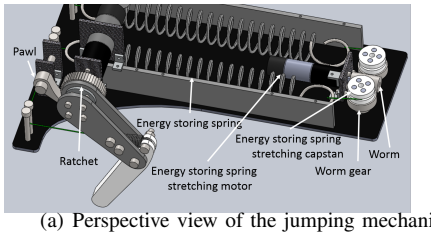
$$r^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos(\pi - \theta_2) \tag{5}$$

By denoting $r = r_0 + \delta r$ and $\theta_2 = \theta_{20} + \delta\theta_2$ with $r_0$ and $\theta_{20}$ are the initial length and angle of the cable and knee spring. We get

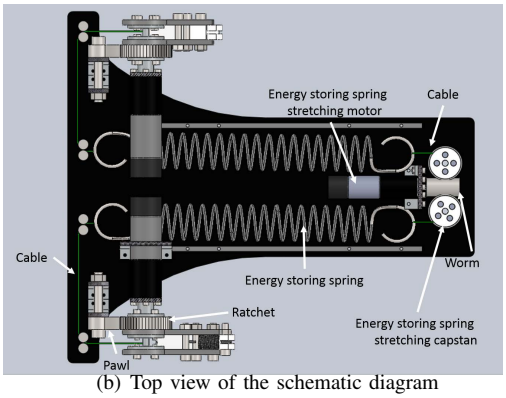
$$\delta r = -\frac{l_1 l_2 \sin(\pi - \theta_2)}{\sqrt{l_1^2 + l_2^2 - 2l_1 l_2 \cos(\pi - \theta_2)}}\delta\theta_2. \tag{6}$$

By denoting $T_c$ as the tension force of the cable and $\tau_k$ as the torque applied on the knee from the spring force, we have $T_c \delta r = \tau_k \delta\theta_2$. As the radius of capstan is $d$, the motor torque of the capstan $\tau_c$ can be given as $T_c = \tau_c d$. Hence, (6) can be further written as

$$\tau_c = -\frac{\sqrt{l_1^2 + l_2^2 - 2l_1 l_2 \cos(\pi - \theta_2)}}{2dl_1 l_2 \sin(\pi - \theta_2)}\tau_k. \tag{7}$$




(a) Perspective view of the jumping mechanism




(b) Top view of the schematic diagram

Fig. 3: Design of jumping mechanism of the rear legs in the proposed quadruped robot.

With the selection of length for thigh $l_1$ and calf $l_2$, radius of capstan $d$, and angle of knee $\theta_2$, the design of cable length and capstan motor can be obtained from the aforemention results. It is observed that $\tau_c$ is independent to the angle of thigh as $\theta_1$ is controlled by the hip motor.

### *B. Jumping Mechanism*

There are various kinds of design that can provide quadruped robotic system to accomplish jumping motion, such as pneumatic actuator, motor, spring, and engine [6], [11], [13]. However, if the power-to-weight ratio is too low, then the jumping motion will require a tremendous energy to achieve a desired jumping performance, height, and distance. Since compliant legs have been tested with good performance in

TABLE I: Specification and parts list of the robot

| Part | Specification |
|---|---|
| DC-Motor (Leg) | SHA YANG YE IG30-GM 12V reduction ratio1/139 |
| DC-Motor (Stretching Spring) | SHA YANG YE IG22-GM 12V reduction ratio1/136 |
| Battery | Lipo 14.8V 2250mAh 35C |
| Knee Spring | SPEC T32330 |
| Energy-Storing Spring | SPEC T32870 |
| Worm | MISUMI WGEAH-1.0-30-1-6-R |
| Worm Gear | MISUMI WGEAU-1.0-1-6-R |
| Ratchet | KHK SRT1-50 |
| Pawl | KHK SRT1-C |
| Length of Thigh | 93 mm |
| Length of Shank | 93 mm |
| Micro-Controller | Arduino Due |
| Gyroscope | MPU-6050$^{TM}$ |
| Communication | BluetoothHC-05HC-06 |
| Net Weight | 5.7 kg |
| Weight of Thigh | 0.15 kg |
| Weight of Shank | 0.04 kg |

jumping [5], [15], in this paper we utilize tension spring, which is an easy to access component for storing energy with preferable energy-efficiency. From the observation of biomechanics, thigh plays an important role on jumping and muscles of thigh contribute to generating jumping power [25]. Thus, the designed jumping mechanism is only connected to the rear legs for providing energy (the jumping steps will be addressed in the next section). The front legs will use the compliant mechanism and knee motor to push the robot's body preparing for jumping posture. The perspective view and schematic diagram of the jumping mechanisms are illustrated in Fig. 3.

As the stretching/compressing velocity of tension springs can be slow, we decided to use high gear ratio motor, which has less weight and is sufficient to stretch springs. In [26], the researchers used a set of screw and nut for stretching springs and storing energy for jumping. Thus, we considered a capstan mechanism in our design to stretch the energy-storing springs. In order to fix the hip joint while the jumping springs are stretching, there are two pairs of ratchet and pawl installed on the rear hip joints. Before the jumping motion, the ratchet is locked by a pawl so that the spring force does not exert on the thighs of the robot's rear legs. Moreover, there is a motor connected to the set of worm and worm gears, which have two capstans that are connected to energy storing springs. Therefore, the stretching force in the energy storing spring can be transmitted to the rear thighs after the pawl releasing the ratchet.

Before the jumping motion, the energy-storing motor rotates to stretch the energy-storing spring so that the required jumping-energy can be stored in the springs. At the meantime, the DC motor for rear legs also rotates to pull the cable connected to the tiptoe; thus, the knee spring is also stretching to store jumping energy. The equivalent torque on the hip joint should be obtained from the energy-storing spring mounted on the robot's body for the study of jumping performance. The forces applied on the rear legs are illustrated in Fig. 2 (b). By denoting $F_2$ as the spring force from the knee spring and $\tau_2$ as the equivalent torque, we get the equation from the principle of virtual work that

$$F_2 \delta l_s + \tau_2 \delta\theta_2 = 0, \tag{8}$$

where the spring force can be given as $F_2 = K_{knee,s}(l_{s0} - l_s)$ with $K_{knee,s}$ as the spring coefficient, $l_s$ as the length of the knee spring, and $l_{s0}$ as the initial length of the knee spring. From the rear leg's configuration, we have that $l_s = \sqrt{l_{1p}^2 + l_{2p}^2 - 2l_{1p}l_{2p}\cos\theta_2}$. Thus, we further have

$$\delta l_s = \frac{l_{1p}l_{2p}\sin\theta_2}{\sqrt{l_{1p}^2 + l_{2p}^2 - 2l_{1p}l_{2p}\cos\theta_2}}\delta\theta_2. \tag{9}$$

Consequently, we get that

$$\tau_2 = \frac{K_{knee,s}l_{1p}l_{2p}(l_s - l_{s0})}{l_s}\sin\theta_2. \tag{10}$$

The torque applied on the rear legs from the jumping spring can also be obtained similarity. As shown in Fig. 2 (b), by denoting $T$ as the tension force from the jumping spring, $d_0$ as the distance from the point of jumping cable to hip joint, $\eta$ as the angle between jumping cable and thigh, the equivalent torque from the jumping spring $\tau_{jum}$ can be given as

$$\tau_{jum} = -Td_0\sin\eta. \tag{11}$$

It can be observed that the angle $\eta$, which relates to the posture of the quadruped robot before jumping, can be adjusted to provide various jumping torque. With the design of the compliant legs and the energy-storing spring for jumping, the motor torque and spring force are applied parallelly on the rear legs. Hence, the force can be added up to the rear legs for providing energy during jumping motion. It is noted that the force from jumping spring has no influence to the compliant legs during walking. With the design and modeling of compliant legs, the jumping motion and control will addressed in the following sections.

### C. Other Components

As the quadruped robot is designed in the size of small dogs, the motor system and springs should be selected accordingly. By considering the size and standing height of the robot, the spring constant of knee springs are chosen to be $94.24$N/cm, and the maximum stretching length of the knee spring is $0.0527$m. The maximum stretching length of the energy storing spring is $0.3$m which is suitable for the size of the robot. Furthermore, the spring constant of the energy storing spring is selected as $1,206$N/m.

In the quadruped robot, the rated torque of hip joint and knee joint motors are $0.785$N-m and the rated speed is $45$rpm. The motors are chosen by considering the swing velocity of leg. The rated torque of energy storing spring stretching motor is $0.363$N-m and the rated speed is $53$rpm. The reduction ratio between worm wheel and worm is $1/30$. The permissible

bending moment of ratchet and worm wheel are 14N-m and 3.38N-m, respectively. The value of the permissible bending moment of ratchet, worm wheel, and reduction ratio between worm wheel and worm are chosen by the maximum tension of the energy storing spring. The specification and part lists of the proposed quadruped robot are summarized in Table I.

### *D. Dynamic Modeling*

The dynamic model developed for the quadruped robot is based on Euler-Lagrange equations. By assuming that the robot can only move on sagittal plane in simulations, the definition of generalized coordinate is shown in Fig. 2 (c). Therefore, the dynamic model can be represented by $M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J^T F_{ext}$, where $M(q)$ is inertia matrix, $C(q,\dot{q})$ is the terms of centripetal force and Coriolis force, $G(q)$ is the vector of gravitational force, $\tau$ is the generalized forces, $J$ is Jacobian matrix of the robot, and $F_{ext}$ is reaction and friction forces between feet and the ground. The generalized coordinates is $q = [x_b, y_b, \theta, \theta_{11}, \theta_{12}, \theta_{21}, \theta_{22}, \theta_{31}, \theta_{32}, \theta_{41}, \theta_{42}]$ where $x_b, y_b$ are the center of mass of the main body, $\theta$ is the pitch angle of main body, $\theta_{11}$ to $\theta_{42}$ are the relative angular position of each joint. More details of $M(q)$, $C(q,\dot{q})$, and $G(q)$ are referred to [27]. The model will be utilized in Section IV to simulate the walking and jumping motions of the proposed robot.

## III. Motion Control

### *A. Walking and Turning Motions*

The desired trajectory of foot is showed in the tiptoe of Fig. 2 (a), where $H$ is the desired step height, $L$ is the desired step length, and $(x, y)$ is the coordinate of the tip position. The trajectory contains a straight line and a parabolic curve determined by points $P_1, P_2$ and $P_3$. We expect the turning method used in this paper can be applied on all foot trajectory with adjustable step length. Therefore, the step length of trajectory that we adopted has to be intuitive and easily changed. The foot trajectory is shown in Fig. 4, where the straight line is the desired trajectory of touchdown phase and the parabolic curve is the desired trajectory of swing (aerial) phase.

In this paper, the proposed quadruped robot is designed to take trotting motion, where the diagonal pairs of legs forward at the same time during walking. Fig. 5 is the timing chart of walking for all legs, where $i = 1, 2, 3, 4$ indicate the front left leg, the front right leg, the rear left leg, and the rear right leg, respectively. In Fig. 5, the dotted lines indicate the time period of swing phase $T_a$, and the solid lines means the touchdown phase $T_d$ of foot tip. By using the trot-like waking pattern, the walking sequence for a period is shown in Fig. 6. Consequently, the design of foot trajectory and timing chart for walking gait can be transformed to actuators outputs by inverse kinematics, as addressed in Section II, to ensure trotting motion for the proposed robot.

The turning method for the quadruped robot can be accomplished by changing the step length $L$ in the foot trajectory. While the robot goes straight, the distance during stance phase is the same for all legs. However, if the robot makes turns, each

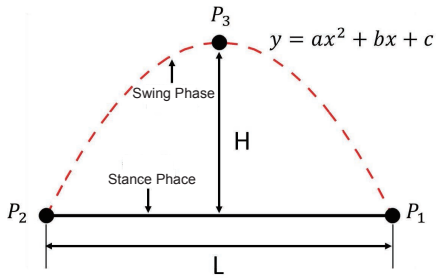



Fig. 4: Foot trajectory for the quadruped robot developed in this paper.

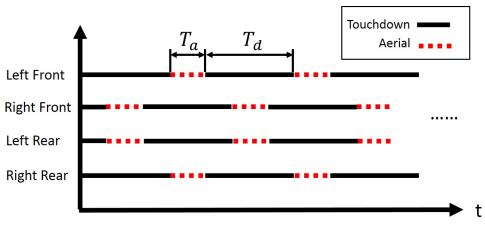



Fig. 5: Timing chart for walking gait. (Dotted lines represent swing phase and solid lines represent stance phase)

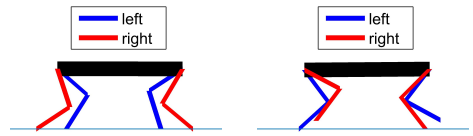
step length, $L_i$, of leg are considered to be regulated by

$$L_i = L_d + K_{pi}(\psi_d - \psi),\ i = 1, 2, 3, 4 \tag{12}$$

$$K_{pi} = \begin{cases} S(i)\frac{\delta L_{max}}{|\psi_d - \psi|} & \text{if } |(\psi_d - \psi)| > \theta_r \\ S(i)\frac{\delta L_{max}}{\theta_r} & \text{otherwise} \end{cases} \tag{13}$$

$$S_i = \begin{cases} -1 & \text{if } i = 1, 3 \\ 1 & \text{otherwise} \end{cases} \tag{14}$$

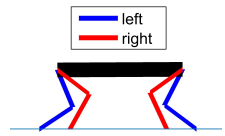
where $L_d$ is nominal stance distance, $K_{pi}$ is a control gain, $\psi_d$ is the desired body orientation, $\psi$ is the current robot orientation, and $\theta_r$ is a positive constant, which is relative to the acceptable error in turning orientation. Fig. 7 shows the flow chart of turning control, which represents the relationship between the motion control for walking and turning addressed in this subsection.



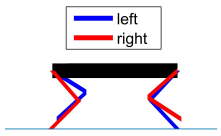

(a) Four legs in stance phase




(b) Front left and rear right legs in swing phase




(c) Four legs in stance phase




(d) Front right and rear left legs in swing phase

Fig. 6: Walking sequence of the quadruped robot.

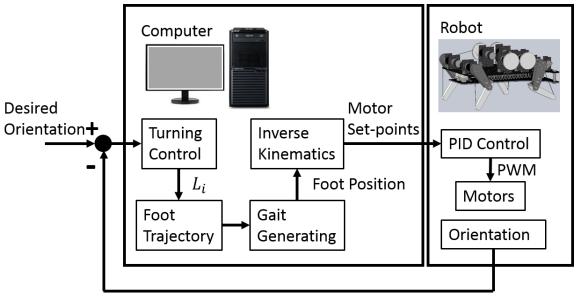



Fig. 7: Flow chart of turning control.

### B. Jumping Motion

The jumping sequences of the proposed quadruped robot are illustrated in Fig. 8. The robot first moved to the location where it is intended to launch jumping motion, as shown in Fig. 8 (a). At the desired location, the robot squats by controlling the hip motor to decrease $\theta_1$. Meanwhile, the motor pulls the cable connected to the foot tip to increasing $\theta_2$, which also stretches the knee springs to store energy for jumping, as Fig. 8 (b). Meanwhile, the pawl locked the ratchets on the hips of both rear legs so that the stretching force from the jumping springs does not influence the pre-jumping posture

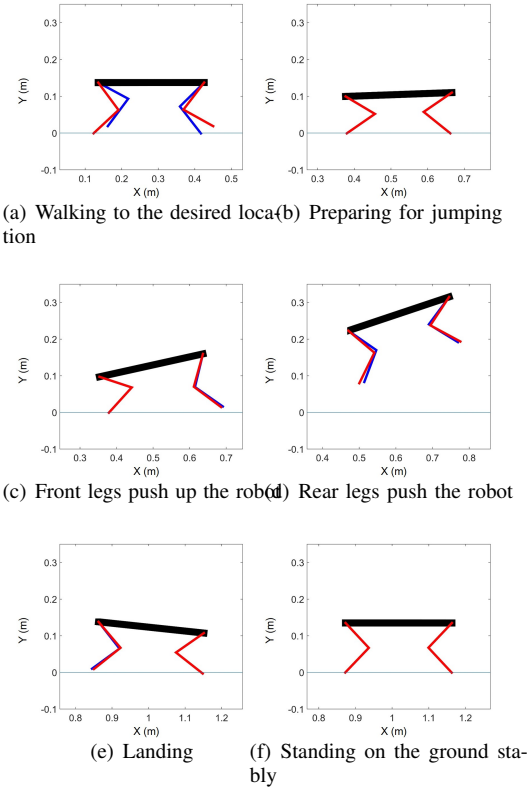
After the jumping springs stretched to the desired distance (required elastic potential energy), the motors in front legs are controlled to push the robot body up by increasing $\theta_1$, and the cable motors rotated in the opposite direction to loosen the cable. The hip motor torques and knee spring forces push the quadruped robot to incline the robot's body with an increase of $\theta$, as illustrated in Fig. 2 (c). When the robot reaches the desired inclined angle, the pawl moves away from the ratchet to release the storing spring energy for the robot to jump off the ground, as seen in Fig. 8 (d). For adjusting jumping height, we changed stretching length of energy storing springs, and the timing of releasing ratchet for various $\theta$. Moreover, different stretching lengths could provide a variety of energy for jumping motion.

While the robot is off the ground, the desired foot positions are controlled by the following equations for stable landing:

$$x_i = x_d + \alpha_x(K_{px}\theta + K_{dx}\dot{\theta}), \tag{15}$$

$$y_i = y_d + \alpha_y(K_{py}\theta + K_{dy}\dot{\theta}), \tag{16}$$

where $x_i$, $y_i$ are the desired foot positions for $i = 1, 2, 3, 4$ representing four legs, the constants $\alpha_x$ and $\alpha_y$ are given as $-1$ for $i = 1, 2$ and 1 for $i = 3, 4$, $y_d$, $x_d$ are the desired stable standing pose after landing, $\theta$ is pitch angle of robot, $K_{px}, K_{dx}, K_{py}, K_{dy}$ are control constants. The purpose of this controller is to reduce angular momentum, which is caused from the velocity of center of mass during landing. Moreover, the angular impulse, coming from the normal ground reaction forces when the front foot gets contact with the ground is also increased. The proposed walking and jumping motions are validated first via numerical examples in the next section.




(a) Walking to the desired location (b) Preparing for jumping

(c) Front legs push up the robot (d) Rear legs push the robot

(e) Landing (f) Standing on the ground stably

Fig. 8: Jumping sequences of the quadruped robot.

## IV. Validation Results

### A. Simulation

The simulation of the proposed quadruped robot is accomplished in Matlab/Simulink based on the dynamic and kinematic models. The reaction force and tangential friction force between the robot feet and the ground are modelled by

$$f_{ni} = -(\lambda\delta_{ni}^m)\dot{\delta}_{ni} - k\delta_{ni}^m \tag{17}$$

$$f_{ti} = \begin{cases} sign(f_{ti})\mu_s|f_{ni}| & \text{if } |f_{ti}| \geq \mu_s|f_{ni}| \\ f_{ti} = -(\lambda\delta_{ti}^m)\dot{\delta}_{ti} - k\delta_{ti}^m & \text{otherwise} \end{cases} \tag{18}$$

where $f_{ni}$ and $f_{ti}$ are the normal forces and friction forces of feet, $\mu_s$ is the coefficient of static friction, $k$ is the model of spring constant of the ground, and $\dot{\delta}_{ni}$ and $\dot{\delta}_{ti}$ are the penetrations and velocity of penetrations in normal and tangent directions, respectively. The details of coefficients, $\lambda$ and power $m$, are described in [27], [28]. If the tangent force of each leg excesses the maximum static friction, the magnitude of the friction is equal to the maximum static friction. The equivalent generalized forces resulted from the ground reaction forces and friction forces are calculated by $J^T F_{ext}$, where $J$ is Jacobian matrix of the robot, and $F_{ext} = [f_{t1}, f_{n1}, f_{t2}, f_{n2}, f_{t3}, f_{n3}, f_{t4}, f_{n4}]^T$ is vector of ground reaction forces and friction forces of four feet. Moreover, the effect of energy storing spring and spring of knee are calculated by the equivalent torques as presented in (10) and (11).

In the numerical examples, parameters in the simulation are $H = 0.03m$, $L = 0.1m$, $T_a = 0.2s$, and $T_d = 0.5s$, $K_{px} = K_{py} = 0.03$, $K_{dx} = K_{dy} = 0.001$, $x_d = 0m$,

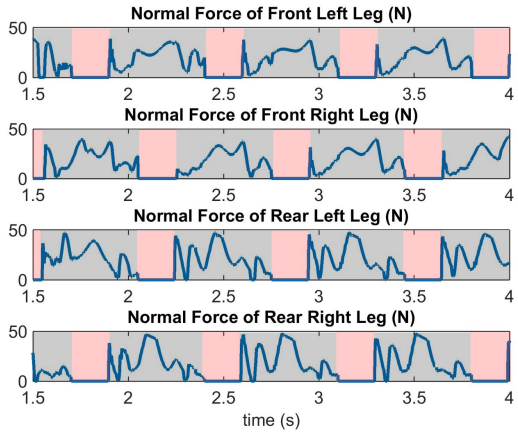



Fig. 9: Walking normal forces in simulations.

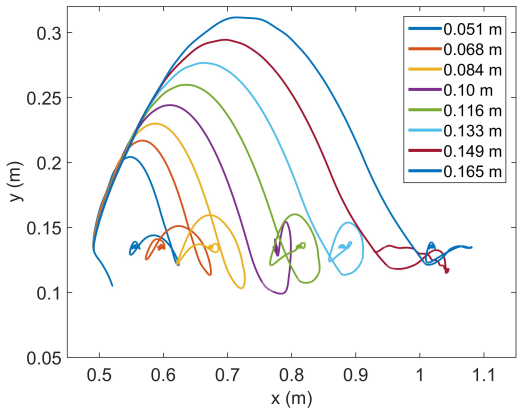



Fig. 10: Jumping trajectories in simulations.

$y_d = -0.135m$, $\lambda = 5.4 \times 10^5$, $k = 9 \times 10^5$, and $\mu_s = 1$. The robot first stands on the ground for 1 second to reach stable stand. Afterward, the quadruped robot starts to walk for 3 seconds and then squatted down preparing for jumping posture, as Fig. 8 (b). After squatting down for 1 second, the front legs push the front part of the robot to achieve a desired inclination. After the robot lift up for 0.1 second, the stored energy in jumping springs is released for rear legs to lift the whole robot up. Fig. 9 shows the normal force of each leg in the time interval between 1.5 to 4 seconds. The normal force equals to zero (pink part) when the leg is in swing phase, and is non-zero when the leg is in touchdown phase (gray part). As shown in the figure, the diagonal pairs of legs touch down at the same time period, and the periods of swing and stance phases are close to $T_a$ and $T_d$. In the jumping simulations, the maximum tension force of the jumping springs varies from 10kgf to 24kgf, which are corresponding to stretching length from 0.051m to 0.165m. The jumping trajectory of $(x_b, y_b)$ in each simulations are plotted in Fig. 10. The results demonstrate that larger deformation of the stretching length for energy-storing spring can make the robot jump higher and longer. It means that jumping height and distance can be adjustable by changing stretching length of energy storing springs.

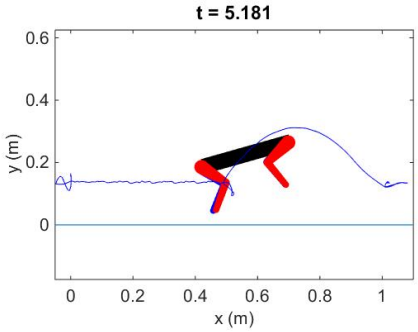



Fig. 11: Snapshot of jumping trajectories in simulations.

### *B. Experiments*

The walking and turning design and motion control are validated via experiments using the desired quadruped robot. In the experiments, the robot makes a $90^o$ turn with different turning speed. The body orientation is sent to a desktop computer via Teensy, and then the computer calculates the set-points for motors from desired trajectory as mentioned in Section III. Subsequently, the set-points are then transferred to the micro controller, Arduino Due, mounted on the robot. The body orientation and motor set-points are exchanged by using Robot Operating System (ROS). Moreover, Arduino Due is responsible for reading angular position of motors and controls motors with PID controller.

Before the experiment, the foot trajectories of the proposed quadruped robot is tested. Markers are put on the tiptoe, knee, and hip of the quadruped robot to measure the trajectories of different gait trajectories. The experimental results are shown in Fig. 12, where the desired height $H$ and length $L$ of each gait are various with $T_a = 0.3$s and $T_d = 0.6$s. It can be observed that the desired trajectories (red-solid lines) are different from the robot's gait (blue dots) for both the front and rear feet. Moreover, it can be observed that the gaits of the rear leg move towards the front leg, and the gaits for front legs move towards the rear leg. This situation results from that the required torques for the cable-driven link is higher with larger deformation, and the torques exceed the maximum motor torque. The height of gait trajectories can be achieved because it is mainly controlled by the DC motor mounted on the hip. Although the trajectories exist larger differences, the gait trajectories in stance phase is closed to the desired trajectories. Therefore, we can still utilize the design and control to test walking performance.

To investigate the walking performance, the walking velocity of the quadruped robot with different gait lengths are conducted. The experimental results for walking velocity with respect to step length are shown in Fig. 13. It can be seen that the actual velocity (blue solid-lines) are lower than the expected velocity (red dashed-lines). The consequence comes from the movement of legs are not faster enough so that the swing-phase is not complete. Another reason for the lower walking velocity results from that the friction coefficient on the walking surface is too lower to provide enough friction force for walking. Therefore, sliding motion is observed during

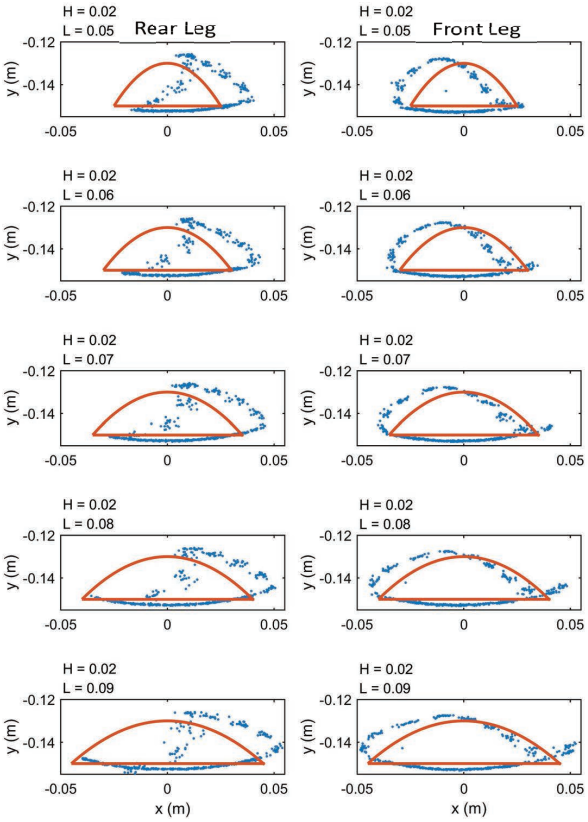



Fig. 12: Gait trajectories of the rear and front legs with various walking height and distance. (Red-solid lines are desired trajectories.)

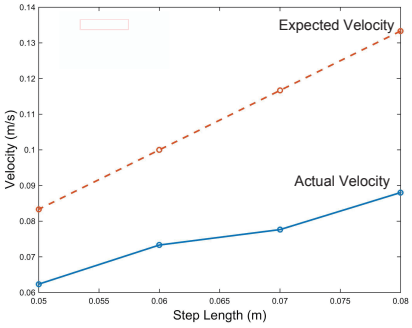
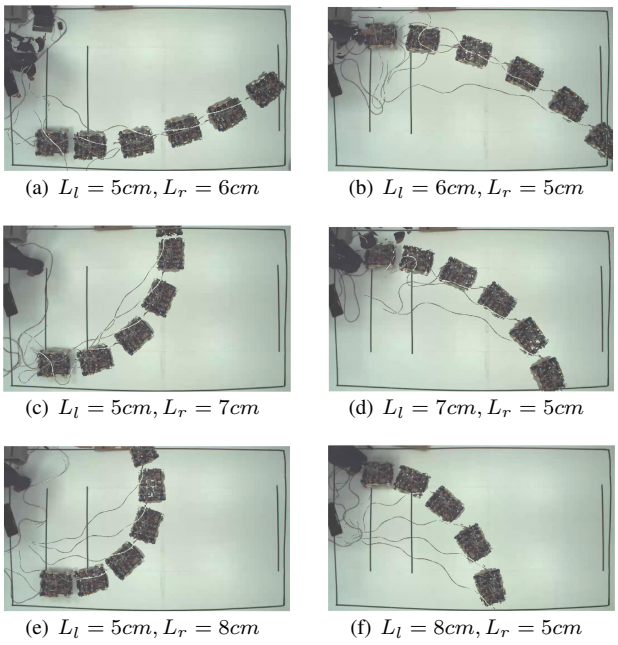



Fig. 13: Walking velocity of the proposed quadruped robot with different step length.

experiments, so that the velocity can not achieve the expected values.

Although the walking velocity is not faster enough comparing to expected value, the robot can walking in a straight line with the same walking speed for right and left legs. Hence, by changing different walking velocity for left and right legs, the quadruped robot is able to make a turn. In the next experiment, we set $H = 0.04$m, $L_d = 0.075$m, $T_a = 0.3$s,



(a) $L_l = 5cm, L_r = 6cm$ (b) $L_l = 6cm, L_r = 5cm$

(c) $L_l = 5cm, L_r = 7cm$ (d) $L_l = 7cm, L_r = 5cm$

(e) $L_l = 5cm, L_r = 8cm$ (f) $L_l = 8cm, L_r = 5cm$

Fig. 14: Snapshots of quadruped robot with various left step distance ($L_f$) and right step distance ($L_r$).

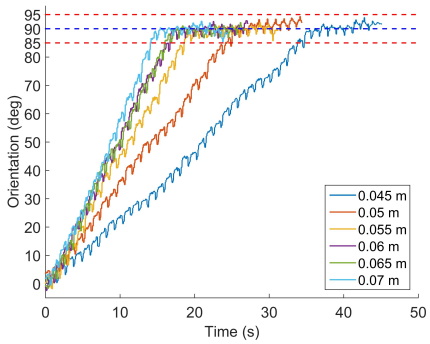
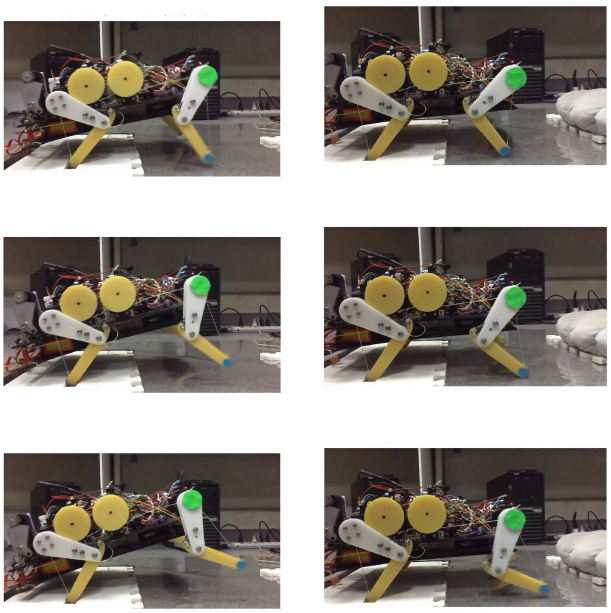



Fig. 15: Orientation of the quadruped robot with the proposed turning method (different $\delta L_{max}$) in the experiments.

$T_d = 0.8$s, $\theta_r = 5°$, and the trajectories difference $\delta L_{max}$ in (13) are changed from 0.045m to 0.07m. The quadruped robot is controlled to make a turn from initial orientation of $0^o$ to $90^o$. Fig. 14 shows the body orientation with different $\delta L_{max}$ in the experiments. It demonstrated that the turning period decreases as $\delta L_{max}$ increases as means that turning can be adjusted by changing the $\delta L_{max}$ value. As seen in Fig 15, the steady-state error in orientation of each experiment is less than $5^o$.

The objective of this research is to make a dog-sized quadruped robot that is capable of waking, making turns, and jumping. Energy-storing mechanisms and cable-driven mechanisms are desired to achieve the aforementioned goals. In the experiments of jumping, the quadruped robot is controlled according to the sequence of jumping motions are shown in Fig. 8. Basically, the front legs with cable-driven legs and DC motors can push the robot to leans the body for a desired inclined angle. The snapshots of jumping sequences for the



(a) Release knee-spring first (b) Release hip-spring first

Fig. 16: Snapshots of jumping sequences for the quadruped robot.

quadruped robot by first releasing knee spring or actuating hip motor are illustrated in Fig. 16. After the quadruped robot leans its body from the front legs, the energy-storing springs and motors on rear legs can be released to push the whole body to achieve jumping motion. However, the friction force and jumping force are limited due to the walking surface and complain mechanisms. Therefore, the quadruped robot can only slightly jumping and not be able to reach the desired motion and jumping height.

### *C. Discussion*

A small-sized quadruped robot is presented to achieve walking, turning, and jumping motion. Compliant legs are designed and utilized with trotting pattern to walk on a plane. Moreover, adjusting walking trajectory can make the quadruped robot to turn right or left. Simulation and experimental results are addressed in this section. From the simulation results, we can observed that walking and jumping motion is achievable for the design of the quadruped robot. However, the robot in experiments can only perform walking and turning motions, and the robot bumps into a certain difficulty in jumping motion.

There are several aspects could be improved to enhance the performance of the motion of the proposed quadruped robot. It can be observed from Fig. 12 that even off the ground, there exists a non-negligible error between the desired and actual foot trajectory. This issue could results from the control of motor mounted on hip and motor for pulling the cable. The compliant legs will suffer to hight spring force with larger deformation from the angle of knee. Therefore, if only controlling the motor position, the driving torque is not higher enough to guarantee the posture of the front and rear legs. This problem could be improved by using motors with higher torque or designing an appropriate control laws to drive the motors.

Another issue for the walking motion results from the desired foot trajectory not being continuous differentiable. Using a straight line and parabolic curve to approximate foot trajectory is easy to implement, but that will cause difficulty in motion control of the driving motor. Foot trajectories for quadruped robots have been studied recently, and these results can be adopted to improved the foot pattern in this paper. Additionally, the weight of the quadruped robot not being in the middle of the robot's body will cause leaning motion while walking on the ground. This phenomenon is observed when the quadruped robot in experiments. To mitigate this problem, increasing walking gaits from static walking to dynamic walking could enhance the walking performance.

Designing a quadruped robot to achieving jumping motion is also an objective of this paper. Although the simulation demonstrates that the design of compliant legs can generate jumping motion, the quadruped robot fails to jump off the ground in experiments. Basically, the problem is in twofold. First, the friction force is not higher enough for the feed to transmit the jumping force to the robot. From the experimental results, if the friction coefficient between the legs and ground is too lower, then slippage occurs between the rear legs and the ground. Therefore, the quadruped robot could fail to finish jumping motion. Second, the energy-storing spring is only mounted for rear legs, but the compliant mechanism and motor for front legs cannot provide enough energy for the robot to achieve desired inclined angle. If the inclined angle is too small, the pushing force from the rear legs will cause a moment that make the robot rotates instead of jumping out. These issues can be taken into account in the design and implementation of quadruped robot to achieve jumping motion.

## V. Conclusion

This paper proposes an integral design for a quadruped robot to achieve walking, turning, and jumping motions. The robot is composed of four compliant legs with cable-driven mechanisms, and energy-storing mechanisms to achieve waking and jumping motions. The design and modeling of the proposed quadruped robot is addressed with the kinematic/dynamic model. Foot trajectories and inverse kinematics are presented to control the leg motion for walking movement. The turning motion can be achieved by regulating the stance phase and walking length accordingly Subsequently, a dynamic model, which is derived by Euler-Lagrange equations, is used in Matlab/Simulink to simulate the jumping motion of the proposed quadruped robot.

From the simulation results, we conclude that the jumping height of the robot can achieve about $0.3$m which is higher than two times of standing height of the robot. The jumping motion of the quadruped robot is also validated via numerical examples. Although jumping height and distance can be adjusted by changing maximum stretching length of energy storing springs, the relation between stretching length

and jumping performance is not clear. The experiments are illustrated to show that the proposed quadruped robot can be controlled to the desired orientation by the turning method. The results show that the robot is able to turn to the desired orientation with error less than $5^o$. Experimental results for jumping motion have also been implements. However, the limits on friction force of the ground, motor torque, and spring force degrade the jumping performance. Future work of this research encompasses the improvement of spring energy for jumping, and friction coefficient for walking motion. Running motion of the proposed robots will also be developed based on the cable-driven and compliant mechanisms.

## REFERENCES

[1] C.-Y. Chan and Y.-C. Liu, "Towards a walking, turning, and jumping quadruped robot with compliant mechanisms," in *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2016.

[2] M. Raibert, "BigDog, the Rough-Terrain Quadruped Robot," in *Proceedings of the 17th IFAC World Congress, 2008*, M. J. Chung, Ed., vol. 17, no. 1.

[3] A. Sprwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, "Towards dynamic trot gait locomotion: Design, control and experiments with cheetah-cub, a compliant quadruped robot," *International Journal of Robotics Research*, vol. 32, no. 8, pp. 932–950, 2013.

[4] H.-W. Park, S. Park, and S. Kim, "Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3d running of mit cheetah 2," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 5163–5170.

[5] M. Zieliński and D. Belter, "Mechanical design and control of compliant leg for a quadruped robot," in *Automation 2018*, 2018.

[6] C. Semini, V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, and D. G. Caldwell, "Design of the hydraulically actuated, torque-controlled quadruped robot hyq2max," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 635–646, 2017.

[7] L. T. Phan, Y. H. Lee, D. Y. Kim, H. Lee, and H. R. Choi, "Stable running with a two-segment compliant leg," *Intelligent Service Robotics*, vol. 10, no. 3, pp. 173–184, 2017.

[8] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1502–1509, 2017.

[9] D. Krasny and D. Orin, "Generating high-speed dynamic running gaits in a quadruped robot using an evolutionary search," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 4, pp. 1685–1696, Aug 2004.

[10] R. Sato, I. Miyamoto, K. Sato, A. Ming, and M. Shimojo, "Development of robot legs inspired by bi-articular muscle-tendon complex of cats," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 1552–1557.

[11] J. Lei, J. Zhu, P. Xie, and M. Tokhi, "Joint variable stiffness of musculoskeletal leg mechanism for quadruped robot," *Advances in Mechanical Engineering*, vol. 9, no. 4, pp. 1–11, 2017.

[12] L. Righetti and A. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, May 2008, pp. 819–824.

[13] E. Koco, D. Mirkovic, and Z. Kovacic, "Hybrid compliance control for locomotion of electrically actuated quadruped robot," *Journal of Intelligent & Robotic Systems*, 2018.

[14] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, "Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 3307–3312.

[15] Y. Song and D. Luchtenburg, "Using compliant leg design for impact attenuation of airdrop landings of quadruped robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[16] D. J. Hyun, S. Seok, J. Lee, and S. Kim, "High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT cheetah," *International Journal of Robotics Research*, vol. 33, no. 11, pp. 1417–1445, 2014.

[17] A. Ananthanarayanan, M. Azadi, and S. Kim, "Towards a bio-inspired leg design for high-speed running," *Bioinspiration and Biomimetics*, vol. 7, no. 4, p. 046005, 2012.

[18] M. Li, Z. Jiang, P. Wang, L. Sun, and S. S. Ge, "Control of a quadruped robot with bionic springy legs in trotting gait," *Journal of Bionic Engineering*, vol. 11, no. 2, pp. 188 – 198, 2014.

[19] S. Bergbreiter, "Effective and efficient locomotion for millimeter-sized microrobots," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept 2008, pp. 4030–4035.

[20] W. Dunwen, G. Wenjie, and L. Yiyang, "The concept of a jumping rescue robot with variable transmission mechanism," in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, Nov 2011, pp. 99–104.

[21] J. Zhao, J. Xu, B. Gao, N. Xi, F. Cintron, M. Mutka, and L. Xiao, "Msu jumper: A single-motor-actuated miniature steerable jumping robot," *Robotics, IEEE Transactions on*, vol. 29, no. 3, pp. 602–614, June 2013.

[22] G.-P. Jung, C. S. Casarez, S.-P. Jung, R. S. Fearing, and K.-J. Cho, "An integrated jumping-crawling robot using height-adjustable jumping module," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[23] K. Graichen, S. Hentzelt, A. Hildebrandt, N. Krcher, N. Gaiert, and E. Knubben, "Control design for a bionic kangaroo," *Control Engineering Practice*, vol. 42, pp. 106 – 117, 2015.

[24] M. Hutter, C. Gehring, M. Hopflinger, M. Blosch, and R. Siegwart, "Toward combining speed, efficiency, versatility, and robustness in an autonomous quadruped," *Robotics, IEEE Transactions on*, vol. 30, no. 6, pp. 1427–1440, Dec 2014.

[25] K. Hosoda, Y. Sakaguchi, and H. Takayama, "Pneumatic-driven jumping robot with anthropomorphic muscular skeleton structure," *Autonomous Robots*, vol. 28, pp. 307–316, April. 2010.

[26] H. Chai, W. Ge, F. Yang, and G. Yang, "Autonomous jumping robot with regulable trajectory and flexible feet," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 228, no. 15, pp. 2820–2832, 2014.

[27] C.-Y. Chan, "Towards a walking, turning, and jumping quadruped robot with compliant and energy storing mechanisms," Master Thesis, National Cheng Kung University, July 2016.

[28] D. Marhefka and D. Orin, "A compliant contact model with nonlinear damping for simulation of robotic systems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 29, no. 6, pp. 566–572, Nov 1999.

**Chen-Yu Chan** received the B.S. degree from National Taiwan University, Taipei, Taiwan in 2013, and M.S. degree from National Cheng Kung University, Tainan, Taiwan in 2016, respectively.

He is currently a software engineer at Compal Electronics. His research interests are in the area of bio-inspired robot and embedded system.

**Yen-Chen Liu** received the B.S. and M.S. degrees in Mechanical Engineering from the National Chiao Tung University, Hsinchu, Taiwan in 2003 and 2005, respectively, and the Ph.D. degree in Mechanical Engineering from University of Maryland, College Park in 2012.

He is currently an Associate Professor in the Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan. He is the recipients of the Ta-You Wu Memorial Award from Ministry of Science and Technology, Taiwan, the Outstanding Robotic Engineering Award from Robotics Society of Taiwan, and Outstanding Young Faculty from Chinese Society of Mechanical Engineering, Taiwan. His research interests include control of robotic system, bilateral teleoperation, multi-robot system, mobile robot network, and human-robot interaction.

# Design and Implementation of DBN-based Gaze Tracker for Auto-aiming System

Chih-Wei Chien, Chih-Yin Liu, Ting-Nan Tsai, Li-Fan Wu, Nien-Chu Fang, and Tzuu-Hseng S. Li*

***Abstract*—This paper proposes a design for an auto-aiming system that is capable of tracking a user's gaze automatically. The auto-aiming system is composed of an electric gun control system and a wearable gaze tracker. According to a user's gaze direction, captured by the helmet camera, the gun can automatically aim at an object which the user is looking at. The mapping model between the user's eye movement and the shooting direction of the gun is established by a Deep Belief Network (DBN). This DBN-based gaze tracking model allows the auto-aiming shooting system to aim at the target in a timely and accurate manner. Finally, the experimental results demonstrate the proposed auto-aiming system for different users can achieve an average accuracy of 96%.**

***Key words*: Gaze tracker, Deep Belief Network, Auto-aiming**

## I. INTRODUCTION

Virtual reality (VR) [1] [2] is a popular technology nowadays in which a player wears a head-mounted display (HMD) [3]. The HMD provides the player experience in a built virtual world so that the player can use his/her body movement to play games. However, VR technology still cannot replace the feeling of playing games in a real environment. To provide a real and intuitive feeling, we propose an auto-aiming system in this paper; this system allows a player to play a shooting game in a real environment by wearing a DBN-based gaze tracker.

The wearable DBN-based gaze tracker is implemented on a helmet and a headset. The helmet contains a web camera and a 9-axis IMU to capture the player's gaze direction in real time. In the meantime, an electrically-controlled gun system is constructed to control a game gun. In this system, a camera is mounted on top of a game gun to capture the environmental image and three servo motors are mounted on the bottom of the game gun to control its movements.

How to connect the wearable gaze tracker and the electrical controlled gun system is the key problem that needs to be solved in this paper. To establish the mapping relationship between gaze directions of the player and the control signal of the electrically-controlled gun system, many machine learning algorithms are considered, such as particle swarm optimization (PSO) [4] [5], artificial bee colony (ABC) [6] [7], genetic algorithm (GA) [8] [9] and neural network (NN) [10]. Since the NN-based algorithm is most suitable for establishing a system identification model, it is adopted for this paper. Furthermore, among NN-based algorithms, the Deep Belief Network (DBN) [11]-[14], which was proposed as an improved neural network algorithm to overcome the limitations of backpropagation, has superior performance in comparison to the basic neural network algorithm. Thus, we chose the DBN to establish the system identification model between gaze directions and control signals.

A DBN has multiple layers between the input and output layer, and each layer of DBN is composed of a Restricted Boltzmann Machine (RBM) [15] [16]. The RBM is a bipartite graph of the energy generation module [17]. It has two layers, one visible layer and one hidden layer. The number of the neurons of the visible layer and the hidden layer are not necessarily the same; only all neurons yet, between the visible and the hidden layer are fully connected. When training weights of a RBM, contrastive divergence (CD) [18] [19] is used to tune the weights. Because the weights are adjusted by CD instead of outputs, the RBM is an unsupervised learning method. After training, outputs of the hidden layer will be inputs of the next layer. Hence, the multiple layers of RBM module arise spontaneously.

Through the training process mentioned above, the RBM can provide the initial weights of DBN. These weights are closer to the best weights; therefore, it can converge fast. Finally, the backpropagation method (BP) [20] [21] is applied to fine-tune all weights.

The paper is constructed as follows: The structure of the auto-aiming system is described in Section II. In Section III, we propose the gaze tracker, which is used to detect the eye region and calculate the gaze direction. The mapping relationship between the gaze direction and the shooting direction is trained by a Deep Belief Network which is described in Section IV, and experiments are presented in Section V. Finally, the paper is concluded in Section VI.

## II. AUTO-AIMING SYSTEM

This paper presents a prototype of an auto-aiming system with high accuracy and low cost. This system aims to shoot targets by detecting the player's gaze directions, and it consists of a gaze tracker system and an electrically-controlled gun system. Fig 1 illustrates the proposed auto-aiming system application scenario. The player wears a specific headset on which a web camera is mounted to detect his/her eye movement. The system then utilizes a DBN-based gaze tracking model to control the shooting direction of the gun.

The electrically-controlled gun system used in this system consists of an electric game gun, three servo motors, and a RGB-D camera, as shown in Fig. 2 (a). We designed a connecting element to mount the electric gun and three

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 105-2221-E-006 -104 -MY2 (e-mail: Q59105021@hotmail.com; scold@mail.ncku.edu.tw; dick10243@gmail.com; keystoradio@gmail.com, f.nienchu@gmail.com, thsli@mail.ncku.edu.tw). Corresponding author: Tzuu-Hseng S. Li.

Fig. 1. Illustration of the proposed auto-aiming system

(a) (b)

Fig. 2. The electrically-controlled gun and the wearable gaze tracker. (a) The electrical controlled gun. (b) The wearable gaze tracker.

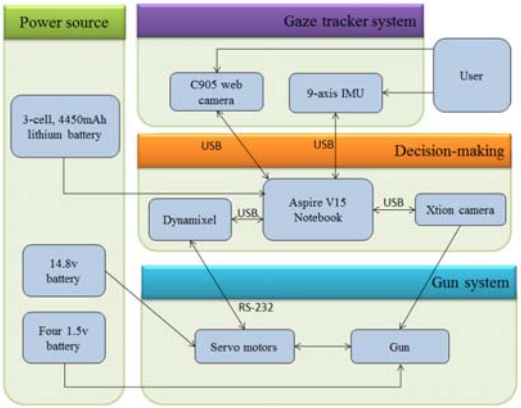



Fig. 3. The overall architecture of the auto-aiming system.

motors on a camera tripod, so that two of the motors can control the pan and tilt rotation of the gun, and the other one can pull the trigger of the gun. In addition, we adopt an Asus Xtion Pro Live Camera to capture images for target-detecting.

We implemented the wearable gaze tracker by a headset, as shown in Fig. 2 (b). A web camera is mounted in front of the headset to detect the eye region and the gaze movement of the player, and an IMU is added on the top of the headset to detect the rotation of the player's head.

The eye movement is extracted from the image captured by the web camera and used as the input of the learning system, which constructs a mapping relationship between eye movement and shooting direction. The calculated direction is then integrated with the head rotation to determine the final pan and tilt angle of the motors. Fig. 3 shows the overall architecture of the auto-aiming system.

## III. EYE TRACKER SYSTEM

In the auto-aiming system, the gaze direction of the player controls the movement of the electrically-controlled gun system to shoot targets. Therefore, the eye region and the gaze direction have to be detected first. We apply Haar-like features [22] to extract eye region. First, the image is translated to an integral image for speeding up the calculation of features. For an input image $\boldsymbol{I}$, the integral value of the pixel $(x, y)$ is defined by the following equation

$$ii(x, y) = \sum_{t \le x, k \le y} i(t, k) \tag{1}$$

where $i(t,k)$ is the gray value of the pixel $(t,k)$, and the value of the integral image equals the summation of gray values of all shadow partition. By this definition, the integral image is derived by scanning all pixels in the original image.

After that, we use AdaBoost [22] to construct a cascade of classifiers. AdaBoost is a kind of iterative algorithm and it can choose some meaningful classifiers from several weak classifiers to institute a strong classifier. This algorithm uses rectangles as features for the basis of classification. The value of a rectangular feature is the difference of total gray value between two rectangles. In the weak classifier, a rectangular feature $j$ corresponds to a weak classifier $h_j$. For a candidate region $x$, the feature value of this region is $f_j(x)$, and the classify function of the weak classifier is shown in the following equation

$$h_j(x) = \begin{cases} 1, \text{ if } p_j f_j(x) < p_j \theta_j \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

where the $p_j$ is parity, and $\theta_j$ is the threshold. Training with the data derived from OpenCV [23], we found two features of the eye region, as shown is Fig 4.

Next, we connect several weak classifiers to construct a cascade classifier. A datum has to pass all classifiers layer by layer before being treated as an eye region. This method not only increases the accuracy of classification but also decreases the computational complexity.

After the eye region is derived, we extract the black region of the eye. The eye image is transformed into a gray scale image, and a threshold is set to extract the most likely pixels of the black region and further transform the grey image into a binary image by the following equation

$$pixel(x, y) = \begin{cases} R = G = B = 255, \text{ if } g(x, y) \le threshold \\ R = G = B = 0, \text{ otherwise} \end{cases} \tag{3}$$

Fig.4. The sample features of AdaBoost.

where $g(x,y)$ is the gray value of the pixel $p(x,y)$. The threshold is set by an entropy method proposed in [24] in which the probability of gray level $f_i$ is defined by the following equation

$$p_i = \frac{f_i}{N} \tag{4}$$

where $N$ is the total number of pixels in the image and $f_i$ is the number of pixels with this gray level. Therefore, the first order entropy of the image can be defined as follows:

$$H = -\sum_{i=0}^{L-1} p_i \log p_i s \tag{5}$$

where $L$ is the maximum gray value of the image. The entropies of the eye region and background are defined using the following equations

$$H_E(s) = -\sum_{i=0}^{s} \frac{p_i}{p_s} \log \frac{p_i}{p_s} \tag{6}$$

$$H_B(s) = -\sum_{i=s+1}^{L-1} \frac{p_i}{1-p_s} \log \frac{p_i}{1-p_s} \tag{7}$$

where $H_E$ and $H_B$ are the entropies of eye region and background, respectively. $s$ is a threshold, and $p_s = \sum_{i=0}^{s} p_i$ . The total entropy is defined by the following equation

$$H_T(s) = H_E(s) + H_B(s) \tag{8}$$

Therefore, the biggest value of $H_T(s)$ is defined as the threshold of image as given by the following equation

$$H_T(s^*) = \max H_T(s) \tag{9}$$

By this algorithm, we can find the best threshold to transform the gray image into the binary image. The original RGB image and the binary image are shown in Fig.5 (a).

However, there is some noise existing in the binary image because of shadow and the eyelash. To wipe out this noise, two image process algorithms, erosion and dilation [25], are used. Erosion and dilation are defined by the following equations

$$A \Theta B = \{x \in E^N \mid x = x + b \in \mathrm{A} \text{ for every } \mathrm{b} \in \mathrm{B}\} \tag{10}$$

$$A \oplus B = \{c \in E^N \mid c = a + b, \text{ for some } \mathrm{a} \in \mathrm{A} \text{ and } \mathrm{b} \in \mathrm{B}\} \tag{11}$$

where $A$ and $B$ are sets in N-space ( $E^N$ ) with elements $a$ and $b$, in which $a = \{a_1, a_2, a_3, ..., a_N\}$ and $b = \{b_1, b_2, b_3, ..., b_N\}$ . In this paper, $A$ is the input image and $B$ is the template. After this, the final eye region is derived as shown in Fig. 5(c); the white region is the region of the iris and the red point is the center of the iris.

The gaze direction is calculated by the difference of iris centers between the initial point and the detected point, as in the following equation

$$V_x = E_{now\_x} - E_{initial\_x} \tag{12}$$

$$V_y = E_{now\_y} - E_{initial\_y} \ . \tag{13}$$

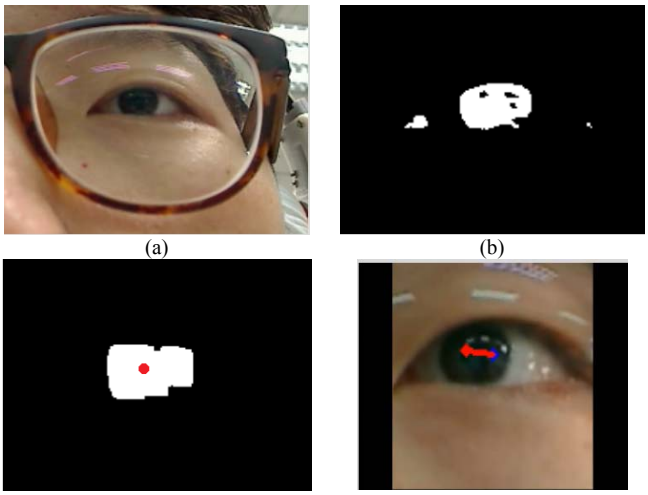


Fig. 5. (a) The original image of the eye. (b)The binary image of the black eye region. (c). The black eye region after erosion and dilation. (d). The result of the gaze direction vector in which the blue point indicates the initial point of the iris center, and the red point indicates the present point of the iris center. The red line is the gaze direction vector.

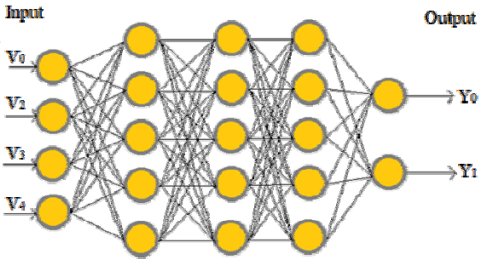



Fig. 6. The architecture of DBN.

The initial point of the iris center $(E_{initial\_x}, E_{initial\_y})$ is captured in a calibration process. In the process, the player is asked to look ahead in the distance in a relaxed manner, so the center of the iris can be viewed as the center of the eye.

Therefore, we can obtain the gaze direction vector, as shown by the red line in Fig. 5(d). The obtained gaze direction vectors are then used as the input of the learning system.

## IV. Deep Belief Network Learning

For construction of the mapping relationship between gaze direction and the electrically-controlled gun system, we use a deep belief network.

### *A. Deep Belief Network (DBN)*

The DBN is a multi-layer network as shown in Fig. 6. In the DBN, every set of two adjacent layers comprise a RBM module. The RBM is a kind of stochastic neural network, and includes a visible layer and a hidden layer. The visible layer describes the feature of observed data, and the hidden layer is the related rate between visible units and variables in the visible layer. Hence, the hidden layer is also called the feature distribution layer.

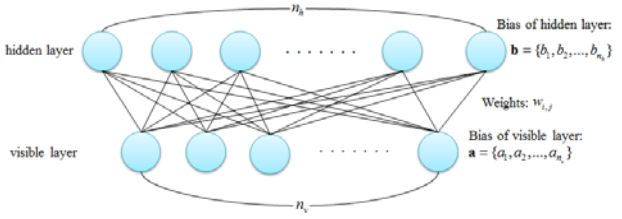



Fig. 7. The architecture of the RBM.

TABLE 1. THE PARAMETERS OF DBN.

| | |
|---|---|
| $n_v, n_h$ | The number of neurons in visible layer and hidden layer, respectively. |
| $\mathbf{v} = (v_1, v_2, \ldots, v_{n_v})^T$ | State vector of visible layer, and $v_i$ describes state of *i*-th neuron. |
| $\mathbf{h} = (h_1, h_2, \ldots, h_{n_h})^T$ | State vector of hidden layer, and the $h_i$ describes the state of *i*-th neuron. |
| $\mathbf{a} = (a_1, a_2, \ldots, a_{n_v})^T \in R^{n_v}$ | Bias vector of visible layer, and the $a_i$ describes the state of *i*-th neuron. |
| $\mathbf{b} = (b_1, b_2, \ldots, b_{n_h})^T \in R^{n_h}$ | Bias vector of hidden layer, and the $b_i$ describes the state of *i*-th neuron. |
| $\mathbf{W} = w_{i,j} \in R^{n_h \times n_v}$ | The weights between the visible layer and the hidden layer, and the $w_{i,j}$ describes the weight of *i*-th neuron of hidden layer, which connects with *j*-th neuron of visible layer. |
| $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{a}, \mathbf{b}\}$ | The set of above parameters $\{\mathbf{W}, \mathbf{a}, \mathbf{b}\}$. |

The neurons of the RBM have two characteristics. One is full connection between the visible and the hidden layer, and the other is the absence of connection among neurons within the same layer. Full connection means that each neuron of the visible layer connects with all neurons of the hidden layer, and, by the same token, each neuron of the hidden layer also connects with all neurons of the visible layer. Therefore, the active condition of each neuron in the hidden layer is independent of the others. Fig. 7 shows the architecture of the RBM, and Table 1 tabulates the parameters of the DBN.

We adopt the binary RBM in which the states of neurons are binary values. Because the RBM is an energy based module, the energy function is defined first. By giving the state of (*v*, *h*), the definition of the energy function [12] is produced by the following equation

$$\mathrm{E}(v,h) = -\sum_{j=1}^{n_v} a_j v_j - \sum_{i=1}^{n_h} b_i h_i - \sum_{j=1}^{n_v}\sum_{i=1}^{n_h} h_i w_{i,j} v_j \tag{14}$$

And the joint probability distribution of the state (*v*, *h*) is defined by the following equations

$$\mathrm{P}(v,h) = \frac{1}{Z} e^{-E(v,h)} \tag{15}$$

$$Z = \sum_{v,h} e^{-E(v,h)} \tag{16}$$

where Z is the partition function. Therefore, the probability distribution of the state of a visible neuron can be defined by the following equation

$$\mathrm{P}(v) = \sum_h \mathrm{P}(v,h) = \frac{1}{Z}\sum_h e^{-E(v,h)} \tag{17}$$

Similarly, the probability distribution of the state of a neuron in the hidden layer is defined by the following equation

$$\mathrm{P}(h) = \sum_v \mathrm{P}(v,h) = \frac{1}{Z}\sum_v e^{-E(v,h)} \tag{18}$$

When the state of a neuron in the visible layer is given, the active probability of a neuron in the hidden layer is defined as $P(h_k = 1 \mid v)$. Similarly, the active probability of a neuron in the visible layer is defined as $P(v_k = 1 \mid h)$. Then, the vector $\mathbf{h}$, which removes $h_k$, is defined by the following equation

$$\mathbf{h}_{-k} = (h_1, h_2, \ldots, h_{k-1}, h_{k+1}, \ldots h_{n_h})^T \tag{19}$$

When we define the following two equations,

$$\alpha_k(v) = b_k + \sum_{i=1}^{n_v} w_{k,i} v_i \tag{20}$$

$$\beta(v, h_{-k}) = \sum_{i=1}^{n_v} w_{k,i} v_i + \sum_{\substack{j=1 \\ j\neq k}}^{n_h} b_j v_i + \sum_{i=1}^{n_v}\sum_{\substack{j=1 \\ j\neq k}}^{n_h} h_j w_{j,i} v_j \tag{21}$$

the energy function (14) can be rewritten as

$$\mathrm{E}(\mathrm{v},\mathrm{h}) = -\beta(\mathrm{v}, h_{-k}) - h_k \alpha_k(v) \tag{22}$$

The active probability of a neuron in the hidden layer, $P(h_k = 1 \mid v)$, can be derived by the method in [10], and finally we obtain the following equation

$$P(h_k = 1 \mid v) = sigmoid(b_k + \sum_{i=1}^{n_v} w_{k,i} v_i) \tag{23}$$

Similarly, the active probability of a neuron in the visible layer, $P(v_k = 1 \mid h)$, is derived by the following equation

$$P(v_k = 1 \mid h) = sigmoid(a_k + \sum_{j=1}^{n_h} w_{j,k} h_j) \tag{24}$$

After defining the probability distribution of a RBM, we can start to train this RBM. The datum and the whole training set are defined by the following equations

$$\mathrm{S} = \{\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^{n_s}\} \tag{25}$$

$$\mathbf{v}^i = (v_1^i, v_2^i, v_3^i, \ldots, v_{n_v}^i) \tag{26}$$

where $n_v$ is the dimension of a datum, and $n_s$ is the number of the data set. The RBM adjusts the parameters **W**, **a**, and **b** to match its probability distribution as close to the training data as possible. Therefore, the maximum of the likelihood function of the RBM can be defined by the following equation

$$\zeta_{\theta,S} = \prod_{i=1}^{n_s} P(v^i) \tag{27}$$

Since the log production is much simpler in computation, we change the maximum of the likelihood function using the following equation

$$\ln \zeta_{\theta,S} = \ln \prod_{i=1}^{n_s} P(v^i) = \sum_{i=1}^{n_s} \ln \mathrm{P}(v^i) \tag{28}$$

We use a gradient ascent method [26] to calculate the maximum value iteratively. The iterative function is defined by the following equation

$$\theta(t+1)=\theta(t)+\eta\frac{\partial \ln \zeta_{\theta,S}}{\partial \theta} \tag{29}$$

where $\eta$ is the learning rate and the term of $\frac{\partial \ln \zeta_{\theta,S}}{\partial \theta}$ presents the partial derivation of all parameters. It can be represented by the following equation after some derivations

$$\frac{\partial \ln \zeta_{\theta,S}}{\partial \theta}=\sum_{m=1}^{n_s}\frac{\partial \ln P(v^m)}{\partial \theta} \tag{30}$$

Furthermore, the partial derivatives of all parameters can be derived by the following equations

$$\frac{\partial \ln \zeta_{\theta,S}}{\partial w_{i,j}}=\sum_{m=1}^{n_s}[P(h_i=1\mid v_i{}^m)v_j{}^m-\sum_v P(v)P(h_i=1\mid v)v_j] \tag{31}$$

$$\frac{\partial \ln \zeta_{\theta,S}}{\partial a_i}=\sum_{m=1}^{n_s}[v_i{}^m-\sum_v P(v)v_i] \tag{32}$$

$$\frac{\partial \ln \zeta_{\theta,S}}{\partial b_i}=\sum_{m=1}^{n_s}[P(h_i=1\mid v_i{}^m)-\sum_v P(v)P(h_i=1\mid v)] \tag{33}$$

Next, we use the contrastive divergence (CD) method to estimate the sigma $P(v)$. Compared with the Markov Chain Monte Carlo (MCMC) which is usually used to solve this problem, the CD method need less time for state transition, and results in faster computation. CD with k steps is called CD-k which gives an initial value $v^{(0)}:=v\,,\forall v\in S$ first ,and executes k steps Gibbs sampling [27]. Each step is performed step by step. Giving the random number $r\in[0,1)$ , when sampling **h** given **v**, it uses $P(h\mid v^{(t-1)})$ to sample the $h^{(t-1)}$, as in the following equations

$$\mathbf{h}=P(h\mid v^{(t-1)})=sigmoid(\mathbf{wv}+\mathbf{b}) \tag{34}$$

$$h_i^s=\begin{cases}1, \text{ if } h_i>r\\ 0, \textit{ otherwise}\end{cases},\ \text{for } i=1,2,...,n_h \tag{35}$$

When sampling **v** given **h**, it uses $P(v\mid h^{(t-1)})$ to sample the $v^{(t-1)}$, as in the following equations

$$\mathbf{v}=P(v\mid h^{(t-1)})=sigmoid(\mathbf{w}^T\mathbf{h}+\mathbf{a}) \tag{36}$$

$$v_i^s=\begin{cases}1, \text{ if } v_i>r\\ 0, \textit{ otherwise}\end{cases},\ \text{for } i=1,2,...,n_v \tag{37}$$

The training steps of CD-k are illustrated in Fig. 8.

After Gibbs sampling and derivation of the $\mathbf{v}^{(k)}$ , the equations (31), (32), and (33) can be approximated as the following equations

$$\frac{\partial \ln P(\mathbf{v})}{\partial w_{i,j}}\approx P(h_i=1\mid \mathbf{v}^{(0)})v_j{}^{(0)}-P(h_i=1\mid \mathbf{v}^{(k)})v_j{}^{(k)} \tag{38}$$

$$\frac{\partial \ln P(\mathbf{v})}{\partial a_i}\approx v_i^{(0)}-v_i^{(k)} \tag{39}$$

$$\frac{\partial \ln P(\mathbf{v})}{\partial b_i}\approx P(h_i=1\mid \mathbf{v}^{(0)})-P(h_i=1\mid \mathbf{v}^{(k)}) \tag{40}$$

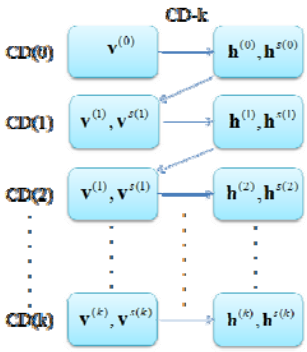


Fig. 8. The training steps of CD-k.

Hence, the energy gradient function of $\zeta_{\theta,S}$ can be derived, and the function of updating parameters can be defined as

$$\mathbf{W}(t+1)=\mathbf{W}(t)+\eta\frac{\partial \zeta_{\theta,S}}{\partial w_{i,j}}=\mathbf{W}(t)+\eta(\mathbf{h}^{s(0)}(\mathbf{v}^{(0)})^T-\mathbf{h}^{(k)}(\mathbf{v}^{s(k)})^T)$$

(41)

$$\mathbf{a}(t+1)=\mathbf{a}(t)+\eta\frac{\partial \zeta_{\theta,S}}{\partial a_i}=\mathbf{a}(t)+\eta(\mathbf{v}^{(0)}-\mathbf{v}^{s(k)}) \tag{42}$$

$$\mathbf{b}(t+1)=\mathbf{b}(t)+\eta\frac{\partial \zeta_{\theta,S}}{\partial b_j}=\mathbf{b}(t)+\eta(\mathbf{h}^{s(0)}-\mathbf{h}^{(k)}) \tag{43}$$

Using the equations (14)-(43), the parameters of RBM can be derived. The DBN is constructed with several layers of RBMs, the hidden layer in the first RBM acting as the visible layer of the second RBM, and so on. When all layers of the RBM are trained, the DBN will fine-tune all parameters by the backpropagation method. The output error of the DBN is defined by the following equation

$$e_j=d_j-y_j \tag{44}$$

where $e_j$ is the output error, $j$ denotes the *j*-th dimension of output, $d_j$ is desired output value, and $y_j$ is the output value of the DBN. Then, the error energy $E$ is defined as

$$E=\frac{1}{2}\sum_{j\in N}e_j{}^2 \tag{45}$$

Minimizing the error energy iteratively can adjust the weights and biases in the DBN, so that the gradient of weights and biases can be updated by the following equations

$$\mathbf{W}(t+1)=\mathbf{W}(t)+\Delta w_{i,j} \tag{46}$$

$$\mathbf{b}(t+1)=\mathbf{b}(t)+\Delta b_{i,j} \tag{47}$$

where $\mathbf{W}$ and $\mathbf{b}$ present the weights and the bias of the DBN, respectively. $\Delta w_{i,j}$ and $\Delta b_{i,j}$ are the gradient of the weight and the bias between the *j*-th layer to the *i*-th layer, respectively. In this way, the DBN can be fine-tuned to approach great accuracy.

TABLE 2. THE RESULT OF AVERAGE MISCLASSIFICATION RATE IN DIFFERENT CLASSIFIERS.

| | Balance | Cancer | Cancer-int | Dermatology | Glass | Iris | Wine | Summarized rank |
|---|---|---|---|---|---|---|---|---|
| BayesNet[29] | 19.74(9) | 4.19(8) | 3.42(6) | 1.08(1) | 29.62(5) | 2.63(9) | 0(1) | 39 |
| MLP ANN[29] | 9.29(3) | 2.93(5) | 5.25(10) | 3.26(3) | 28.51(4) | 0(1) | 1.33(4) | 30 |
| RBF[29] | 33.61(13) | 20.27(14) | 8.17(14) | 43.29(13) | 44.44(13) | 9.99(14) | 2.88(9) | 90 |
| Kstar[29] | 10.25(4) | 2.44(3) | 4.57(8) | 4.66(8) | 17.58(1) | 0.52(6) | 3.99(11) | 41 |
| Bagging[29] | 14.77(7) | 4.47(9) | 3.93(7) | 3.47(5) | 25.36(3) | 0.26(5) | 2.66(8) | 44 |
| MultiBoost[29] | 24.2(12) | 5.59(10) | 5.14(9) | 53.26(14) | 53.7(14) | 2.63(9) | 17.77(14) | 82 |
| NBTree[29] | 19.74(9) | 7.69(13) | 5.71(12) | 1.08(1) | 24.07(2) | 2.63(9) | 2.22(6) | 52 |
| Ridor[29] | 20.63(11) | 6.36(11) | 5.48(11) | 7.92(12) | 31.66(7) | 0.52(6) | 5.1(12) | 70 |
| VFI[29] | 38.85(14) | 7.34(12) | 5.71(12) | 7.6(11) | 41.11(11) | 0(1) | 5.77(13) | 74 |
| PSO[29] | 13.12(6) | 3.49(7) | 2.64(5) | 6.08(10) | 38.67(9) | 5.26(13) | 2.88(9) | 59 |
| ABC[29] | 15.38(8) | 2.81(4) | 0(1) | 5.43(9) | 41.5(12) | 0(1) | 0(1) | 36 |
| GSA[29] | 12.92(5) | 1.39(1) | 0.91(2) | 4.12(7) | 31.89(8) | 2.89(12) | 1.58(5) | 40 |
| DBN | 2.24(1) | 2.23(2) | 1.16(3) | 3.26(3) | 30.63(6) | 0(1) | 1.11(3) | 19 |
| NN | 4.33(2) | 3.18(6) | 2.14(4) | 3.86(6) | 39.38(10) | 0.57(8) | 2.22(6) | 42 |

To demonstrate the performance of the DBN, we compared it with several classifiers in seven data sets chosen from the UCI machine learning repository [28]. These seven datasets include Balance, Cancer, Cancer-Int, Dermatology, Glass, Iris, and Wine.

We used a four-fold cross-validation method to examine the performance of the conventional neural network (NN) and the DBN, comparing them with the performance of the other 12 classifiers that were proposed in [29]. The learning rate of each RBM in DBN is set as 0.1, and the iteration of each RBM is 100. The average misclassifications and the ranking of all classifiers are tabulated in Table 2.

Compared with these classifiers, the DBN has the best performance in the Balance and the Iris data set. For overall performance, the DBN has the best performance and with it the highest summarized rank among these classifiers, which is shown in the right-most column in Table 2. Generally speaking, the DBN has great accuracy in classification problems; therefore, we chose it as the learning method in this paper for learning the mapping relationship between the gaze direction and the electrically-controlled gun system.

### *B. DBN for gaze tracker*

After constructing a DBN and evaluating its performance, we next implemented it to build the relationship between the gaze direction, which was extracted by the method mentioned in Section III, and the electrically-controlled gun system. In the training of DBN, the first step is to train RBM layer by layer. The training input is the gaze direction $(V_x, V_y)$ calculated in (12) and (13), and it is normalized to a value between 0 and 1.

In DBN, the expectation output needs to be defined first in order to fine-tune the parameters. The expectation output is a set of the pan angle and tilt angle of motors, which is defined as $(M_x, M_y)$. After the training, the gaze direction can map to the rotation of motors.

In the real application, not only the gaze direction needs to be considered; the head rotation of the player also affects the gaze point estimation. Therefore, we added an IMU on the headset, so that the pitch, roll, and yaw angles of the head can be estimated. The angles of yaw and pitch are used to adjust the final angle of the output, shown in the following equations

$$M_{x\_final} = M_x + \theta_{yaw} \tag{48}$$

$$M_{y\_final} = M_y + \theta_{pitch} \tag{49}$$

Finally, the final rotation angle of motors, $(M_{x\_final}, M_{y\_final})$, is obtained. When the player is looking at a target, the electrically-controlled gun system will aim at the target in a timely and accurate manner to allow shooting.

## V. EXPERIMENTS

There are many factors of the DBN that affect its performance, such as the number of layers, the learning rates, and so on. Therefore, we first examine the DBN with

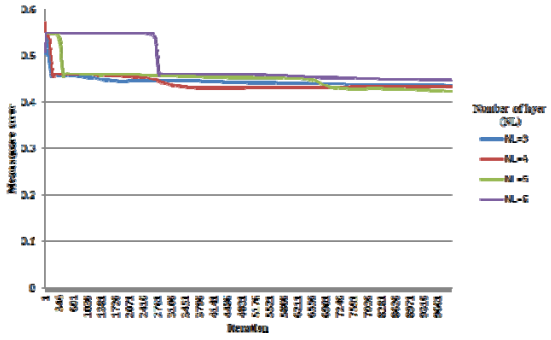


Fig.9. The comparison of different numbers of layers.

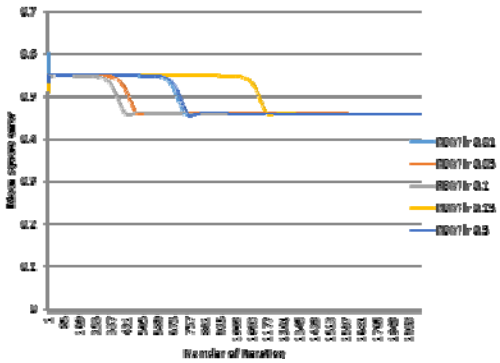


Fig.10. The results of different learning rates of RBM.

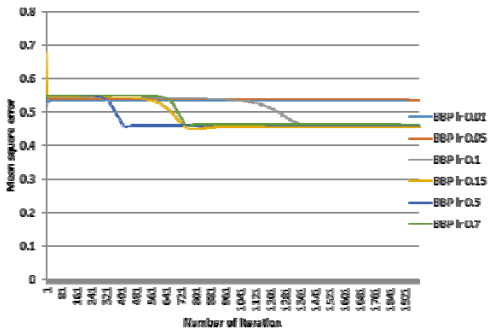



Fig. 11. The results of different learning rates of backpropagation

different numbers of layers. Generally, the higher the number of the layers, the more accurate the DBN is. However, the more layers there are, the more time is needed for converging. To strike a balance between the accuracy and the convergence speed, we tested several numbers of layers. Fig. 9 shows the results. When the layer number is five, the system has the best performance, so we chose a 5-layer structure DBN for all the following experiments.

The learning rate is another factor which affects the performance of the DBN. We examine the learning rate in the RBM on each layer and the fine-tuning in DBN, respectively. Fig. 10 shows the mean square errors with the learning rates in the RBM with 0.01, 0.05, 0.1, 0.15, and 0.5. After 2000 iterations, the mean square errors of all learning rates are similar. However, when the learning rate is equal to 0.1, the convergence speed is the fastest. Therefore, we set the learning rate of the RBM to 0.1.

Because we use the backpropagation method for the fine-turning in the DBN, the learning rate in backpropagation also affects the performance of the learning system. The mean square errors of different learning rates are shown in Fig. 11. When the learning rate equals 0.01 and 0.05, the system cannot converge. Because the mean square error in the others' conditions are similar, we chose the learning rate that has the best convergence speed.

Next, we set an experiment field to demonstrate the efficiency of our auto-aiming system in our laboratory. The area is about $440 \times 360$ ($cm^2$). In the training phase, the player is asked to look at 20 training points, which were located on the ground in the test area at an average pitch of 10 cm, and are represented by the yellow circles in Fig. 12. The data were recorded as the training input of the DBN. In the testing phase, five target boards are randomly located in the area, and the player has to look at the target boards to shoot them down, as shown by the yellow boards in Fig. 13. The size of the target board is $20 \times 20$ ($cm^2$).

There were five human subjects invited to participate in the experiment, including four male and one female; one of them wears glasses. The training time and the shooting accuracy are shown in Table 3. Generally speaking, the DBN only took a little time for training and the performance was great. Only one shot of the fourth participant missed; all other shots correctly shot down the targets. This results are satisfying and the system is trustworthy.

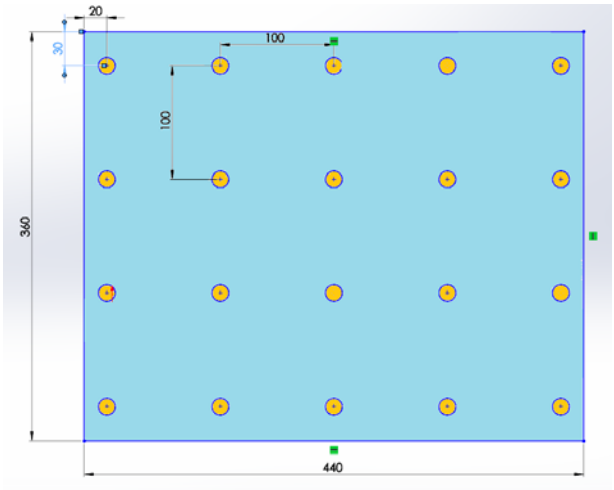



Fig. 12. The simulated picture of the experimental ground.

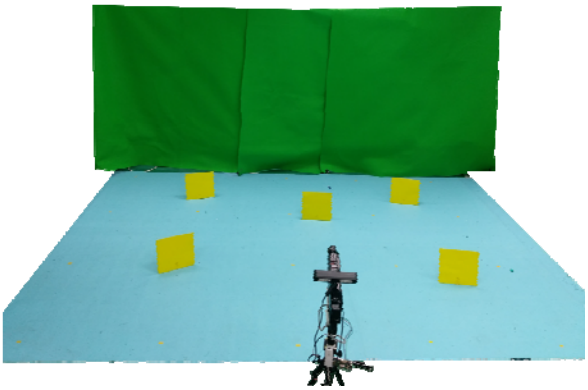


Fig. 13. The real experimental environment.

TABLE 3. THE DETAILS AND RESULTS OF USERS.

| | Gender | glasses | Training time(s) | Shooting Accuracy |
|---|---|---|---|---|
| User 1 | Male | Yes | 45.5 | 100% |
| User 2 | Male | No | 50.6 | 100% |
| User 3 | Female | No | 40.2 | 100% |
| User 4 | Male | No | 41.8 | 80% |
| User 5 | Male | No | 46.2 | 100% |
| Average | | | 44.86 | 96% |

## VI. CONCLUSION

This paper has proposed an auto-aiming system which is composed of electrically-controlled gun system and a wearable gaze tracker. The gaze tracker detected the region of the player and calculated the iris position to estimate its displacement. The displacement is defined as the gaze direction vector and used as the input of the Deep Belief Network (DBN) to train the mapping relationship between the gaze direction and the electrically-controlled gun system. The DBN is composed of multiple layers of the RBM, which includes a visible layer and a hidden layer, and uses a backpropagation method to fine-tune. The simulations in seven datasets showed the DBN has a competitive performance with other famous classifiers. The experiments

also demonstrated that the proposed aiming system is suitable for many participants, and enables them to shoot the targets correctly.

## REFERENCES

[1] J. Steuer, "Defining Virtual Reality: Dimensions Determining Telepresence," *Journal of Communication*, vol. 42, no. 4, pp. 73-93, 1992.
[2] F. Pallavicini, et al., "What Distinguishes a Traditional Gaming Experience from One in Virtual Reality? An Exploratory Study," in *Proceedings of International Conference on Applied Human Factors and Ergonomics*, pp. 225-231, 2017.
[3] C. Cruz-Neira, D. Sandin, and T. DeFanti, "Surround-screen Projection-based Virtual Reality," in *Proceedings of the annual conference on Computer graphics and interactive techniques*, 1993.
[4] S. Cheng, et al., "A quarter century of particle swarm optimization," *Complex and Intelligent Systems*, https://doi.org/10.1007/s40747-018-0071-2
[5] J. Kennedy, "Particle Swarm Optimization," *Encyclopedia of Machine Learning*, pp.760-766, 2011.
[6] S. Aslan, D. Karaboğa, and A. Aksoy, "Performance analysis of artificial bee colony algorithm on ARM based mobile platform," in *Proceedings of IEEE International Conference on Computer Science and Engineering,* pp. 155-159, 2017.
[7] D. Karaboga and B. Basturk, "On the Performance of Artificial Bee Colony (ABC) Algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687-697, 2008.
[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II," *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.
[9] S. K. Pal and P. P. Wang, *Genetic algorithms for pattern recognition*. CRC press, 2017.
[10] Z. Zhang, "Artificial neural network," in *Multivariate Time Series Analysis in Climate and Environmental Research*, pp. 1-35, Springer, Cham, 2018.
[11] Z. Zhao, et al., "Discriminant deep belief network for high-resolution SAR image classification," *Pattern Recognition*, vol. 61, pp. 686-701, 2017.
[12] G. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504-507, 2006.
[13] L. Deng and D. Yu, "Deep Learning: Methods and Applications," *FNT in Signal Processing*, vol. 7, no. 3-4, pp. 197-387, 2014.
[14] M. Z. Uddin, et al., "Facial expression recognition utilizing local direction-based robust features and deep belief network," *IEEE Access*, vol. 5, pp. 4525-4536, 2017.
[15] L. W. Kim, "DeepX: Deep learning accelerator for restricted boltzmann machine artificial Neural Networks," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1441-1453, 2018.
[16] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," *Neural Networks: Tricks of the Trade*, pp. 599-619, 2012.
[17] E. Scheinerman, *Mathematics: A Discrete Introduction*, Boston, pp. 363, 2012.
[18] P. N. Krivitsky, "Using contrastive divergence to seed Monte Carlo MLE for exponential-family random graph models," *Computational Statistics and Data Analysis*, vol. 107, pp. 149-161, 2017.
[19] M. A. Carreira-Perpignan and G. E. Hinton, "On Contrastive Divergence Learning," in *Proceedings of International Conference on Artificial Intelligence and Statistics*, pp. 33-41, 2005.
[20] L. Jiang, et al., "Toward blind nonlinearity estimation in back-propagation algorithm for coherent optical transmission systems," in *Proceedings of IEEE Optical Fiber Communications Conference and Exhibition,* pp. 1-3, 2017.
[21] J. C. Whittington and R. Bogacz, "An approximation of the error backpropagation algorithm in a predictive coding network with local Hebbian synaptic plasticity," *Neural computation*, vol. 29, no. 5, pp. 1229-1262, 2017.
[22] S. Zhang, C. Bauckhage, and A. B. Cremers, "Informed haar-like features improve pedestrian detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,* pp. 947-954, 2014.
[23] OpenCV. [Online] Available: http://opencv.org/.
[24] F. Nie, "Tsallis cross-entropy based framework for image segmentation with histogram thresholding," *Journal of Electronic Imaging*, vol. 24, no. 1, https://doi.org/10.1117/1.JEI.24.1.013002, 2015.
[25] D. Moses, C. Sammut, and T. Zrimec, "Automatic segmentation and analysis of the main pulmonary artery on standard post-contrast CT studies using iterative erosion and dilation," *International journal of computer assisted radiology and surgery*, vol. 11, no. 3, pp. 381-395, 2016.
[26] N. Khaneja, et al., "Optimal Control of Coupled Spin Dynamics: Design of NMR Pulse Sequences by Gradient Ascent Algorithms," *Journal of Magnetic Resonance*, vol. 172, no. 2, pp. 296-305, 2005.
[27] C. Dupuy and F. Bach, "Online but accurate inference for latent variable models with local Gibbs sampling," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 4581-4625, 2017.
[28] UCI Machine Learning Repository. [Online] Available: http://archive.ics.uci.edu/ml/.
[29] A. Bahrololoum, et al., "A Prototype Classifier Based on Gravitational Search Algorithm," *Applied Soft Computing*, vol. 12, pp. 819-825, 2012.

**Chih-Wei Chien** received the B.S. and the M.S. degree from the Department of Electrical Engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, R.O.C., in 2014 and 2016, respectively. His major research interests include humanoid robot, deep learning, intelligent system, and robotic application.

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

**Ting-Nan Tsai** received the M.S. degree from the Department of Electrical Engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, R.O.C., in 2016, and pursuing the B.S. degree in NCKU currently. His major research interests include humanoid robot and robotic application.

**Li-Fan Wu** received the B.S. degree from the Department of Mechanical Engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, R.O.C., in 2015. He received the M.S. degree in the Department of Electrical Engineering, NCKU, Tainan, Taiwan, R.O.C., in 2017. His major research interests include fuzzy control, intelligent system, humanoid robot, gait pattern generation, and robotic application.

**Nien-Chu Fang** received the B.S. and the M.S. degree from the Department of Electrical Engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, R.O.C., in 2015 and 2017, respectively. His major research interests include humanoid robot, robot cooperation, robot team formation, intelligent system, and image processing.

**Tzuu-Hseng S. Li** received the B.S. degree in electrical engineering from the Tatung Institute of Technology, Taipei, Taiwan (R.O.C.), in 1981, and the M.S. and Ph.D. degrees in electrical engineering from the National Cheng Kung University (NCKU), Tainan, Taiwan, in 1985 and 1989, respectively. Since 1985, he has been with the Department of Electrical Engineering, NCKU, where he is currently a Distinguished Professor. From 1996 to 2009, he was a Researcher with the Engineering and Technology Promotion Center, National Science Council, Tainan. From 1999 to 2002, he was the Director of the Electrical Laboratories, NCKU. From 2009 to 2012, he was the Dean of the College of Electrical Engineering and Computer Science, National United University, Miaoli City, Taiwan. He was the Vice President of the Federation of International Robot-Soccer Association (FIRA) from 2009 to 2016. He has been the Director of the Center for Intelligent Robotics and Automation, NCKU, since 2014. His current research interests include artificial and/or biological intelligence and applications, fuzzy system and control, home service robots, humanoid robots, mechatronics, 4WIS4WID vehicles, and singular perturbation methodology.

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

# Intelligent Integral Terminal Sliding-Mode Consensus Formation Control for Heterogeneous Networking Omnidirectional Mobile Multi-Robots

Hsiao-Lang Wu, Ching-Chih Tsai

***Abstract*—This paper presents an intelligent integral terminal sliding-mode consensus formation control using output recurrent fuzzy wavelet neural networks (ORFWNN) for a group of networking heterogeneous omnidirectional mobile robots (OMRs) with uncertainties. Each uncertain OMR is modelled by a reduced three-input-three-output second-order state equation with uncertainties and the multi-robot system is modeled by directed graph theory. By using the sliding mode control theory and online learning of ORFWNN, an intelligent integral terminal sliding mode consensus control approach is presented to achieve formation control in presence of uncertainties. Three simulations are conducted to show the effectiveness and merits of the proposed method.**

***Keywords:*** **Consensus formation control, heterogeneous, integral terminal sliding-mode (ITSM) control, omnidirectional mobile robots (OMRs), output recurrent fuzzy wavelet neural networks (ORFWNN).**

## I. Introduction

Homogeneous cooperative control problems have been widely investigated by many researchers. They obtained many successful applications to multi-vehicles, multi-robots and multi-agents with the almost same system dynamics [1]-[7]. Consensus formation control approach is one of the useful formation control laws with the benefits of network flexibility. In contrast, heterogeneous consensus control problems have also received growing attention among researchers; especially, the authors in [8]-[10] proposed formation control approaches for a class of heterogeneous multi-agents and multi-robots with different nonlinear dynamics. However, the multi-robot formation control systems in [9]-[10] dealt with the same type of mobile multi-robots with different physical sizes, which are not really heterogeneous. Motivated by [8]-[10], heterogeneous networking multi-robot systems deserve further studies for consensus formation control of many types of omnidirectional mobile multi-robots.

Omnidirectional mobile robots (OMRs) been shown to be useful for building robots or vehicles for our daily life or material handling in automation factories due to their omnidirectional capability. OMRs can be constructed by using different omnidirectional-wheeled arrangements, such as car-like four-wheeled OMRs in [9], three-wheeled OMRs in [10], and four-wheeled OMRs in [11]. Although these three above-mentioned OMRs have heterogeneous motion characteristics, the existing methods in [9]-[11] revealed that the dynamic behavior of these three types of OMRs could be reduced to a unified dynamic model so as to easily derive their consensus formation control laws.

Terminal sliding mode control (TSMC) has been developed to reach the equilibrium in finite time rather than exponential in the sliding mode control [12]-[13]. The TSMC adopted nonlinear hyperplanes and owns the property of making system convergence speed which can be changed by adjusting terminal parameters. In recent years, TSMC control method has gained numerous successful applications on robot control and other system identification techniques [14]-[15]. On the other hand, the authors in [16] claimed that TSMC may suffer from steady-state errors under parameter variations, and then proposed an integral terminal sliding-mode (ITSM) control to improve the performance of TSMC. However, the ITSM control in [16] was proposed for single-input single-output systems, and it hasn't been applied to address the multi-input multi-output (MIMO) consensus formation control problem for uncertain networking OMRs yet.

Fuzzy wavelet neural networks (FWNN) have been proved to excellently approximate time-varying nonlinear functions or nonlinear dynamics [17]. This merit can be easily applied to controller design for many systems. For example, Tsai and Wu [18] proposed an integrated method combining NTSM control and FWNN for trajectory tracking of OMRs. Output recurrent fuzzy wavelet neural networks are a class of recurrent FWNNs whose one-sample delayed, amplitude-modulated outputs are feedback to the hidden-layered neurons, in order to achieve higher approximation accuracy and faster convergence [19]. Inspired by [17]-[19], the ORFWNN would be helpful in designing an intelligent consensus formation controller for a class of heterogeneous OMRs.

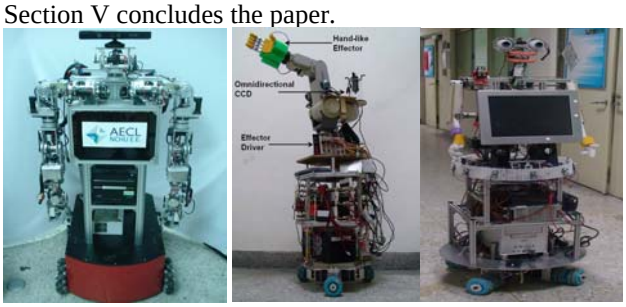
The objectives of the paper are to derive the intelligent ITSM consensus formation control laws for a group of uncertain heterogeneous networking OMRs and to verify the feasibility and effectiveness of the proposed controller via simulations. By comparing to the state of the art in the field of consensus formation control, the paper is novel in proposing stable and intelligent ITSM consensus formation control laws for a group of uncertain networking OMRs using sliding mode theory and consensus algorithm.

The rest of this paper is constructed as follows. Section II states the modeling and formation formulation. Section III designs the intelligent ITSM finite-time consensus formation control, Simulations are conducted to demonstrate the

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 106-2221-E-005-003.

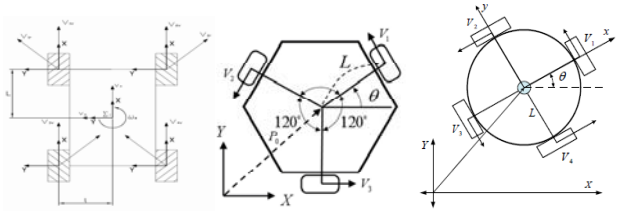
H. S. Wu and C.C. Tsai are with the Department of Electrical Engineering, National Chung Hsing University, Taichung, 40227, Taiwan. (e-mail: u670152827@gmail.com, cctsai@nchu.edu.tw).
Corresponding author: Ching-Chih Tsai (cctsai@nchu.edu.tw)

performance of the proposed approaches in Section IV. Section V concludes the paper.



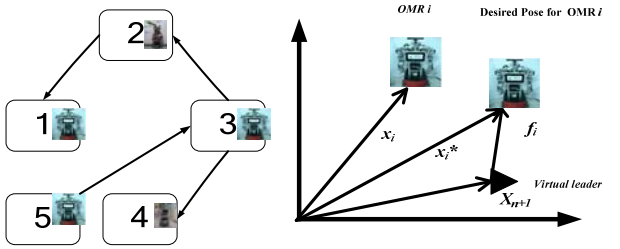

(a) (b) (c)

Fig. 1. Three types of OMRs; (a) a picture of a car-like four-wheeled OMR; (b) a picture of a three-wheeled OMR;(c) a picture of a four-wheeled OMR.



(a) (b) (c)

Fig.2. Structures and geometries of the omnidirectional mobile robots; (a) car-like four-wheeled OMR ;(b)three-wheeled OMR; (c) four-wheeled OMR.




(a) (b)

Fig. 3. The interconnected OMR system; (a) a formation example of the four follower OMRs with the fifth robot as a virtual leader;(b) illustration of the current and goal pose of OMR *i* .

## II. Problem Formulation

This section aims to first introduce a unified dynamic model for three kinds of OMRs, which are the car-like four-wheeled OMR, three-wheeled OMR and four-wheeled OMR, and a graph theory based model of the multi-OMRs system, and finally formulating the control problem. The three kinds of OMRs under consideration are mobile robots respectively equipped with four 45-degree Omnidirectional wheels, three 90-degree Omnidirectional wheels and four 90-degree Omnidirectional wheels, as shown in Fig. 1 (a-c).

### *2.1 Modeling any OMRs in a Unified Framework*

This subsection will establish a unified dynamic equation of three types of OMRs in a unified framework. Therefore, we define the position, velocity and bounded disturbance vectors as $\mathbf{x}_i = [x_i \;\; y_i \;\; \theta_i]^T$, $\mathbf{v}_i = [\dot{x}_i \;\; \dot{y}_i \;\; \dot{\theta}_i]^T$ and $\boldsymbol{\xi}_i = \left[\xi_{ix} \;\; \xi_{iy} \;\; \xi_{i\theta}\right]^T$, then perform some simple operations for the control vector for the $i^{th}$ OMR which can be car-like four-wheeled OMR, three-wheeled OMR and four-wheeled OMR in [9-11]. Hence, we obtain a unified dynamic model of the $i^{th}$ OMR in the following vector-matrix unified form;

$$\dot{\mathbf{x}}_i = \mathbf{v}_i, \dot{\mathbf{v}}_i = -\mathbf{g}_i + \mathbf{u}_i + \boldsymbol{\xi}_i \tag{1}$$

where $\mathbf{u}_i \in R^3$ and $\mathbf{g}_i \in R^3$ respectively denotes the control and uncertain vectors which are easily derived from [9-11]. $\boldsymbol{\xi}_i$ is the bounded disturbance satisfying $|\boldsymbol{\xi}_i| \le h_{\xi\max}$ .

### *2.2 Modeling a Networking Multi-OMR System*

Suppose that interconnection topology of $n$ follower OMRs is a directed subgraph $G$, and $n$ networking OMRs can be regarded as $n$ nodes. The relevant weighted adjacency matrix of the subgraph $G$ is denoted as $\mathbf{A} = [a_{ij}]$ and $a_{ij} \ge 0, \forall i, j \in \{1, 2, ..., n\}$. Moreover we assume $a_{ii}$ =0. The Laplacian matrix $\mathbf{L} \in R^{n\times n}$ of the directed subgraph $G$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D} = diag(d_1, d_2, ..., d_n)$, and $d_i = \sum_{j=1}^{n} a_{ij}$. Therefore, the whole multi-OMR system consists of the $n$ follower OMRs and a virtual leader OMR considered as the $n+1^{th}$ OMR. The interconnection topology of the overall multi-OMR system is also a directed graph $\bar{G}$, and its Laplacian matrix $\bar{\mathbf{L}}$ is given by $\bar{\mathbf{L}} = \bar{\mathbf{D}} - \bar{\mathbf{A}}$. Fig. 2(a) illustrates a formation example of the four follower OMRs with the fifth robot as the virtual leader. In order to achieve this consensus formation control objective, three assumptions about the communication topology are made in the following.

*Assumption 1:* The graph $\bar{G}$ with its relevant Laplacian matrix $\bar{\mathbf{L}}$ is directed and has a spanning tree with root being the virtual leader, the $n+1^{th}$ OMR.
*Assumption 2:* Every OMR must be connected from the leader on the network, but all of the OMRs are not necessarily directly connected from the leader.
*Assumption 3:* Movement of the leader must be independent from any OMR.

According to Assumption 1, the Laplacian matrix $\bar{\mathbf{L}} \in R^{(n+1)\times(n+1)}$ is symmetric and has only one zero eigenvalue and all other nonzero eigenvalues with positive real parts. On basis of Assumptions 1-3, all the entries in the last row of Laplacian matrix $\bar{\mathbf{L}} \in R^{(n+1)\times(n+1)}$ are zeros, and there exists a diagonal matrix $\mathbf{B} = diag\{a_{1(n+1)}, ..., a_{n(n+1)}\} \in R^{n\times n}$ where it has at least one positive diagonal entry such that the summation of both matrices, i.e., $\mathbf{L} + \mathbf{B}$ is invertible and has all the nonzero eigenvalues with positive real parts [3],[8].

### *2.3 Problem Statement*

This subsection considers a mission that $n$ OMRs follow their virtual leader, the follower OMRs and the leader move together in consensus formation. The leader individually provides its own position and desired positions to the only directly connected OMRs in consensus formation. Hence, the proposed control law is aimed at not only achieving that a group of OMRs cooperatively moves in consensus formation, but also accomplishing that each OMR generates the

geometric configuration of the consensus formation. Hence, the control objective to achieve this mission is delineated as follows.

Each of the OMRs follows its leader, and its pose asymptotically converges to the desired pose given by leader. This is formulated as $\mathbf{x}_i(t) \to \mathbf{x}_i^*(t) = 0$, and $\dot{\mathbf{x}}_i(t) \to \dot{\mathbf{x}}_i^*(t) = 0$, $i$=1,…,$n$, in finite time, where $\mathbf{x}_i^*(t) = \mathbf{x}_{n+1}(t) + \mathbf{f}_i(t)$ is denoted by the desired pose of the $i^{th}$ OMR as shown in Fig. 3(b), $\mathbf{x}_i(t)$ is the current pose of the $i^{th}$ OMR at time $t$, $\mathbf{x}_{n+1}(t)$ is the pose of the leader, and $\mathbf{f}_i(t)$ is the desired relative pose vector between the $i^{th}$ OMR and leader. Note that $\mathbf{f}_{n+1}(t) = 0$, $\dot{\mathbf{f}}_{n+1}(t) = 0$. However, each OMR may not get its desire pose from the leader due to the communication topology. Therefore, the control goal can be re-stated in the consensus terminology, namely that $\bar{\mathbf{x}}_i(t) \to \bar{\mathbf{x}}_j(t) = 0$, and $\dot{\bar{\mathbf{x}}}_i(t) \to \dot{\bar{\mathbf{x}}}_j(t) = 0, \quad \forall i, j \in \{1, 2, ..., n+1\}$, in finite time, where $\bar{\mathbf{x}}_j(t) = \mathbf{x}_j(t) - \mathbf{f}_j(t)$. To achieve the intelligent ITSM consensus formation control goal, it is necessary to make the two more assumptions.

*Assumption 4:* $\mathbf{x}_{n+1}(t)$ and $\mathbf{f}_i(t)$, $i$=1,..,$n$, are twice differentiable. $\mathbf{f}_i(t)$, $\dot{\mathbf{f}}_i(t)$ and $\ddot{\mathbf{f}}_i(t)$, $i$=1,..,$n$, are known for the $i^{th}$ OMR.

*Assumption 5:* The pose vector and its rate of the $i^{th}$ OMR are known.

## III. Intelligent ITSM Finite-Time Consensus Formation Control

This section will derive an intelligent ITSM finite time consensus formation control law using ORFWNN AND sliding mode theory. The ORFWNN will be used to on-line learn the nonlinear and state-dependent function, $\boldsymbol{g}_i$, in(1). The intelligent ITSM consensus formation controller is synthesized to not only move a group of heterogeneous OMRs in consensus formation, but also steer each OMR to track desired trajectories.

### *3.1 ORFWNN Function Approximation*

Fig. 4 depicts the schematic diagram of the proposed ORFWNN structure whose three layers are described by

$$\hat{\mathbf{g}}_\mathbf{i} = \sum_{j=1}^{M} w_{ij}\varphi_j(x) = \hat{\mathbf{W}}_i^T \hat{\varphi}(\mathbf{x}, \hat{\mathbf{c}}, \hat{\boldsymbol{\omega}}), i = 1, 2, 3. \quad (2)$$

where $\hat{\mathbf{W}}_i = [\hat{w}_{i1} \quad \hat{w}_{i2} \quad \cdots \quad \hat{w}_{iM}]^T$, $\hat{\boldsymbol{\varphi}}_i = [\hat{\varphi}_1 \quad \hat{\varphi}_2 \quad \cdots \quad \hat{\varphi}_M]^T$; $m$ is the number of input nodes and $M$ is the number of the wavelet functions given by

$$\varphi_i(x(k)) = \prod_{j=1}^{m} \phi_i(\bar{x}_j) \quad (3)$$

$$\phi_i(\bar{x}_j(k)) = \left[1 - \hat{\omega}_{ij}^2(\bar{x}_j(k) - \hat{c}_{ij})^2\right] e^{\left\{-\hat{\omega}_{ij}^2(\bar{x}_j(k)-\hat{c}_{ij})^2\right\}} \quad (4)$$

where $\bar{x}_j = x_j + \sum_i^3 \delta_{ji}\varphi_i(x(k-1))$ and $\delta_{ji}$ denotes the output recurrent constant parameter. Next, use the fuzzy wavelet network to approximate the nonlinear term $\boldsymbol{g}_i$. Let the ideal approximation result of the nonlinear function in (1) be

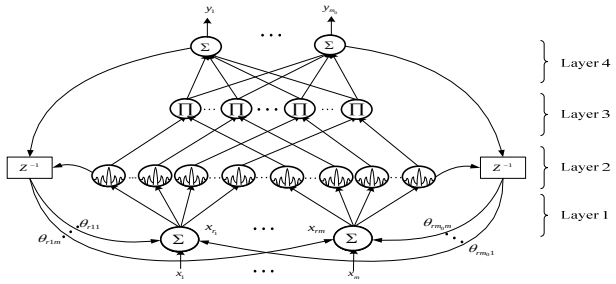



Fig. 4. Structure of the ORFWNN.

$$\mathbf{g}_\mathbf{i} = \mathbf{W}_\mathbf{i}^{*T} \boldsymbol{\varphi}_\mathbf{i}^*(\mathbf{x}_\mathbf{i}, \mathbf{c}_\mathbf{i}^*, \boldsymbol{\omega}_\mathbf{i}^*) + \boldsymbol{\varepsilon}_{\mathbf{ig}}^* \quad (5)$$

where $\mathbf{W}_i^*$ is the optimal weight vector; $\boldsymbol{\varphi}_i^*$ is the optimal fuzzy wavelet basis function vector; $\boldsymbol{\varepsilon}_{ig}^*$ is the small and bounded error vector. Note that $\hat{\mathbf{W}}_\mathbf{i}$, $\hat{\boldsymbol{\varphi}}_\mathbf{i}$, $\hat{\mathbf{c}}_\mathbf{i}$, $\hat{\boldsymbol{\omega}}_\mathbf{i}$ in (2) are estimates of their corresponding optimal true $\mathbf{W}_\mathbf{i}^*$, $\boldsymbol{\varphi}_\mathbf{i}^*$, $\mathbf{c}_\mathbf{i}^*$, $\boldsymbol{\omega}_\mathbf{i}^*$ in (5), and all the norms of these optimal weights are bounded. Defining $\tilde{\mathbf{W}}_\mathbf{i} = \mathbf{W}_\mathbf{i}^* - \hat{\mathbf{W}}_\mathbf{i}$, $\tilde{\boldsymbol{\varphi}}_\mathbf{i} = \boldsymbol{\varphi}_\mathbf{i}^* - \hat{\boldsymbol{\varphi}}_\mathbf{i}$, $\tilde{\boldsymbol{\omega}}_\mathbf{i} = \boldsymbol{\omega}_\mathbf{i}^* - \hat{\boldsymbol{\omega}}_\mathbf{i}$, $\tilde{\mathbf{c}}_\mathbf{i} = \mathbf{c}_\mathbf{i}^* - \hat{\mathbf{c}}_\mathbf{i}$, we have

$$\begin{aligned} \tilde{\mathbf{g}}_i &= \mathbf{g}_i - \hat{\mathbf{g}}_i = \hat{\mathbf{W}}_\mathbf{i}^T \tilde{\boldsymbol{\varphi}}_\mathbf{i} + \tilde{\mathbf{W}}_\mathbf{i}^T \hat{\boldsymbol{\varphi}}_\mathbf{i} + \tilde{\mathbf{W}}_\mathbf{i}^T \tilde{\boldsymbol{\varphi}}_\mathbf{i} + \boldsymbol{\varepsilon}_{\mathbf{if}}^* \\ &= \hat{\mathbf{W}}_\mathbf{i}^T \mathbf{A}_\mathbf{i} \tilde{\boldsymbol{\omega}}_\mathbf{i} + \hat{\mathbf{W}}_\mathbf{i}^T \mathbf{B}_\mathbf{i} \tilde{\mathbf{c}}_\mathbf{i} + \tilde{\mathbf{W}}_\mathbf{i}^T \hat{\boldsymbol{\varphi}}_\mathbf{i} + \mathbf{h}_i \end{aligned} \quad (6)$$

where $\tilde{\boldsymbol{\varphi}}_\mathbf{i} = \boldsymbol{\varphi}_\mathbf{i}^* - \hat{\boldsymbol{\varphi}}_\mathbf{i}$ is expanded by Taylor series as

$$\tilde{\boldsymbol{\varphi}}_i = \frac{\partial \tilde{\boldsymbol{\varphi}}_i}{\partial \hat{\boldsymbol{\omega}}_i} \tilde{\boldsymbol{\omega}}_i + \frac{\partial \tilde{\boldsymbol{\varphi}}_i}{\partial \hat{\mathbf{c}}_i} \tilde{\mathbf{c}}_i + \mathbf{H}_i = \mathbf{A}_\mathbf{i} \tilde{\boldsymbol{\omega}}_\mathbf{i} + \mathbf{B}_\mathbf{i} \tilde{\mathbf{c}}_\mathbf{i} + \mathbf{H}_\mathbf{i} \quad (7)$$

and $\mathbf{H}_\mathbf{i}$ contains the higher order terms and bounded $\mathbf{h}_i = \hat{\mathbf{W}}_\mathbf{i}^T \mathbf{H}_\mathbf{i}$. $+\tilde{\mathbf{W}}_\mathbf{i}^T \tilde{\boldsymbol{\varphi}}_\mathbf{i} + \boldsymbol{\varepsilon}_{\mathbf{if}}^*$ and $\|\mathbf{h}_i\|_\infty < h_{i\max} < \infty$.

### *3.2 Intelligent ITSM Consensus Formation Control*

This section will develop an intelligent ITSM consensus formation controller for the heterogeneous multi-OMR system in which each OMR has the system model (1) with uncertainty in $\mathbf{g}_i$. In particular, the uncertainties can be decomposed into two portions: nominal and perturbed, i.e, $\mathbf{g}_i = \mathbf{g}_i^0 + \tilde{\mathbf{g}}_i$. Assume that the nominal potion is known, and perturbed portion is unknown but bounded, i.e, $\|\tilde{\mathbf{g}}_i\| \le h_{i\max}$. The aim of the intelligent ITSM formation controller is, given the system model (1), to design each ITSM consensus formation control law $\mathbf{u}_i$ to fulfill consensus tracking in finite time, i.e., $\bar{\mathbf{x}}_i(t) - \bar{\mathbf{x}}_j(t) = 0$, and $\dot{\bar{\mathbf{x}}}_i(t) - \dot{\bar{\mathbf{x}}}_j(t) = 0$, $i$, $j$= 1, …,n+1, namely that all the $n$ follower OMRs converge to their desired poses and pose velocities given by the leader OMR. In doing so, the overall consensus formation system based on the system model (1) can be rewritten in a vector-matrix form

$$\dot{\mathbf{x}} = \mathbf{v}, \ \dot{\mathbf{v}} = \mathbf{u} - \mathbf{g} + \boldsymbol{\xi} \quad (8)$$

where $\mathbf{x} = (\mathbf{x}_1^T, ..., \mathbf{x}_n^T)^T \in R^{3n}$, $\mathbf{v} = (\mathbf{v}_1^T, ..., \mathbf{v}_n^T)^T \in R^{3n}$, $\mathbf{u} = (\mathbf{u}_1^T, \mathbf{u}_2^T, ..., \mathbf{u}_n^T)^T \in R^{3n}$, $\mathbf{g} = (\mathbf{g}_1^T ..., \mathbf{g}_n^T)^T \in R^{3n}$ and

$\xi = (\xi_1^{\mathrm{T}}, ..., \xi_n^{\mathrm{T}})^T \in R^{3n}$.

To design the intelligent ITSM decentralized consensus formation controller, one defines the two formation error and error rate vectors for the i$^{th}$ OMR as

$$\mathbf{e}_{1i} = \sum_{j=1}^{n+1} a_{ij}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j), i = 1, ..., n \tag{9}$$

$$\mathbf{e}_{2i} = \sum_{j=1}^{n+1} a_{ij}(\dot{\bar{\mathbf{x}}}_i - \dot{\bar{\mathbf{x}}}_j), i = 1, ..., n \tag{10}$$

where $\bar{\mathbf{x}}_j(t) = \mathbf{x}_j(t) - \mathbf{f}_j(t), j = 1, .., n+1$ .Therefore, the formation error and error rate vectors for the whole networking multi-OMR system can be written in the vector-matrix form.

$$\begin{aligned} \mathbf{e}_1 &= (\mathbf{L} + \mathbf{B}) \otimes \mathbf{I}_3(\mathbf{x} - \mathbf{x}^*) \\ \mathbf{e}_2 &= (\mathbf{L} + \mathbf{B}) \otimes \mathbf{I}_3(\dot{\mathbf{x}} - \dot{\mathbf{x}}^*) \end{aligned} \tag{11}$$

where $\mathbf{x}^* = (\mathbf{x}_1^{*\mathbf{T}}, ..., \mathbf{x}_n^{*\mathbf{T}})^T \in R^{3n}$ is the desired poses, $\otimes$ is the Kronecter product, and the matrix $\mathbf{L} + \mathbf{B} \in R^{3n \times 3n}$ is reduced from the graph Laplacian $\bar{\mathbf{L}}$ and given by

$$\mathbf{L} + \mathbf{B} = \begin{bmatrix} \sum_{j=1}^{n+1} a_{1j} & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & \sum_{j=1}^{n+1} a_{2j} & \cdots & -a_{2n} \\ \cdots & \cdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & \sum_{j=1}^{n+1} a_{nj} \end{bmatrix}$$

Note that the matrix $\mathbf{L} + \mathbf{B}$ has all the nonzero eigenvalues with positive real parts where the five assumptions hold.

To design the intelligent ITSM consensus formation control law for the whole networking multi-OMR system, one defines

$$\mathbf{e}_0 = \int_0^t \mathbf{e}_1(\tau) d\tau \tag{12}$$

Therefore, one obtains

$$\begin{aligned} \dot{\mathbf{e}}_0 &= \mathbf{e}_1 \\ \dot{\mathbf{e}}_1 &= \mathbf{e}_2 \\ \dot{\mathbf{e}}_2 &= (\mathbf{L} + \mathbf{B}) \otimes \mathbf{I}_3(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}^*) = (\mathbf{L} + \mathbf{B}) \otimes \mathbf{I}_3(\mathbf{u} - \mathbf{g} + \boldsymbol{\xi} - \ddot{\mathbf{x}}^*) \end{aligned} \tag{13}$$

Next, we choose the two ITSM sliding surface $\mathbf{S}_1$ and $\mathbf{S}_2$ as

$$\mathbf{S}_1 = \mathbf{e}_1 + \boldsymbol{\alpha}_1 \mathbf{e}_0^{\gamma_1} + \boldsymbol{\beta}_1 \mathbf{e}_0^{\rho_1} \tag{14}$$

$$\mathbf{S}_2 = \dot{\mathbf{S}}_1 + \boldsymbol{\alpha}_2 \mathbf{S}_1^{\gamma_2} + \boldsymbol{\beta}_2 \mathbf{S}_1^{\rho_2} \tag{15}$$

where $0.5 < \gamma_1 < 1$ , $0.5 < \gamma_2 < 1$ and $\rho_1 = p_1 / q_1 > 1$ , $\rho_2 = p_2 / q_2 > 1$ where $q_1$ and $q_2$ are positive and odd numbers, $p_1$ and $p_2$ are positive and even numbers; $\boldsymbol{\alpha}_1 = diag[\boldsymbol{\alpha}_{11}\ \boldsymbol{\alpha}_{12} \cdots \boldsymbol{\alpha}_{1n}] \in R^{3n \times 3n}$ , $\boldsymbol{\alpha}_2 = diag[\boldsymbol{\alpha}_{21}\ \boldsymbol{\alpha}_{22} \cdots \boldsymbol{\alpha}_{2n}] \in R^{3n \times 3n}$ , $\boldsymbol{\beta}_1 = diag[\boldsymbol{\beta}_{11}\ \boldsymbol{\beta}_{12} \cdots \boldsymbol{\beta}_{1n}] \in R^{3n \times 3n}$ and $\boldsymbol{\beta}_2 = diag[\boldsymbol{\beta}_{21}\ \boldsymbol{\beta}_{22} \cdots \boldsymbol{\beta}_{2n}] \in R^{3n \times 3n}$ are diagonal and positive-definite where $\boldsymbol{\alpha}_{1i} = diag[\alpha_{1i1}, \alpha_{1i2}, \alpha_{1i3}] \in R^{3 \times 3}$ , $\boldsymbol{\alpha}_{2i} = diag[\alpha_{2i1}, \alpha_{2i2}, \alpha_{2i3}] \in R^{3 \times 3}$ , $\boldsymbol{\beta}_{1i} = diag[\beta_{1i1}, \beta_{1i2}, \beta_{1i3}] \in R^{3 \times 3}$ and $\boldsymbol{\beta}_{2i} = diag[\beta_{2i1}, \beta_{2i2}, \beta_{2i3}] \in R^{3 \times 3}$ are also diagonal and positive-definite .

$\mathbf{e}_0^{\,p} = \left[ (\mathbf{e}_{01}^{\,p})^T \quad (\mathbf{e}_{02}^{\,p})^T \quad \cdots \quad (\mathbf{e}_{0n}^{\,p})^T \right]^T, p = \gamma_1, \rho_1$ and $\mathbf{S}_1^{\,q} = \left[ (\mathbf{S}_{11}^{\,q})^T \quad (\mathbf{S}_{12}^{\,q})^T \quad \cdots \quad (\mathbf{S}_{1n}^{\,q})^T \right]^T, q = \gamma_2, \rho_2$ where $\mathbf{e}_{0i}^{\,p} = \left[ e_{0i1}^{\,p} \quad e_{0i2}^{\,p} \quad e_{0i3}^{\,p} \right]^T$ and $\mathbf{S}_{1i}^{\,q} = \left[ \mathbf{S}_{1i1}^{\,q} \quad \mathbf{S}_{1i2}^{\,q} \quad \mathbf{S}_{1i3}^{\,q} \right]^T$ . Moreover, $\mathbf{S}_1 = [\mathbf{S}_{11}^{\,T}, ..., \mathbf{S}_{1n}^{\,T}]^T$, $\mathbf{S}_{1j} \in R^3$, $j = 1, 2, ..., n$.

Differentiating (14) and (15) gives

$$\begin{aligned} \dot{\mathbf{S}}_1 &= \dot{\mathbf{e}}_1 + \boldsymbol{\alpha}_1 \gamma_1 diag\left(\mathbf{e}_0^{\gamma_1 - 1}\right) \dot{\mathbf{e}}_0 + \boldsymbol{\beta}_1 \rho_1 diag\left(\mathbf{e}_0^{\rho_1 - 1}\right) \dot{\mathbf{e}}_0 \\ &= \mathbf{e}_2 + \boldsymbol{\alpha}_1 \gamma_1 diag\left(\mathbf{e}_0^{\gamma_1 - 1}\right) \mathbf{e}_1 + \boldsymbol{\beta}_1 \rho_1 diag\left(\mathbf{e}_0^{\rho_1 - 1}\right) \mathbf{e}_1 \end{aligned} \tag{16}$$

$$\dot{\mathbf{S}}_2 = \ddot{\mathbf{S}}_1 + \boldsymbol{\alpha}_2 \gamma_2 diag\left(\mathbf{S}_1^{\gamma_2 - 1}\right) \dot{\mathbf{S}}_1 + \boldsymbol{\beta}_2 \rho_2 diag\left(\mathbf{S}_1^{\rho_2 - 1}\right) \dot{\mathbf{S}}_1 \tag{17}$$

Moreover

$$\begin{aligned} \ddot{\mathbf{S}}_1 &= \ddot{\mathbf{e}}_2 + \boldsymbol{\alpha}_1 \gamma_1 diag\left(\mathbf{e}_0^{\gamma_1 - 1}\right) \dot{\mathbf{e}}_1 + \boldsymbol{\beta}_1 \rho_1 diag\left(\mathbf{e}_0^{\rho_1 - 1}\right) \dot{\mathbf{e}}_1 \\ &\quad + \boldsymbol{\alpha}_1 \gamma_1 (\gamma_1 - 1) diag\left(\mathbf{e}_0^{\gamma_1 - 2}\right) diag\left(\dot{\mathbf{e}}_0\right) \mathbf{e}_1 \\ &\quad + \boldsymbol{\beta}_1 \rho_1 (\rho_1 - 1) diag\left(\mathbf{e}_0^{\rho_1 - 2}\right) diag\left(\dot{\mathbf{e}}_0\right) \mathbf{e}_1 \\ &= \ddot{\mathbf{e}}_2 + \boldsymbol{\alpha}_1 \gamma_1 diag\left(\mathbf{e}_0^{\gamma_1 - 1}\right) \dot{\mathbf{e}}_1 + \boldsymbol{\beta}_1 \rho_1 diag\left(\mathbf{e}_0^{\rho_1 - 1}\right) \dot{\mathbf{e}}_1 \\ &\quad + \boldsymbol{\alpha}_1 \gamma_1 (\gamma_1 - 1) diag\left(\mathbf{e}_0^{\gamma_1 - 2}\right) diag\left(\mathbf{e}_1\right) \mathbf{e}_1 \\ &\quad + \boldsymbol{\beta}_1 \rho_1 (\rho_1 - 1) diag\left(\mathbf{e}_0^{\rho_1 - 2}\right) diag\left(\mathbf{e}_1\right) \mathbf{e}_1 \\ &= (\mathbf{L} + \mathbf{B}) \otimes \mathbf{I}_3(\mathbf{u} - \mathbf{g} + \boldsymbol{\xi} - \ddot{\mathbf{x}}^*) + \boldsymbol{\alpha}_1 \gamma_1 diag\left(\mathbf{e}_0^{\gamma_1 - 1}\right) \mathbf{e}_2 \\ &\quad + \boldsymbol{\beta}_1 \rho_1 diag\left(\mathbf{e}_0^{\rho_1 - 1}\right) \mathbf{e}_2 \\ &\quad + \boldsymbol{\alpha}_1 \gamma_1 (\gamma_1 - 1) diag\left(\mathbf{e}_0^{\gamma_1 - 2}\right) diag\left(\mathbf{e}_1\right) \mathbf{e}_1 \\ &\quad + \boldsymbol{\beta}_1 \rho_1 (\rho_1 - 1) diag\left(\mathbf{e}_0^{\rho_1 - 2}\right) diag\left(\mathbf{e}_1\right) \mathbf{e}_1 \end{aligned} \tag{18}$$

Then,

$$\dot{\mathbf{S}}_2 = \ddot{\mathbf{S}}_1 + \boldsymbol{\alpha}_2 \gamma_2 diag\left(\mathbf{S}_1^{\gamma_2 - 1}\right) \dot{\mathbf{S}}_1 + \boldsymbol{\beta}_2 \rho_2 diag\left(\mathbf{S}_1^{\rho_2 - 1}\right) \dot{\mathbf{S}}_1$$

Finally, we have

$$\begin{aligned} \dot{\mathbf{S}}_2 &= (\mathbf{L} + \mathbf{B}) \otimes \mathbf{I}_3(\mathbf{u} - \mathbf{g} + \boldsymbol{\xi} - \ddot{\mathbf{x}}^*) + \boldsymbol{\alpha}_1 \gamma_1 diag\left(\mathbf{e}_0^{\gamma_1 - 1}\right) \mathbf{e}_2 \\ &\quad + \boldsymbol{\beta}_1 \rho_1 diag\left(\mathbf{e}_0^{\rho_1 - 1}\right) \mathbf{e}_2 + \boldsymbol{\alpha}_1 \gamma_1 (\gamma_1 - 1) diag\left(\mathbf{e}_0^{\gamma_1 - 2}\right) diag\left(\mathbf{e}_1\right) \mathbf{e}_1 \\ &\quad + \boldsymbol{\beta}_1 \rho_1 (\rho_1 - 1) diag\left(\mathbf{e}_0^{\rho_1 - 2}\right) diag\left(\mathbf{e}_1\right) \mathbf{e}_1 \\ &\quad + \boldsymbol{\alpha}_2 \gamma_2 diag\left(\mathbf{S}_1^{\gamma_2 - 1}\right) \left( \mathbf{e}_2 + \boldsymbol{\alpha}_1 \gamma_1 diag\left(\mathbf{e}_0^{\gamma_1 - 1}\right) \mathbf{e}_1 + \boldsymbol{\beta}_1 \rho_1 diag\left(\mathbf{e}_0^{\rho_1 - 1}\right) \mathbf{e}_1 \right) \\ &\quad + \boldsymbol{\beta}_2 \rho_2 diag\left(\mathbf{S}_1^{\rho_2 - 1}\right) \left( \mathbf{e}_2 + \boldsymbol{\alpha}_1 \gamma_1 diag\left(\mathbf{e}_0^{\gamma_1 - 1}\right) \mathbf{e}_1 + \boldsymbol{\beta}_1 \rho_1 diag\left(\mathbf{e}_0^{\rho_1 - 1}\right) \mathbf{e}_1 \right) \end{aligned} \tag{19}$$

Hence, the intelligent ITSM consensus formation control law for the whole networking multi-OMR system in formation is proposed by,

$$\begin{aligned}\mathbf{u} &= \ddot{\mathbf{f}} + (\mathbf{L}+\mathbf{B})^{-1}\otimes\mathbf{I}_3(\mathbf{b}\otimes\ddot{\mathbf{x}}_{n+1}) + \hat{\mathbf{g}} \\ &-(\mathbf{L}+\mathbf{B})^{-1}\otimes\mathbf{I}_3\left(c_0\mathbf{S}_2 + c_1\,\mathrm{sgn}(\mathbf{S}_2)\right) \\ &+\boldsymbol{\alpha}_1\gamma_1 diag\left(\mathbf{e}_0^{\gamma_1-1}\right)\mathbf{e}_2 + \boldsymbol{\beta}_1\rho_1 diag\left(\mathbf{e}_0^{\rho_1-1}\right)\mathbf{e}_2 \\ &+\boldsymbol{\alpha}_1\gamma_1(\gamma_1-1) diag\left(\mathbf{e}_0^{\gamma_1-2}\right) diag\left(\mathbf{e}_1\right)\mathbf{e}_1 \\ &+\boldsymbol{\beta}_1\rho_1(\rho_1-1) diag\left(\mathbf{e}_0^{\rho_1-2}\right) diag\left(\mathbf{e}_1\right)\mathbf{e}_1 \\ &+\boldsymbol{\alpha}_2\gamma_2 diag\left(\mathbf{S}_1^{\gamma_2-1}\right)\left(\mathbf{e}_2 + \boldsymbol{\alpha}_1\gamma_1 diag\left(\mathbf{e}_0^{\gamma_1-1}\right)\mathbf{e}_1\right) \\ &+\boldsymbol{\alpha}_2\gamma_2 diag\left(\mathbf{S}_1^{\gamma_2-1}\right)\boldsymbol{\beta}_1\rho_1 diag\left(\mathbf{e}_0^{\rho_1-1}\right)\mathbf{e}_1 \\ &+\boldsymbol{\beta}_2\rho_2 diag\left(\mathbf{S}_1^{\rho_2-1}\right)\left(\mathbf{e}_2 + \boldsymbol{\alpha}_1\gamma_1 diag\left(\mathbf{e}_0^{\gamma_1-1}\right)\mathbf{e}_1\right) \\ &+\boldsymbol{\beta}_2\rho_2 diag\left(\mathbf{S}_1^{\rho_2-1}\right)\boldsymbol{\beta}_1\rho_1 diag\left(\mathbf{e}_0^{\rho_1-1}\right)\mathbf{e}_1\Big)\end{aligned} \quad (20)$$

where $\ddot{\mathbf{f}} = [\ddot{\mathbf{f}}_1^T,...,\ddot{\mathbf{f}}_n^T]^T$ , $\hat{\mathbf{g}} = \left[\hat{\mathbf{g}}_1^{\,T},\hat{\mathbf{g}}_2^{\,T},...,\hat{\mathbf{g}}_n^{\,T}\right]^T \in R^{3n}$ ; $\mathbf{b} = [a_{1(n+1)},...,a_{n(n+1)}]^T \in R^n$ ; $c_0$ and $c_1$ are two positive control gains; the notation, sgn, means the signum function.

In the following, if Assumptions 1-5 are satisfied, then each of the OMRS with the control law (20) will be shown to achieve leader-following consensus tracking in finite time. Via the following relation

$$(\mathbf{L}+\mathbf{B})\otimes\mathbf{I}_3(\ddot{\mathbf{f}} - \ddot{\mathbf{x}}^*) = -(\mathbf{L}+\mathbf{B})\otimes\mathbf{I}_3(\mathbf{1}\otimes\ddot{\mathbf{x}}_{n+1}) = -\mathbf{b}\otimes\ddot{\mathbf{x}}_{n+1} \quad (21)$$

where $\mathbf{1} = [1,...,1]^T \in R^n$ , $\mathbf{L}\times\mathbf{1} = 0$ and $\mathbf{B}\times\mathbf{1} = \mathbf{b}$ . Thus, the intelligent ITSM consensus formation control law (20) is applied to (18), thereby obtaining the time derivative of the sliding-mode surface $S_2$ as below.

$$\begin{aligned}\dot{\mathbf{S}}_2 &= (\mathbf{L}+\mathbf{B})\otimes\mathbf{I}_3\big(\cancel{\ddot{\mathbf{f}}} + \cancel{(\mathbf{L}+\mathbf{B})^{-1}\otimes\mathbf{I}_3(\mathbf{b}\otimes\ddot{\mathbf{x}}_{n+1})} \\ &\quad -\tilde{\mathbf{g}} + \boldsymbol{\xi} - \cancel{\ddot{\mathbf{x}}^*}\big) - c_0\mathbf{S}_2 - c_1\,\mathrm{sgn}(\mathbf{S}_2) \\ &= -c_0\mathbf{S}_2 - c_1\,\mathrm{sgn}(\mathbf{S}_2) - (\mathbf{L}+\mathbf{B})\otimes\mathbf{I}_3\tilde{\mathbf{g}} + \bar{\boldsymbol{\xi}}\end{aligned} \quad (22)$$

where $\bar{\boldsymbol{\xi}} = (\mathbf{L}+\mathbf{B})\otimes\mathbf{I}_3\boldsymbol{\xi}$ and is the bounded disturbance satisfying $\|\bar{\boldsymbol{\xi}}\| \le h_{\xi\max}$ . To show the formation convergence in finite time, we propose the following Lyapunov function

$$V_2 = \frac{1}{2}\left(\mathbf{S}^T\mathbf{S} + \sum_{i=1}^{n}\frac{1}{\eta_{W_i}}tr\left(\tilde{\mathbf{W}}_\mathbf{i}^{\mathrm{T}}\tilde{\mathbf{W}}_\mathbf{i}\right) + \sum_{i=1}^{n}\frac{1}{\eta_{\omega_i}}tr\left(\tilde{\boldsymbol{\omega}}_\mathbf{i}^{\mathrm{T}}\tilde{\boldsymbol{\omega}}_\mathbf{i}\right) + \sum_{i=1}^{n}\frac{1}{\eta_{c_i}}tr\left(\tilde{\mathbf{c}}_\mathbf{i}^{\mathrm{T}}\tilde{\mathbf{c}}_\mathbf{i}\right)\right) \quad (23)$$

Taking the time derivative of *V* along the trajectories of (22) obtains

$$\begin{aligned}\dot{V}_2 &= -c_0\mathbf{S}_2^T\mathbf{S}_2 - c_1\mathbf{S}_2^T\,\mathrm{sgn}(\mathbf{S}_2) - \mathbf{S}_2^T(\mathbf{h} - \bar{\boldsymbol{\xi}}) \\ &\quad -\sum_{i=1}^{n}\mathbf{S}_2^T\left(\hat{\mathbf{W}}_\mathbf{i}\mathbf{A}_\mathbf{i}\tilde{\boldsymbol{\omega}}_\mathbf{i} + \hat{\mathbf{W}}_\mathbf{i}\mathbf{B}_\mathbf{i}\tilde{\mathbf{c}}_\mathbf{i} + \tilde{\mathbf{W}}_\mathbf{i}\hat{\boldsymbol{\varphi}}_\mathbf{i}\right) \\ &\quad +\sum_{i=1}^{n}\frac{1}{\eta_{W_i}}tr\left(\dot{\tilde{\mathbf{W}}}_i^T\tilde{\mathbf{W}}\right) + \sum_{i=1}^{n}\frac{1}{\eta_{\omega_i}}tr\left(\dot{\tilde{\boldsymbol{\omega}}}_i^{\,T}\tilde{\boldsymbol{\omega}}_i\right) + \sum_{i=1}^{n}\frac{1}{\eta_{c_i}}tr\left(\dot{\tilde{\mathbf{c}}}_i^{\,T}\tilde{\mathbf{c}}_i\right)\end{aligned} \quad (24)$$

where $\mathbf{h} = \left[\left(\mathbf{h}_1\right)^T ...,\left(\mathbf{h}_n\right)^T\right]^T \in R^{3n}$ . Thus, use the relation $\mathbf{S}_2^{\,T}\,\mathrm{sgn}(\mathbf{S}_2) = \|\mathbf{S}_2\|_1 = \sum_{i=1}^{3n}|s_{2i}|$ and both inequalities $\mathbf{S}_2^{\,T}\bar{\boldsymbol{\xi}} \le h_{\xi\max}\|\mathbf{S}_2\|_1$ and

$$\begin{aligned}-\mathbf{S}_2^{\,T}(\mathbf{L}+\mathbf{B})\otimes\mathbf{I}_3\tilde{\mathbf{g}} &\le \left\|\mathbf{S}_2^{\,T}\right\|\left\|(\mathbf{L}+\mathbf{B})\otimes\mathbf{I}_3\right\|\|\tilde{\mathbf{g}}\| \\ &= \left\|\mathbf{S}_2^{\,T}\right\|\|(\mathbf{L}+\mathbf{B})\|\|\tilde{\mathbf{g}}\| \le h_{\max}\|(\mathbf{L}+\mathbf{B})\|\|\mathbf{S}_2\|_1\end{aligned} \quad (25)$$

where $h_{\max} = \max\left[h_{1\max}, h_{2\max},..., h_{n\max}\right]$ and $h_{\xi\max} \ge \left|\bar{\xi}_i\right|$ , i=1,…,n, we have

$$\begin{aligned}\dot{V}_2 &\le -c_0\mathbf{S}_2^T\mathbf{S}_2 - \left(c_1 - \left(h_{\max} + h_{\xi\max}\right)\right)\|\mathbf{S}_2\|_1 \\ &\quad +\sum_{i=1}^{n}tr\left(\left(\frac{1}{\eta_{W_i}}\dot{\hat{\mathbf{W}}}_\mathbf{i} - \hat{\boldsymbol{\varphi}}_\mathbf{i}\mathbf{S}_{2\mathbf{i}}^{\mathrm{T}}\right)^T\tilde{\mathbf{W}}_i\right) \\ &\quad +\sum_{i=1}^{n}tr\left(\left(\frac{1}{\eta_{\omega_i}}\dot{\hat{\boldsymbol{\omega}}}_\mathbf{i} - \mathbf{A}_\mathbf{i}^{\mathrm{T}}\hat{\mathbf{W}}_\mathbf{i}^{\mathrm{T}}\mathbf{S}_{2\mathbf{i}}\right)^T\tilde{\boldsymbol{\omega}}_i\right) \\ &\quad +\sum_{i=1}^{n}tr\left(\left(\frac{1}{\eta_{c_i}}\dot{\hat{\mathbf{c}}}_i - \mathbf{B}_\mathbf{i}^{\mathrm{T}}\hat{\mathbf{W}}_\mathbf{i}^{\mathrm{T}}\mathbf{S}_{2\mathbf{i}}\right)^T\tilde{\mathbf{c}}_i\right)\end{aligned} \quad (26)$$

Let the parameter updating rules of the $i^{th}$ ORFWNN be

$$\dot{\hat{\mathbf{W}}}_\mathbf{i} = \eta_{W_i}\hat{\boldsymbol{\varphi}}_\mathbf{i}\mathbf{S}_{2\mathbf{i}}^{\mathrm{T}},\ \dot{\hat{\boldsymbol{\omega}}}_\mathbf{i} = \eta_{\omega_i}\mathbf{A}_\mathbf{i}^{\mathrm{T}}\hat{\mathbf{W}}_\mathbf{i}^{\mathrm{T}}\mathbf{S}_{2\mathbf{i}},\ \dot{\hat{\mathbf{c}}}_i = \eta_{c_i}\mathbf{B}_\mathbf{i}^{\mathrm{T}}\hat{\mathbf{W}}_\mathbf{i}^{\mathrm{T}}\mathbf{S}_{2\mathbf{i}} \quad (27)$$

Then, (26) turns out

$$\dot{V}_2 \le -c_0\mathbf{S}_2^T\mathbf{S}_2 - \left(c_1 - \left(h_{\max} + h_{\xi\max}\right)\right)\|\mathbf{S}_2\|_1 \quad (28)$$

which is negative-definite since $c_1 - (h_{\max} + h_{\xi\max}) > 0$ and $-\mathbf{S}_2^{\,T}\mathbf{S}_2$ is negative-definite. From the Lyapunov stability, it implies that $\mathbf{S}_2 \to 0_{3n}$ asymptotically. Once the sliding surface $\mathbf{S}_2$ has been shown to converge to a 3n by 1 zero vector, one easily shows from (15) that $\mathbf{S}_1 \to 0_{3n}$ and $\dot{\mathbf{S}}_1 \to 0_{3n}$ in finite time, and $\mathbf{e}_0 \to 0_{3n}, \mathbf{e}_1 \to 0_{3n}$ in finite time. From (14), one obtains $\mathbf{S}_1 \to 0_{3n}$ in finite time. Since $\mathbf{S}_1 = 0 = \mathbf{e}_1 + \boldsymbol{\alpha}_1\mathbf{e}_0^{\gamma_1} + \boldsymbol{\beta}_1\mathbf{e}_0^{\rho_1}$ , it follows that $\mathbf{e}_1 = \dot{\mathbf{e}}_0 = -\boldsymbol{\alpha}_1\mathbf{e}_0^{\gamma_1} - \boldsymbol{\beta}_1\mathbf{e}_0^{\rho_1} \to 0_{3n}$ and $\mathbf{e}_1, \mathbf{e}_0 \to 0_{3n}$ in finite time and $\mathbf{e}_2 \to 0_{3n}$ since $\dot{\mathbf{S}}_1 \to 0_{3n}$ . Thus

$$\begin{bmatrix}\mathbf{x} - \mathbf{x}^* \\ \dot{\mathbf{x}} - \dot{\mathbf{x}}^*\end{bmatrix} = \begin{bmatrix}(\mathbf{L}+\mathbf{B})^{-1}\otimes\mathbf{I}_3 0_{3n} \\ (\mathbf{L}+\mathbf{B})^{-1}\otimes\mathbf{I}_3 0_{3n}\end{bmatrix} \Rightarrow \begin{bmatrix}\mathbf{x} \\ \dot{\mathbf{x}}\end{bmatrix} = \begin{bmatrix}\mathbf{x}^* \\ \dot{\mathbf{x}}^*\end{bmatrix} \quad (29)$$

Hence, the results in (29) imply that $\mathbf{x}_i(t) \to \mathbf{x}_i^{\,*}(t) = 0$, and $\dot{\mathbf{x}}_i(t) \to \dot{\mathbf{x}}_i^{\,*}(t) = 0$ , *i*=1,…,*n*, in finite time, namely that $\bar{\mathbf{x}}_i(t) = \bar{\mathbf{x}}_j(t) = \mathbf{x}_{n+1}(t)$ and $\dot{\bar{\mathbf{x}}}_i(t) = \dot{\bar{\mathbf{x}}}_j(t) = \dot{\mathbf{x}}_{n+1}(t)$ , $\forall i, j \in \{1,2,...,n+1\}$ ,are done in finite time. This shows that the control law (20) can drive the multi-robot system to achieve asymptotical consensus tracking. The following theorem summarizes the aforementioned design procedures and asymptotic consensus analysis.

***Theorem 1****: Given the MIMO second-order nonlinear multi-OMR system model in (8), network structure and assumptions therein, the intelligent consensus formation control law (20) is designed such that* $\bar{\mathbf{x}}_i(t) = \bar{\mathbf{x}}_j(t) = \mathbf{x}_{n+1}(t)$ *and* $\dot{\bar{\mathbf{x}}}_i(t) = \dot{\bar{\mathbf{x}}}_j(t) = \dot{\mathbf{x}}_{n+1}(t)$, *i,j*=1,2,…,*n*+1, *are done asymptotically where* $\bar{\mathbf{x}}_\mathbf{j}(t) = \mathbf{x}_j(t) - \mathbf{f}_j(t)$ *and* $\dot{\bar{\mathbf{x}}}_\mathbf{j}(t) = \dot{\mathbf{x}}_j(t) - \dot{\mathbf{f}}_j(t)$ *, namely that all the n follower OMRs converge to their desired poses and pose rates given by the*

TABLE I. MAJOR PARAMETERS OF THE FOUR HETEROGENEOUS OMRS.

| Parameter | Robot 1 | Robot 2 | Robot 3 | Robot 4 |
|---|---|---|---|---|
| $L$ [m] | 0.6 | 0.3 | 0.6 | 0.23 |
| $m$ [kg] | 40 | 25 | 40 | 25 |
| $J$ [kg-m$^2$] | 0.1 | 20 | 0.1 | 20 |
| R[m] | 0.1016 | 0.1016 | 0.1016 | 0.1016 |

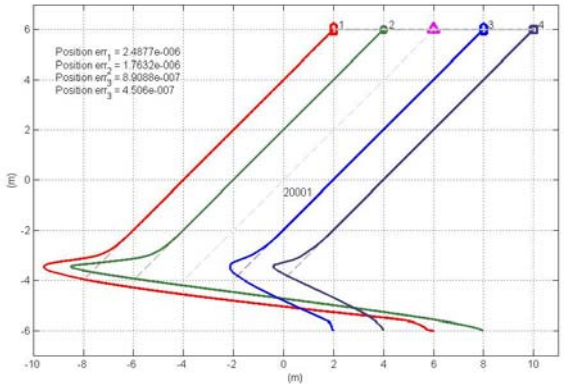



Fig. 5. Simulations results of the intelligent ITSM consensus formation controller where the virtual leader moves from one pose to another and all the followers move in the horizontal line formation.

*virtual leader, and achieve finite-time consensus formation control.*

## IV. Simulations and Discussion

This section aims to conduct three sumulations to show the performance of the proposed control method in ***Theorem 1***. All simulations consider a group of four heterogeneous OMRs and a virtual leader moving at a desired speed. All simulations are conducted using Matlab/Simulink codes where the fourth Range-Kutta numerical method is adopted with a fixed step size of 0.001 seconds.The communication topology is shown in Fig. 1(b), and all the OMRs' major parameters are given in Table 1 for computer simulations. Note that all the static frction coefficients are set as 5 Newtons. Thus, the reduced graph Laplacian $\mathbf{L}+\mathbf{B}$ and $(\mathbf{L}+\mathbf{B})^{-1}$ are computed as follows;

$$\mathbf{L}+\mathbf{B}=\begin{bmatrix}1 & -1 & 0 & 0\\ 0 & 1 & -1 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & -1 & 1\end{bmatrix}, (\mathbf{L}+\mathbf{B})^{-1}=\begin{bmatrix}1 & 1 & 1 & 0\\ 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 1\end{bmatrix}$$

The goal of the first simulation is to verify the feasibility and effectveness of the proposed ITSM formation control law for the four heterogemeous OMRs moving together along a horizontal line, in which the virtual leader moves from one pose to another. In the horizontal line formation, the relative disances between any two neighboring robots are the same and equal to 2 m. In Fig.5, the triangle represents the virtual leader which gets started at the pose of [-4m -4m $0^0$]$^T$ and stops at the desired pose of [0m 6m $0^0$]$^T$. Fig. 5 depicts the trajectories of four heterogeneous OMRs using the proposed consensus formation control method in Theorem 1.

The second simulation is carried out to achieve a rectangular formation where the four OMRs get started at

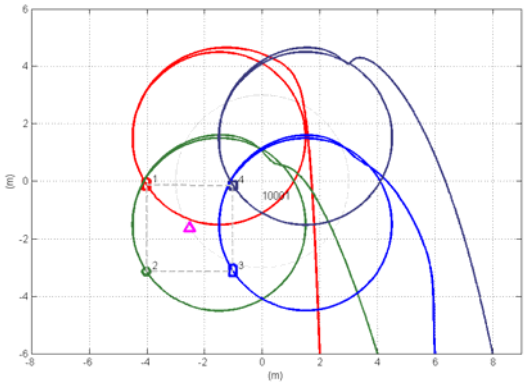



Fig. 6. Simulations results of the proposed intelligent ITSM consensus formation controller where the formation pattern is a rectangle form and the virtual leader moves along an circle trajectory.

TABLE II. Steady-state Errors of the Two Controllers for the Four Heterogeneous Omnidirectional Mobile Multi-Robots.

| Method | Robot 1 | Robot 2 | Robot 3 | Robot 4 |
|---|---|---|---|---|
| ITSMC | 2.4877e-6 | 1.7632e-6 | 8.9088e-7 | 4.508e-7 |
| NTSM[20] | 5.0454e-4 | 5.0089e-4 | 4.7855e-4 | 1.3042e-2 |

their fixed initial positions, the virtual leader moves in a circular trajectory. Fig. 6 shows the simulated trajectories of four OMRs, thereby showing that the proposed controller of each OMR is capable of not only tracking its desired circular trajectory, but also simultaneously maintaining the desired rectangular shape formation among OMRs.

The third simulation is to compare the proposed controller with the adaptive NTSM controller in[20]. The adaptive NTSM controller is applied to achieve the same scenario as in the first simulation.Table II depicts the comparative results, showing that the steady state errors of the adaptive NTSM controller is much larger than those of the proposal method. The results indicate that the proposed method outperforms the adaptive NTSM control method in terms of steady-state errors.

## V. Conclusions and Future Work

This paper has theoretically presented an intelligent integral terminal sliding mode consensus formation control using ORFWNN for a group of uncertain, heterogeneous, networking mobile OMRs, where the networking multi-OMR system is modeled by a directed spanning tree with root being the virtual leader. Based on the uncertain MIMO second-order dynamic model and the communication topology, the intelligent consensus formation control law has been presented based on the concepts of consensus tracking, ORFWNN, ITSM control and graph theory. By sliding mode theory, not only the overall system tracking can be guaranteed, but also the formation keeping can also be ensured. Through the three simulation results, the proposed formation method has been shown effective and powerful in moving the entire heterogeneous OMRs in formation as well as achieving the trajectory tracking. An interesting topic for future research would be to conduct experiments to verify the applicability of the proposed consensus formation control method with collision and obstacle avoidance.

## REFERENCES


[1] R. Murray, "Recent Research in Cooperative Control of Multivehicle Systems," *Journal of Dynamic Syst., Measurement, and Contr.*, vol. 129, pp. 571-583, 2007.

[2] W. Ren et al., "Information consensus in multivehicle cooperative Control," *IEEE Contr. Syst. magazine*, Vol. 27, No. 2, pp. 71-82, 2007.

[3] F. L. Lewis, K. Hengster-Movric, A. Das, and H. Zhang, *Cooperative Control of multi-agent systems*. Springer, 2014.

[4] 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.

[5] 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 Trans. on Syst. Man and Cybernetics- Part B-Cybernetics*, 42(2): pp. 444-457,April 2012.

[6] Z. Meng, W. Ren et al., "Leaderless and Leader-Following Consensus With Communication and Input Delays Under a Directed Network Topology," *IEEE Trans. on Syst., Man, and Cybernetics*, Vol. 41, No. 1, pp. 75-88, 2011.

[7] 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," *Intern. Journal of Fuzzy Syst.*, 2016. DOI 10.1007/s40815-016-0239-0.

[8] 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," *IEEE Trans. on Fuzzy Syst.*, DOI. 10.1109/TFUZZ.2015.2486817, Oct. 2015.

[9] C. C. Tsai, H. L. Wu, and F. C. Tai, "Intelligent Sliding-Mode Formation Control for Uncertain Networked Heterogeneous Mecanum-Wheeled Omnidirectional Platforms," *IEEE International Conference on Systems, Man and Cybernetics*, Budapest, Hungary, October 9-12, 2016.

[10] C. C. Tsai, Y. S. Chen, and F. C. Tai, "Intelligent Adaptive Distributed Consensus Formation Control for Uncertain Networked Heterogeneous Swedish-Wheeled Omnidirectional Multi-Robots," in *Proc. of the SICE 2016*, Tsukuba International Congress Center, Tsukuba, Japan, September 20-23, 2016.

[11] X. F. Wang, C. C. Tsai, and F. C. Tai, "Intelligent Adaptive Backstepping Distributed Consensus Formation Control for Uncertain Networked Four Swedish-Wheeled Omnidirectional Multi- Robots," *Intern. Conf. on Advanced Robotics and Intelligent Systems (ARIS 2016)*, Taipei Nangang Exhibition Center, Taipei, Taiwan, August 31-Sptember 2, 2016.

[12] X. Yu and Z. Man, "Model reference adaptive control systems with terminal sliding modes," *International Journal of Control*, vol. 64, issue. 6, pp. 1165-1176, 1996.

[13] X. Yu and Z. Man, "On Finite Time Convergence Mechanism: Terminal sliding modes," *International Workshop on Variable Structure Systems*, pp.164-168, Tokyo Japan, 1996.

[14] X. H. Yu, M. Zhihong, and Y.Q.Wu: "Terminal sliding modes with fast transient performance," *Proc. Int. Conf. Decision and Control*, 1997, vol. 2, pp. 962–963.

[15] Z. Man and X. Yu, "Terminal sliding mode control of MIMO systems", *in press, IEEE Trans on Circuits and Systems: Part I*, 1997.

[16] F. J. Chang, E. C. Chang, T. J. Liang, and J. F. Chen, "Digital-signal-processor-based DC/AC inverter with integral-compensation terminal sliding-mode control," *IET Power Electronics*, DOI: 10.1049/iet-pel.2010.0071, 2010.

[17] D.W.C. Ho, P.A. Zhang and J. Xu, "Fuzzy wavelet networks for function learning," *IEEE Trans. on Fuzzy Syst.*, vol. 9, no.1, pp.200-211, Feb. 2001.

[18] C. C. Tsai and H. L. Wu, "Nonsingular Terminal Sliding Control Using Fuzzy Wavelet Networks for Mecanum Wheeled Omni-directional Vehicles," *IEEE International Conference on Fuzzy Systems*, Barcelona, Spain, 18-23 July 2010.

[19] C. H. Lee and H. S. Chang, "Output recurrent wavelet neural network-based adaptive backstepping controller for a class of MIMO nonlinear non-affine uncertain systems," *Neural Computing and Applications*, vol. 24, no. 5, pp. 1035-1045, April 2014.

[20] H. L. Wu, C. C. Tsai, and F. C. Tai, "Adaptive Nonsingular Terminal Sliding-Mode Formation Control Using ORFWNN for Uncertain Networked Heterogeneous Mecanum-Wheeled Omnidirectional Robots," *Intern. Conf. on Systems, Man, and Cybernetics (SMC 2017)*, Banff, Canada, Oct.5-8, 2017.


**Hsiao-Lang Wu** received the Diplomat in Electrical Engineering from National Taipei Institute of Technology, Taipei, Taiwan, ROC, the M.S. degrees from the Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, ROC. in 1989 and 2008, respectively. He is currently working toward his Ph.D. degree in the Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan, ROC. His current research interests include mobile robots, nonsingular terminal sliding mode control, fuzzy neural networks, formation control, consensus control and their applications to industrial processes and machines.

**Ching-Chih Tsai** (S'90-M'92-SM'00) 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.

# Development of a Biologically Inspired RBFNN and Its Application to Self-Tuning Motion Control of Autonomous Vehicles

Sendren Sheng-Dong Xu, Hsu-Chih Huang, Tai-Chun Chiu and Shao-Kang Lin

***Abstract*—This paper presents a biologically inspired radial basis function neural network (RBFNN) and its application to self-tuning motion control of autonomous vehicles. A hybrid GA (Genetic Algorithm)-PSO (Particle Swarm Optimization) metaheuristic algorithm is proposed to optimize the RBFNN structure. This GA-PSO RBFNN is then applied to self-tuning online motion control of autonomous vehicles by considering the vehicle kinematics. The parameters of the proposed motion controller are online tuned via the biologically inspired GA-PSO RBFNN. Lyapunov stability theory is employed to prove the system stability. In comparison with traditional approaches, this hybrid metaheuristics takes the benefits of neural network, GA, PSO and online control. Simulation results illustrate the merit and effectiveness of the proposed GA-PSO based RBFNN optimal controller for autonomous vehicles.**

***Index Terms*—GA, PSO, Metaheuristic, neural network, optimization, intelligent control.**

## I. Introduction

Biologically inspired algorithms have attracted considerable attention in both academic and industry and become one of the most popular techniques for dealing with multi-objective optimization problems [1-2]. There are some popular nature-inspired algorithms widely used in various application domains, including GA, PSO, ACO (Ant Colony Optimization) and ABC (Artificial Bee Colony) optimization [1-4]. These population-based algorithms have been proven useful in solving real-world optimization problems [5-8]. They have become efficient tools in solving complex optimization problems from various areas [7-8].

Among these biological algorithms, GA inspired by Charles Darwin's theory of natural evolution is a robust, stochastic search method. Each candidate solution in search space is defined and encoded as a chromosome. This computing paradigm searches for the optimal solution via the selection, crossover and mutation operations. On the other hand, PSO inspired by the social behavior of birds and fishes is one of the swarm intelligences. PSO algorithms have the advantages of self-search experience and global-search experience. GA and PSO are the best popular evolutionary algorithms in solving multidimensional problems. However, both the bio-inspired algorithms have the difficult of premature convergence. These classical population-based algorithms should be modified to improve the global search performance for addressing practical optimization problems [7]. This paper presents a hybrid GA-PSO algorithm to increase the search diversity and overcome the disadvantage. The proposed hybrid GA-PSO algorithm is employed to design an optimal RBFNN.

RBFNN introduced by Broomhead and Lowe is an three layered feed-forward artificial neural network that uses radial basis functions as activation functions [9-10]. This methodology takes the advantages of fast learning capacity and universal approximation. To date, it is a useful neural network architecture to resolve many complex engineering problems [11-12]. However, the RBFNN structure influences the system performance, namely that the selection of centers, widths and output weights for Gaussian functions is an important consideration [13].

Over the years, some methods have been developed for dealing with this RBFNN optimization problem [14-16]. Overall, these conventional methods are computationally expensive and may be trapped at local optimum in solving high-dimensional optimization problems [16]. Since the RBFNN optimization can be regarded as a search problem, the bio-inspired algorithms are useful to design biologically optimal RBFNNs. This study employs the GA-PSO to optimize the RFBNN structure for designing online motion controller of Swedish autonomous vehicles.

Autonomous vehicles equipped with four Swedish omnidirectional wheels have been proven superior to provide greater mobility over conventional vehicles [17-20]. This kind of mobile vehicles is particularly useful in designing robot soccer team, industrial robots, nursing-care robots, and etc [18-20]. Since the four-wheeled vehicles have four degree of freedoms (DOF), there exists a redundancy problem. Vehicle modeling and control of the four-wheeled autonomous vehicles have been addressed by several researches [19-21]. Overall, these studies neither coped with the redundant online RBFNN control, nor developed the GA-PSO evolutionary optimization algorithms.

The objective of this research is to develop an hybrid metaheuristic GA-PSO based RBFNN optimization method, and apply this pragmatic computing to optimal motion control of four-wheeled autonomous vehicles. The control parameters are online tuned by using the proposed GA-PSO RBFNN. The remaining of this paper is organized as follows. In Section II, the hybrid GA-PSO RBFNN is proposed to obtain an optimal network structure. Section III elaborates the application to optimal GA-PSO RBFNN redundant control of four-wheeled autonomous vehicles to achieve trajectory tracking. Section IV conducts simulation results to show the

S. S.-D. Xu and T.C. Chiu are with Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei, Taiwan.

H.C. Huang and S.K. Lin are with the Department of Electrical Engineering, National Ilan University, Yilan, Taiwan.

Corresponding author: Hsu-Chih Huang (hchuang@niu.edu.tw)

performance and merit of the proposed methods. Section V concludes this paper.

## II.GA-PSO BASED RBFNN

### *A. Standard RBFNN*

Fig. 1 depicts the three-layered structure of classical RBFNNs with an input layer, a hidden layer and an output layer [9]. As shown in Fig. 1, the inputs of hidden layer are the linear combinations of scalar weights and the input vector $X = [x_1\ x_2\ ...\ x_n]^T \in R^n$. These vectors are mapped by the radial basis functions in each node. The RBFNN output layer yields a vector $Y = [y_1\ y_2\ ...\ y_p]^T \in R^p$ for $m$ outputs by linear combination of the outputs of the hidden nodes. In summary, the output of the standard RBFNN networks in Fig. 1 is described by

$$y_p = \sum_{j=1}^{m} w_{jp} h_j \tag{1}$$

where $H = [h_1\ h_2\ ...\ h_m]^T$ is the radial basis vector expressed by the following Gaussian function

$$h_j = \exp\left(\frac{-\left\|X - C_j\right\|^2}{2b_j^2}\right),\ j = 1, 2, ..., m \tag{2}$$

where $\|\bullet\|$ is the Euclidean norm, $C_j = [c_{j1}\ c_{j2}\ ...\ c_{jm}]^T$ is the center vector of the $j$th node and $B = [b_1\ b_2\ ...\ b_m]^T$ is the basis width vector. $w_{jp}$ is the connect weight between $j$th basis function and $p$th output node.

Gradient descent training method for RBFNN networks has proven to be more effective than the conventional training approaches [19-21]. These parameters of RBFNN architecture can be trained using the following rules:

$$w_j(k+1) = w_j(k) + \eta e(k) h_j(k) \tag{3}$$

$$C_{ji}(k+1) = C_{ji}(k) + \eta e(k) w_j h_j \frac{x_i(k) - C_{ji}(k)}{b_j^2(k)} \tag{4}$$

$$b_j(k+1) = b_j(k) + \eta e(k) w_j h_j \frac{\left\|X(k) - C_j(k)\right\|^2}{b_j^3(k)} \tag{5}$$

where $\eta$ is the learning rate and $e(k)$ is the error in $k$th sampling. However, this approach suffers from the local optimum problem because the RBFNN structure is not well tuned. It is very important to determine the network structure parameters, thereby improving the RBFNN's performance. This parameter tuning problem can be addressed using the evolutionary algorithms [22]. This paper developed a hybrid GA-PSO algorithm to optimize the RBFNN network structure for solving real-world engineering problems.

### *B. Hybrid GA-PSO*

*(1) Classical GA*

GA developed by John Holland is categorized as a search algorithm based on the mechanics of the natural selection process. The most basic concept "survival of the fittest" is that the strong tend to adapt and survive while the weak tend to die out [5]. This paradigm employs probabilistic rules to evolve a population from one generation to the next via the genetic operators: selection, crossover and mutation.

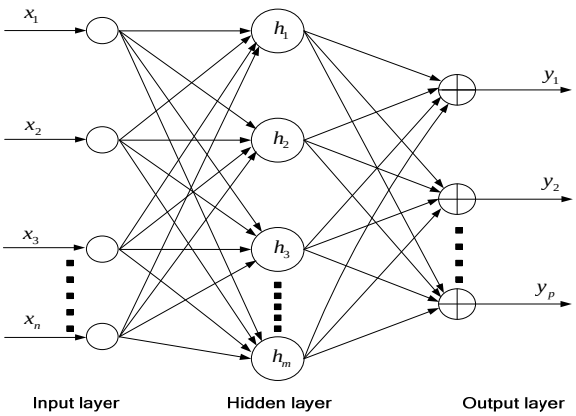



Fig. 1. Structure of the classical RBFNN network [9].

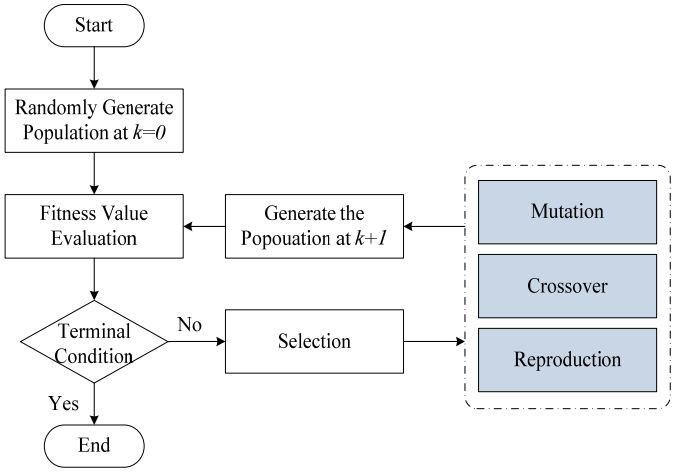



Fig. 2. Flowchart of classical GA.

When applying GA to deal with optimization problems, an initial population of feasible solutions is generated. Each feasible solution is encoded as a chromosome string, called genotype. These chromosomes are evaluated using a predefined fitness function or objective function based on the optimization problems.

Fig. 2 presents the flowchart of GA, the initial population is randomly generated and the fitness function is defined before GA evolutionary process. In the proposed hybrid GA-PSO paradigm, the three key operations are included in the PSO process. This study employs the tournament selection, single -point crossover and single-point mutation strategies to develop the hybrid GA-PSO algorithm.

*(2) Classical PSO*

PSO introduced by Kennedy and Eberhart is also a population-based algorithm used to explore the search space of a given problem to find the settings or parameters required to maximize a particular objective [4]. This swarm intelligence is inspired by social behavior of birds and fishes by combining self-experience with social experience. PSO uses a number of particles that constitute a swarm moving around in the search space looking for the best solution. Each particle in search space adjusts its “flying” according to its own flying experience as well as the flying experience of other particles.

```
For each particle
   Initialize the swarm in the solution space
END

Do
   For each particle
      Calculate fitness value
      If the fitness value is better than its personal best
      set current value as the new pbest
   End

   Choose the particle with the best fitness value of all as gbest
   For each particle
      Calculate particle velocity using (7)
      Update particle position using (6)
   End
While stopping condition (maximum iterations or minimum
error criteria) is not reached
```

Fig. 3. Pseudo code of the classical PSO algorithm.

PSOs can be employed to solve for the optimal problems with the multimodal function $f(x) = f(x_1, x_2, ...x_n)$ by using a population of particles. $x_i(k)$ denote the position of particle *i* in the search space at discrete time step *k*. The position of the particle is changed by adding a velocity, $v_i(k)$ to the current position, given by

$$x_i(k+1) = x_i(k) + v_i(k+1) \tag{6}$$

with an initial position $x_i(0)$. The velocity vector drives the PSO optimization process and reflects both the experimental knowledge of the particle and socially exchanged information from the particle's neighborhood [6].

In PSO algorithm, the best position reached by the single particle (*pbest*) is responsible for the attraction. On the other hand, the best location found by the rest of the swarm (*gbest*) indicates the influence of the swarm on the single particle. Each particle *i* move around the search space, and update its velocity using its past experience and the population's experience, expressed by

$$v_{ij}(k+1) = wv_{ij}(k) + c_1\varphi_1(p_{best} - x_{ij}(k)) + c_2\varphi_2(g_{best} - x_{ij}(k)) \tag{7}$$

where $v_{ij}(k)$ is the velocity of particle *i* in dimension *j* at time step *k*, $x_{ij}(k)$ is the position of particle *i* in dimension *j* at time step *k*, $c_1$ and $c_2$ are positive acceleration constants. $\varphi_1$ and $\varphi_2$ are random numbers. *w* is called inertia. Fig. 3 presents the pseudo code of classical PSO algorithm. The stopping condition is set by the maximum iteration or minimum error in practical applications.

*(3)Hybrid GA-PSO*

Although GAs and PSOs have been widely applied to deal with various optimization problems, these bio-inspired algorithms suffer from the premature convergence. In other words, these traditional computing paradigms may be trapped into local optimum. This subsection aims at developing a hybrid PSO to improve the searching diversity by combining the GA operators, including selection, crossover and mutation. The proposed hybrid GA-PSO metaheuristics is then applied to design an optimal RBFNN for self-tuning motion control on autonomous vehicles.

In the proposed hybrid GA-PSO algorithm, the PSO particles are defined and the evolutionary PSO process is then executed. After the PSO process, the particles are transformed into GA chromosomes and the three GA operations are performed. The local best and global are obtained by evaluating the individuals. Finally, the position and velocity of particles are updated via (7) and (6). The proposed hybrid GA-PSO algorithm can be summarized by the following steps:

**Step 1:** Initialize the PSO swarm size, neighbourhood size, search space, acceleration coefficients, and number of iterations.

**Step 2:** Randomly generate PSO particles and initialize the position and velocity of the particles.

**Step 3:** Calculate the fitness value for all the particles.

**Step 4:** GA selection, crossover and mutation operations are executed after all particles have constructed a solution.

**Step 5:** (1) Search of personal best population.
(2) Search of global best population.

**Step 6:** (1)Update the velocity using (7).
(2)Update the position using (6).

**Step 7:** Check the stopping criterion. If the stopping criterion is not matched, go to Step 3 and set $k = k+1$, otherwise, output the solution and stop the hybrid GA-PSO algorithm.

*C. Optimized RBFNN based on GA-PSO*

In classical RBFNN shown in Fig. 1, there are three parameters that can be adjusted: centers $C_j$ and its width of the hidden layer's basis function $B$, and the connection weights $w_{jp}$. These network structure parameters influence the RBFNN performance. This subsection aims at tuning the RBFNN by means of the proposed hybrid GA-PSO algorithm

When applying the GA-PSO to cope with RBFNN optimization problems, a particle in PSO is defined by the structure parameters $Particle = \{C_j, B, w_{jp}\}$ and the chromosome in GA is defined by $Chromosome = \{C_j, B, w_{jp}\}$. The optimal RBFNN structure $Particle^* = \{C_j^*, B^*, w_{jp}^*\}$ will be evolved via the GA-PSO evolutionary process. The following objective function with $N_T$ sample is applied to evaluate the PSO particles in searching process.

$$F_{rbfnn} = \frac{1}{N_T}\sum_{k=1}^{N_T}\left(y_p^*(k) - y_p(k)\right)^2 \tag{8}$$

where $y_p(k)$ is the output at *k*th sampling. $y_p^*(k)$ is the predicted output.

This proposed hybrid GA-PSO approach overcomes the local optimum problem in conventional RBFNN optimizers by increasing the search diversity. The proposed hybrid GA-PSO based RBFNN is applicable to other multi-dimensional optimization problems by modifying the objective function in (8). In this study, the GA-PSO RBFNN is applied to online motion control of autonomous vehicles.

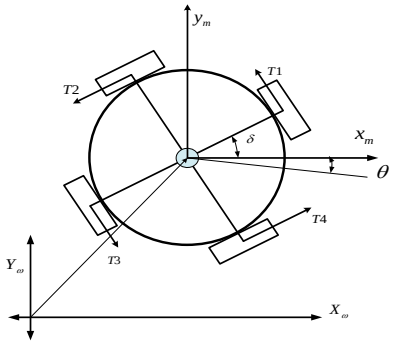



Fig. 4. Geometry of the four-wheeled autonomous vehicle.

## III. Application to Motion Control of Autonomous Vehicles

This section aims at employing the GA-PSO based RBFNN to design an intelligent motion controller for Swedish autonomous vehicles. With the vehicle model, the motion control scheme is directly synthesized via Lyapunov stability theory. The proposed GA-PSO based RBFNN is then applied to design a self-tuning motion controller in which the control parameters are online tuned.

### *A. Vehicle kinematics*

Fig. 4 depicts the geometry of the four-wheeled autonomous vehicle configuration with respect to the world frame. The Swedish omnidirectional wheels are equally equipped at 90 degrees. Due to structural symmetry, the platform has the property that the center of geometry coincides with the center of mass [23]. The kinematic model of the four-wheeled Swedish autonomous vehicle is described by

$$\upsilon(t)=\begin{bmatrix}\upsilon_1(t)\\ \upsilon_2(t)\\ \upsilon_3(t)\\ \upsilon_4(t)\end{bmatrix}=\begin{bmatrix}r\omega_1(t)\\ r\omega_2(t)\\ r\omega_3(t)\\ r\omega_4(t)\end{bmatrix}=P(\theta(t))\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix} \tag{9}$$

where

$$P(\theta(t))=\begin{bmatrix}-\sin(\delta+\theta) & \cos(\delta+\theta) & L\\ -\cos(\delta+\theta) & -\sin(\delta+\theta) & L\\ \sin(\delta+\theta) & -\cos(\delta+\theta) & L\\ \cos(\delta+\theta) & \sin(\delta+\theta) & L\end{bmatrix}$$

$r$ denotes the radius of the Swedish wheel; $\theta$ represents the vehicle orientation; $L$ is the distance from the wheel's center to the center to the geometric center of the mobile vehicle; $\upsilon_i(t)$ and $\omega_i(t), i=1,2,3,4$, respectively denote the linear and angular velocities of each wheel; $[x(t)\ y(t)\ \theta(t)]^T$ represent the pose of the four-wheeled autonomous vehicle. Notice that the matrix $P(\theta(t))$ is singular for any $\theta$ in this redundant robotic system. Its left inverse matrix can be found using $P^{\#}(\theta(t))P(\theta(t))=I$, expressed by

$$P^{\#}(\theta(t))=\begin{bmatrix}\frac{-\sin(\delta+\theta)}{2} & \frac{-\cos(\delta+\theta)}{2} & \frac{\sin(\delta+\theta)}{2} & \frac{\cos(\delta+\theta)}{2}\\ \frac{\cos(\delta+\theta)}{2} & \frac{-\sin(\delta+\theta)}{2} & \frac{-\cos(\delta+\theta)}{2} & \frac{\sin(\delta+\theta)}{2}\\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L}\end{bmatrix}$$

where $I$ is the 3x3 identity matrix.

### *B. Motion control*

Point-to-point stabilization and trajectory tracking control are two important topics in robotics research. This subsection is devoted to designing an unified kinematic controller to achieve stabilization and trajectory tracking for the four-wheeled Swedish vehicle in Fig. 4. The proportional-integral-derivative (PID) control strategy is employed to simultaneous solve the stabilization and tracking problem in four-wheeled autonomous vehicles.

#### *(1) Point-to-point stabilization*

The control goal of the point-to-point stabilization for autonomous vehicles is to find the controlled angular velocity vector $[\omega_1(t)\ \omega_2(t)\ \omega_3(t)\ \omega_4(t)]^T$ to steer the autonomous vehicles from any starting pose $Y_0=[x_0\ \ y_0\ \ \theta_0]^T$ to any desired destination pose $Y_d=[x_d\ \ y_d\ \ \theta_d]^T$. Notice that the current pose of the mobile vehicle is $Y_1(t)=[x(t)\ \ y(t)\ \ \theta(t)]^T$. To design the stabilizer, one defines the following pose error

$$Y_e(t)=\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}=\begin{bmatrix}x(t)\\ y(t)\\ \theta(t)\end{bmatrix}-\begin{bmatrix}x_d\\ y_d\\ \theta_d\end{bmatrix} \tag{10}$$

which gives

$$\dot{Y}_e(t)=\begin{bmatrix}\dot{x}_e(t)\\ \dot{y}_e(t)\\ \dot{\theta}_e(t)\end{bmatrix}=\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix}=P^{\#}(\theta(t))\begin{bmatrix}r\omega_1(t)\\ r\omega_2(t)\\ r\omega_3(t)\\ r\omega_4(t)\end{bmatrix} \tag{11}$$

In order to asymptotically stabilize the control system of mobile vehicle, this study proposes the PID-like control law as follows:

$$\begin{bmatrix}\omega_1(t)\\ \omega_2(t)\\ \omega_3(t)\\ \omega_4(t)\end{bmatrix}=\frac{1}{r}P(\theta(t))\left(-K_p\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}-K_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix}\right) \tag{12}$$

Note that the matrices $K_P$ and $K_I$ are symmetric and positive definite,

i.e., $K_P=diag\{K_{p1},K_{p2},K_{p3}\}=K_P^{\ T}>0$, $K_I=diag\{K_{i1},K_{i2},K_{i3}\}=K_I^{\ T}>0$. The control system proposed in (12) takes the advantages of PID control theory which is widely used in industrial control systems [23-24]. Taking (12) into (11), the dynamics of the closed-loop error system becomes

$$\dot{Y}_e(t)=\begin{bmatrix}\dot{x}_e(t)\\ \dot{y}_e(t)\\ \dot{\theta}_e(t)\end{bmatrix}=-K_p\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}-K_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix} \tag{13}$$

Lyapunov stability theory is the standard tool and one of the most important tools in the analysis of nonlinear systems. It provides a strategy for constructing stabilizing feedback controllers. In order to prove stability of the system, a Lyapunov function candidate is chosen by

$$V_1(t)=\frac{1}{2}\begin{bmatrix}x_e(t) & y_e(t) & \theta_e(t)\end{bmatrix}\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}+\frac{1}{2}\begin{bmatrix}\int_0^t x_e(\tau)d\tau & \int_0^t y_e(\tau)d\tau & \int_0^t \theta_e(\tau)d\tau\end{bmatrix}K_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix} \quad (14)$$

Taking the time derivative of $V_1(t)$, one obtains

$$\dot{V}_1(t)=\begin{bmatrix}x_e(t) & y_e(t) & \theta_e(t)\end{bmatrix}\begin{bmatrix}\dot{x}_e(t)\\ \dot{y}_e(t)\\ \dot{\theta}_e(t)\end{bmatrix}+\begin{bmatrix}\int_0^t x_e(\tau)d\tau & \int_0^t y_e(\tau)d\tau & \int_0^t \theta_e(\tau)d\tau\end{bmatrix}K_I\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix} \quad (15)$$

$$=-\begin{bmatrix}x_e(t) & y_e(t) & \theta_e(t)\end{bmatrix}K_P\begin{bmatrix}x_e\\ y_e\\ \theta_e\end{bmatrix}<0$$

Since $\dot{V}_1$ is negative semidefinite, the stability of the closed-loop system is ensured, namely that $Y_e(t)=\begin{bmatrix}x_e(t) & y_e(t) & \theta_e(t)\end{bmatrix}^T\to\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ as $t\to\infty$.

*(2) Trajectory tracking*

This subsection considers the trajectory tracking problem of the four-wheeled autonomous vehicle. The mobile vehicle is steered to track the desired differentiable trajectory $Y_d(t)=\begin{bmatrix}x_d(t) & y_d(t) & \theta_d(t)\end{bmatrix}^T$. Unlike the point-to-point stabilization, the desired trajectory is a function of time. One defines the following tracking error vector

$$Y_e(t)=\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}=\begin{bmatrix}x(t)\\ y(t)\\ \theta(t)\end{bmatrix}-\begin{bmatrix}x_d(t)\\ y_d(t)\\ \theta_d(t)\end{bmatrix} \quad (16)$$

Thus, one obtains

$$\dot{Y}_e(t)=\begin{bmatrix}\dot{x}_e(t)\\ \dot{y}_e(t)\\ \dot{\theta}_e(t)\end{bmatrix}=\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix}-\begin{bmatrix}\dot{x}_d(t)\\ \dot{y}_d(t)\\ \dot{\theta}_d(t)\end{bmatrix}=P^{\#}(\theta(t))\begin{bmatrix}r\omega_1(t)\\ r\omega_2(t)\\ r\omega_3(t)\\ r\omega_4(t)\end{bmatrix}-\begin{bmatrix}\dot{x}_d(t)\\ \dot{y}_d(t)\\ \dot{\theta}_d(t)\end{bmatrix} \quad (17)$$

Similarly, the control goal of trajectory tracking is to find the motors' angular velocities $\begin{bmatrix}\omega_1(t) & \omega_2(t) & \omega_3(t) & \omega_4(t)\end{bmatrix}^T$ such that the mobile vehicle is able to track the desired time-varying trajectory. In this paper, the trajectory tracking law is proposed as follows:

$$\begin{bmatrix}\omega_1(t)\\ \omega_2(t)\\ \omega_3(t)\\ \omega_4(t)\end{bmatrix}=\frac{1}{r}P(\theta(t))\left(-K_p\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}-K_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix}+\begin{bmatrix}\dot{x}_d(t)\\ \dot{y}_d(t)\\ \dot{\theta}_d(t)\end{bmatrix}\right) \quad (18)$$

where the matrices, $K_P$ and $K_I$, are symmetric and positive definite. Substituting (18) into (17), the closed-loop error system is described by

$$\dot{Y}_e(t)=\begin{bmatrix}\dot{x}_e(t)\\ \dot{y}_e(t)\\ \dot{\theta}_e(t)\end{bmatrix}=-K_p\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}-K_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix} \quad (19)$$

Similar to point-to-point stabilization, the closed-loop error system can be proven asymptotically stable by means of Lyapunov stability theory. The Lyapunov function candidate can be chosen as (14). From (15), the closed-loop error system for trajectory motion control is asymptotically stable. Worthy of mention is that the point-to-point stabilization control law (12) can be included in the tracking control law (18), meaning that the motion control problem can be simultaneously achieved by the unified control law (18). The desired pose can be either the time-varying trajectory or the fixed destination pose.

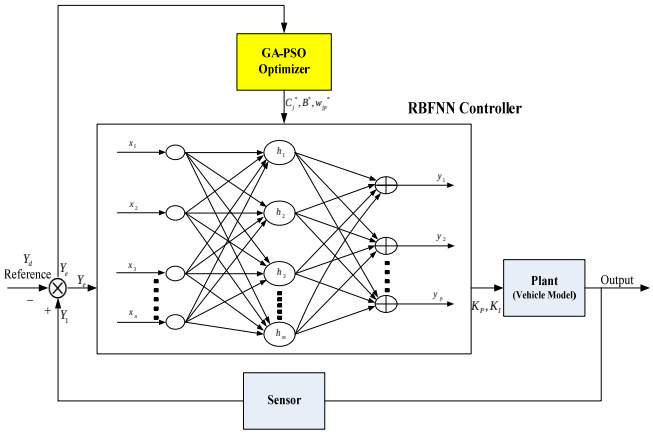



Fig. 5. Intelligent GA-PSO RBFNN motion control scheme.

*C. GA-PSO RBFNN online control*

Although the pseudoinverse unified motion controller of the Swedish autonomous vehicles is synthesized in (18), the two control matrices $K_P$ and $K_I$ are not properly determined. These control parameters influence the system stability and performance. Several researches employed the trial-and-error or offline tuning methods to determine the control parameters [23] in which the control parameters remain unchanged at every sampling.

The online controller outperforms the offline approaches because the parameters are self-tuned at every sampling. Fig. 5 presents the intelligent GA-PSO optimized RBFNN motion control scheme of the four-wheeled autonomous vehicles. The proposed GA-PSO RBFNN is incorporated with the gradient method to online tune the control parameters at every sampling. The control matrices, $K_P$ and $K_I$ are self-updated periodically. The fitness function of the proposed GA-PSO RBFNN is defined by the weighted integral square error (ISE)

$$F=w_f\int_0^t\left(x_e^{\,2}(\tau)+y_e^{\,2}(\tau)+\theta_e^{\,2}(\tau)\right)d\tau \quad (20)$$

where $x_e$ and $y_e$ are the position errors and $\theta_e$ is the error of vehicle's orientation for autonomous vehicles.

## IV. Simulation Results and Discussions

*A. Circular trajectory tracking*

In first simulation, the proposed GA-PSO based RBFNN motion controller aims at tracking the circle trajectory, expressed by $\begin{bmatrix}x_d(t) & y_d(t) & \theta_d(t)\end{bmatrix}^T=\begin{bmatrix}200\cos(\omega_i t)\,\text{cm} & 200\sin(\omega_i t)\,\text{cm} & \pi/4\,\text{rad}\end{bmatrix}^T$, $\omega_i=0.15$ rad/sec. The initial pose is set at $Y_0=\begin{bmatrix}100\text{cm} & 0\text{cm} & \pi/4\text{rad}\end{bmatrix}^T$. Fig. 6 presents the simulation result of circular trajectory tracking and Fig. 7 depicts the tracking errors for the four-wheeled autonomous vehicle. As shown in Figs. 6-7, the proposed optimal self-tuning GA-PSO RBFNN redundant controller (18) successfully steers the vehicle to track this circle trajectory with satisfactory performance.

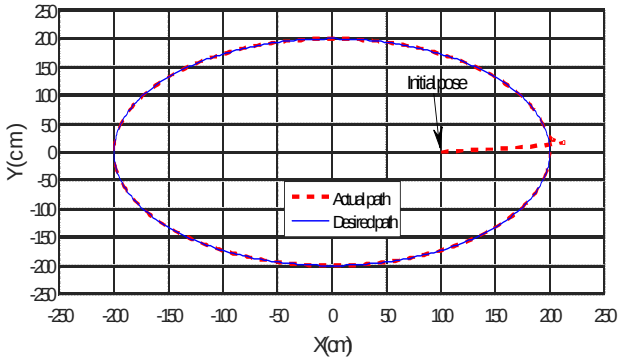



Fig. 6. Simulation result of the circular trajectory tracking.

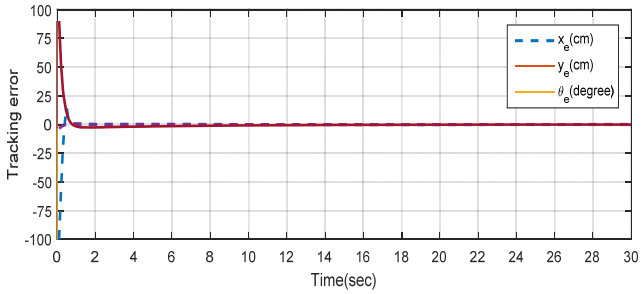



Fig. 7. Tracking errors of the circular trajectory tracking.

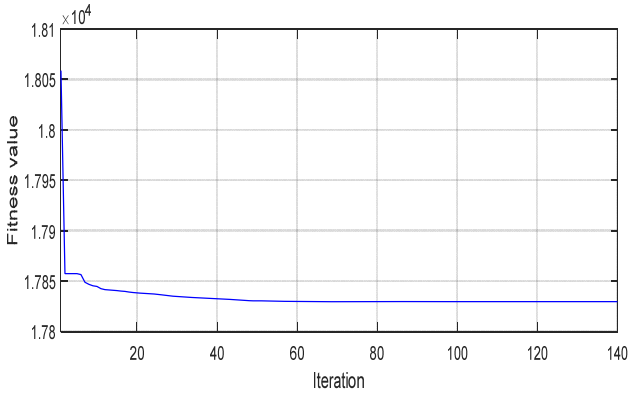



Fig. 8. Fitness value of the GA-PSO based RBFNN in achieving the circular trajectory tracking.

Fig. 8 depicts the evolutions of objective function for the proposed GA-PSO based RBFNN optimal online controller to achieve the circle tracking task. Fig. 8 indicates that the proposed GA-PSO based RBFNN converges to the optimum successfully.

*B. Lemniscate curve trajectory tracking*

This tracking simulation is conducted to investigate the tracking performance of Lemniscate curve trajectory using GA-PSO based RBFNN control scheme. This special trajectory is described by $[x_d(t) \quad y_d(t) \quad \theta_d(t)]^T =$ $[200\cos(\omega_i t)\ \text{cm} \quad 200\sin(2\omega_i t)\ \text{cm} \quad 0\ \text{rad}]^T$, $\omega_i = 0.2$ rad/sec. The initial pose is set at $Y_0 = [250\text{cm} \quad 0\text{cm} \quad 0\text{rad}]^T$. Fig. 9 presents the simulation result of the GA-PSO based RBFNN controller to track this special trajectory.

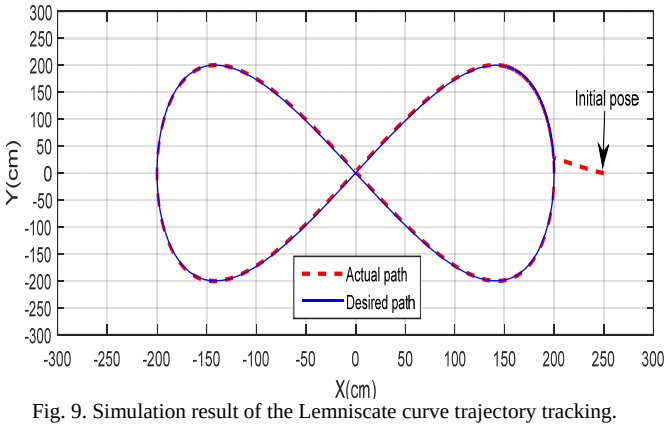



Fig. 9. Simulation result of the Lemniscate curve trajectory tracking.

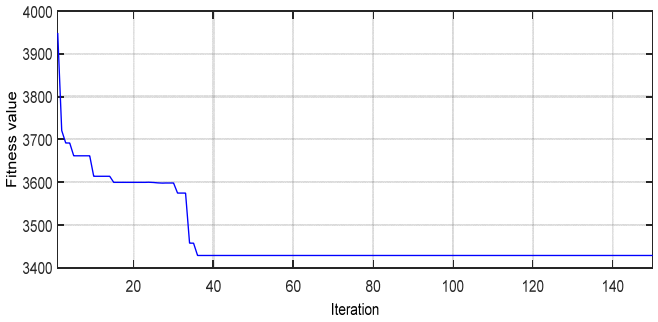



Fig. 10. Fitness value of GA-PSO RBFNN to achieve the Lemniscate curve tracking.

Moreover, Fig. 10 depicts the convergent behavior of the GA-PSO based RBFNN controller to achieve this Lemniscate curve trajectory tracking. These simulation results clearly indicates that the proposed GA-PSO based RBFNN online controller steers the four-wheeled autonomous vehicle to track this trajectory.

Compared with the traditional offline controllers [25-27], the proposed GA-PSO based RBFNN control method online updated the control parameters to obtain better performance. The proposed control approach takes the advantages of GA, PSO, hybrid metaheuristics, RBFNN and online control methodologies.

## V. Conclusion

This paper has presented a GA-PSO based RBFNN and its application to self-tuning motion control of four-wheeled autonomous vehicles. The proposed GA-PSO RBFNN is applied to self-tuning online motion control of autonomous vehicles by considering the kinematic model. The parameters of the proposed motion controller are online tuned via the biologically inspired GA-PSO RBFNN. Through simulation results, the proposed evolutionary RBFNN optimization strategy and self-tuning motion controller have been shown superior to conventional methods. The interesting topic for future work is the implementation of this intelligent robotic system for real-world applications.

## References

[1] Y. J. Gong, M. Shan, J. Zhang, O. Kaynak, W. N. Chen and Z. H. Zhan, "Optimizing RFID network planning by using a particle swarm

optimization algorithm with redundant reader elimination," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 900-912, 2012.
[2] G. Tambouratzis, "Using an ant colony metaheuristic to optimize automatic word segmentation for ancient greek," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 742-753, 2009.
[3] C. H. Hsu and C. F. Jung, "Multi-objective continuous-ant-colony-optimized FC for robot wall-following control," *IEEE Computational Intelligence Magazine*, vol. 8, no. 3, pp. 28-40, 2013.
[4] K. Y. Chan, C. K. Yiu, S. Tharam, S. Nordholm and S. H. Ling, "Enhancement of speech recognitions for control automation using an intelligent particle swarm optimization," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 869-879, 2012.
[5] K. Chen, H. Chen, L. Wang and H. Wu, "Modified real GA for the synthesis of sparse planar circular arrays," *IEEE Antennas Wireless Propagation Letter*, vol. 15, pp. 274-277, 2012.
[6] P. A. S Dayal, G. S. N Raju and S. Mishra S, "Pattern synthesis using accelerated particle swarm optimization, *Modelling, Measurement and Control*, vol. 89, no. 1, pp. 58-76, 2016.
[7] M. Al-Rashidi and M. Hawary, "A survey of particle swarm optimization applications in electric power systems," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 913-918, Apr. 2009.
[8] M. A. Cavuslu, C. Karakuzu and F. Karakaya, "Neural identification of dynamic systems on FPGA with improved PSO learning," *Applied Soft Computing*, vol. 12, no. 9, pp. 2707-2718, 2012.
[9] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adptive networks," *Complex Systems*, vol. 2, pp. 321-355, 1988.
[10] Y. Li, S. Qiang, X. Zhuang, and O. Kaynak, "Robust and adaptive backstepping control for nonlinear systems using RBF neural networks," *IEEE Transactions on Neural Network*, vol. 15, no. 3, pp. 693-701, May 2004.
[11] H. Khosravi, "A novel structure for radial basis function networks-WRBF," *Neural Processing Letters*, vol. 35, pp. 177-186, 2012.
[12] R. J. Schilling, J. J. Carroll Jr., and A. F. Al-Ajlouni, "Approximation of nonliner systems with radial basis function neural networks", *IEEE Transactions on Neural Networks*, vol.12, no.1, pp. 1-15, 2001.
[13] C. Yang, Z. Li, R. Cui and B. Xu, "Neural network-based motion control of an underactuated wheeled inverted pendulum model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp.2004-2016, 2014.
[14] F. Dong, X. Lei and W. Chou, "The adaptive radial basis function neural network for small rotary-wing unmanned aircraft," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 9, pp. 4808-4815, 2014.
[15] P. A. Gutiérrez, C. Hervas-Martinez and F. J. Martínez-Estudillo,"Logistic regression by means of evolutionary radial basis function neural networks," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 246-263, 2011.
[16] G. W. Chang, C. I. Chen and Y. F. Teng,"Radial-basis-function-based neural network for harmonic detection," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 6, pp. 2171-2179, 2010.
[17] C. C. Tsai, Z. R.Wu, Z. C.Wang, and M. F. Hisu, "Adaptive dynamic motion controller design for a four-wheeled omnidirectional mobile robot," *International Conference on System Science and Engineering*, pp. 233-238, 2010.
[18] E. Hashemi, M. G. Jadidi and N. G. Jadidi, "Model-based PI-fuzzy control of four-wheeled omni-directional mobile robots," *Robotics and Autonomous Systems*, vol. 59, pp.930-942, 2011.
[19] J. B. Song and K. S. Byun, "Design and control of a four-wheeled omnidirectional mobile robot with steerable omnidirectional wheels," *Journal of Robotic Systems*, vol. 21, no. 4, pp. 193-208, 2004.
[20] M. Wada and H. Asada, "Design and control of a variable footprint mechanism for holonomic omnidirectional vehicles and its application to wheelchairs," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 6, pp. 978-989, 1999.
[21] R. Siegwart, I. R. Nourbakhsh and D. Scaramuzza, "Introduction to autonomous mobile robots," *2nd Edition, MIT Press*, London, 2011.
[22] H. C. Huang and C. H. Chiang, "An evolutionary radial basis function neural network with robust genetic-based immunecomputing for online tracking control of autonomous robots," *Neural Processing Letters*, vol. 4, no. 1, pp. 19-35, 2016.
[23] H. C. Huang, "Intelligent motion control for four-wheeled holonomic mobile robots using FPGA-based artificial immune system algorithm," *Advances in Mechanical Engineering*, vol. 2013, pp. 1-11, 2013.
[24] L. A. Yusuf and N. Magaji, "GA-PID controller for position control of inverted pendulum," *Proceeding of IEEE 6th International Conference on Adaptive Science & Technology*, pp. 1-5, 2014.
[25] T. Kalmar-Nagy, R. D. Andrea and P. Ganguly,"Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle," *Robotics and Autonomous Systems*, vol. 46, no. 1, pp. 47-64, 2004.
[26] G. Indiveri,"Swedish wheeled omnidirectional mobile robots: kinematics analysis and control,"*IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 164-171, 2009.
[27] D. Chwa, "Sliding-mode tracking control of nonholonomic wheeledmobile robots in polar coordinates," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 637-644, 2004.

**Sendren Sheng-Dong Xu** received his Ph.D. degree in Electrical and Control Engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in February 2009. From 2009 to 2010, he was with NCTU, conducting postdoctoral research. In August 2010, he joined the Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei, Taiwan, where he is currently an Associate Professor. His research interests include intelligent control systems, signal processing, image processing, and embedded systems.

**Hsu-Chih Huang** received the M.S. degree in Institute of Biomedical Engineering from National Cheng-Kung University, Tainan, Taiwan, and Ph.D in Electrical Engineering from National Chung-Hsing University, Taichung, Taiwan, in 1999, and 2009, respectively. He is currently a Professor in the Department of Electrical Engineering, National Ilan University, Yilan, Taiwan. His current research interests include intelligent control, mobile robots, embedded systems, SoPC and nonlinear control.

**Tai-Chun Chiu** is currently working toward his M.S degree in the Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei, Taiwan. His current research interests include intelligent control, signal processing, image processing, and embedded systems.

**Shao-Kang Lin** is currently working toward his M.S degree in the Department of Electrical Engineering, National Ilan University. His current research interests include intelligent control, mobile robots, embedded systems and SoPC.

# A Bilateral Tendon-driven Joint Design and Its Application on an Adult-size Bipedal Robot

Lukman Arif Kurniawan, Yu-Cheng Kuo, Ming-Yu Yan and Chung-Hsien Kuo

***Abstract*—This paper presents a bilateral tendon-driven joint (BTDJ) implementation for fabricating an adult-size bipedal humanoid robot with the considerations of low cost and soft-backlash. The proposed BTDJ is a bio-inspired design to mimic the muscle-tendon activation of the human's joint. The BTDJ is configured with a pair of controlled tendon wires, and the bilateral activations of the tendon wires were used to produce the joint movement. In addition to the joint movement, the benefits of tightened tendon wires may produce the soft-backlash effects so that expensive harmonic drive could not be necessarily used especially for adult-size bipedal humanoid robot. Based on the proposed BTDJs, a 12 degrees of freedom adult-size bipedal humanoid robot prototype with 155 cm height and 37 kg weight was produced in our laboratory for preliminary test. Based on the experiment results, the BTDJ-based bipedal humanoid robot performed small soft-backlash, and the overall bipedal humanoid robot is feasible for operation.**

*Index Terms*—biped humanoid robot; tendon-driven joint; antagonistic control; bio-inspired robot design.

## I. Introduction

A humanoid robot is a robot which its body shape resembling the human body. In general, biped robots have a torso, a head, two arms and two legs. However, some forms of biped robots only models of some part of human body. A biped robot technology has been growth in the last two decades. The humanoid robots that ever been built without using expensive harmonic drives have backlash concerns. The backlash is a crucial problem in biped robot design.

Thus, many researchers have developed new technologies to overcome this problem. One of them is called tendon-wire or tendon-driven for biped robot [1] – [3]. The tendon-driven design is developed to solve a backlash problem as well as to make the biped robot motion more similar to a human.

A tendon driven biped robot is a technology which has several advantages as follows: (1) using inexpensive and not easily damaged style gear for collision to build robot joints; (2) adopting to outdoor environment; (3) without using air pressure resulting from an internal combustion engine or hydraulic system.

Hence, this new technology is bio-inspired in human biomechanics muscle fibers of initiative and antagonists (antagonistic actuation) based on the designs with high-precision antagonist's bilateral tendon-driven joint (BTDJ) modules. A representative example of biped robot using tendon-driven technology with a single motor actuator system had been built by Delft Bio-robotics Laboratory, Delft University of Technology, Netherland [2].

Tendon-driven design for biped robots had been widely discussed [2], [3], [4], [5]. A tendon-driven actuator [6] – [7] can be designed to achieve rotary joint motion by transmitting linear driving force from linear actuator motors to its end effectors with thrust wires [8]. Nonetheless, a large mechanical loss and frictional loss were occurred especially in the connection points of the thrust wires and the end effector because of their complicated structure.

Moreover, a new type of tendon-driven or tendon driven rotary actuator with flexible actuators [9] is needed with two actuators. In that proposed rotary actuator, it is possible to convert linear driving force into torque, and a polyethylene (PE) line is attached to tip points of the thrust wires in order to connect the tip points of thrust wires and its end effector. Hence, simple structure and simple control can be achieved with a PE line, and the problem of backlash can be decreased [10].

Based on the PE line configuration, the linear motors can be freely placed away from the end effector although the connecting point of thrust wires and the PE line should be fixed near the end effector. Therefore, the end effector of the rotary actuator can be miniaturized [10].

Several works performed antagonistic tendon-driven solution for impact force absorption which is attracting attention. In [11], a hand arm system with antagonistically driven finger was proposed. They developed a hand that performed stiffness regulation by using an antagonistically driven finger. That hand had succeeded in absorbing a shock received in just 67 ms. In addition, tendon-drivens also applied for the upper body of Kojiro project [12]. In that work, impact absorption was checked during the impact of ball catch actions, body attack, etc.

Furthermore, artificial muscle driving joint [13],

This work was supported by the National Science Council, Taiwan, R.O.C. under Grants NSC MOST 106 - 2221 - E - 011 - 004 - MY3 and the "Center for Cyber-physical System Innovation" from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

L.A. Kurniawan, Y.C. Kuo, M.Y. Yan and C.H. Kuo are with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, 106 Taiwan (corresponding author e-mail: chkuo@mail.ntust.edu.tw;).

tendon-driven driver joint [14], [15] researches also have been proposed to be used for different parts of the robot. For example, in order to reduce the rigidity of robots, the stiffness of wire tension also discussed in [16]. By utilizing forward kinematics, the position of the robot is realized. Then with the analysis of static, the torque balance relationship between cables is realized.

Based on the importance of tendon-driven design, the main objective of this paper is to design a prototype BTDJ-based biped robot with low cost price for reducing backlash. The BTDJ is desired to move precisely with antagonistically of two tendons-driven methods. To validate the feasibility of the BTDJ, a bipedal humanoid robot prototype was designed with two legs. Each leg has six degrees of freedom. Totally, there are 12 degrees of freedom in all legs.

Finally, this paper is organized as the following. Section II describes the mechanical design of BTDJ; Section III elaborates the BTDJ control system; Section IV presents the fabrication of a BTDJ-based adult-size bipedal robot; Section V provides the experiments and results; Section VI is the conclusion and future works.

## II. Mechanical Design of a BTDJ

This section presents the mechanical design of a BTDJ. The proposed BTDJ is a bio-inspired design to mimic the muscle-tendon activation of the human's joint. The BTDJ is configured with a pair of controlled tendon wires, and the antagonistic activations of the bilateral tightened tendon wires were used to produce the joint movement with small soft-backlash effects.

Practically, the antagonistic bilateral tendon wire control mechanism is desired to mimic the agonist and antagonist (contracted and relaxed) that move two muscle tendons, such as biceps and triceps muscle. When a bicep muscle is contacted and the triceps muscle is relaxed, the arm is lifted up. When bicep muscle is pushed down and triceps muscle is pulled up, the arm is brought down. Figure 1 illustrates the biomechanics arm movement concept.

Although a BTDJ uses two actuators to drive a single joint, it does not require high-precision gear reducer which can reach high accuracy and low soft-backlash because of connected bilateral tightened tendon wires to the common joint.

In order to realize the BTJD, the mechanical CAD models is shown in Figure 2. The design in this research used one pair of electric DC screw motors which met proper tension force and speed requirements. The movement of screw stroke in the motor of each module is an antagonist. First, the cable tension of each screw motor is set in the same level when the pulley moves into the center joint angular position. After the tension of each tendon-driven was established, the screw motor worked in an antagonist way.

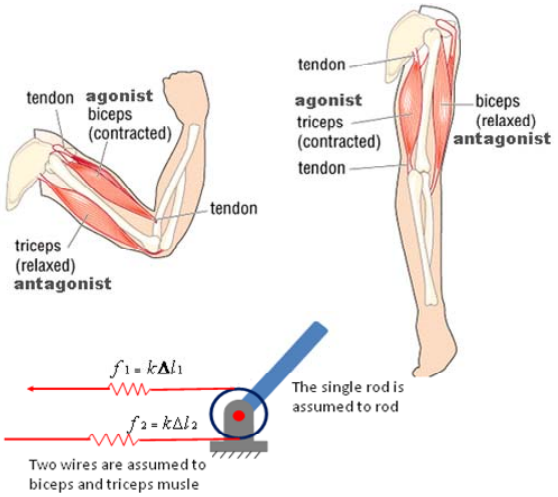



Fig. 1. Illustration of biomechanics arm movement of a BTDJ module.

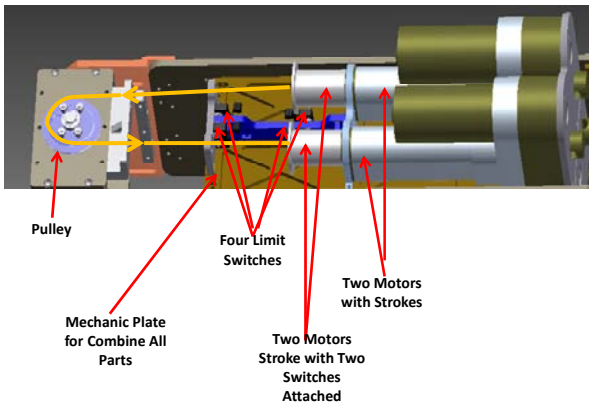



Fig. 2. Mechanical design of a BTDJ module.

During the motor's operation, some unexpected control errors may be happened. In order to prevent this situation, each module is equipped with four limit switches to prevent the over-stroke of the screw motors to ensure safety. The activation of any limit switch will result in the stop of all screw motors in this BTDJ module.

Moreover, two DC motors were equipped with linear screws and potentiometers. The potentiometer is to measure the screw motor position so that the tension force can be obtained in terms of a joint rotary potentiometer. The mechanical plates were made with aluminum material. The ends of two screw motors were connected together with a pulley by using steel wires to form the tendon-driven configuration. The DC screw motors used HIWIN linear actuator (HIWIN-LAS3-1) that could perform 1200 N force.

Then, the pulley is made by calculating the size of the tendon-driven attached on each joint. The diameter of the wires is 1.5 mm. The calculation of the edge of the wire is shown in the Figure 3.

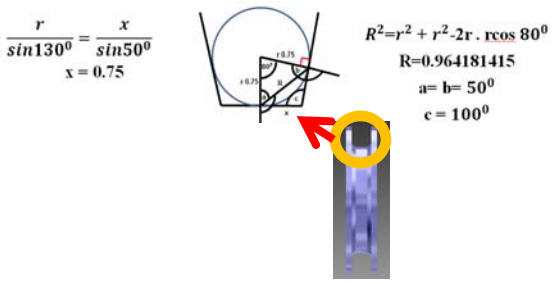



Fig. 3. Calculation of the pulley's edge.

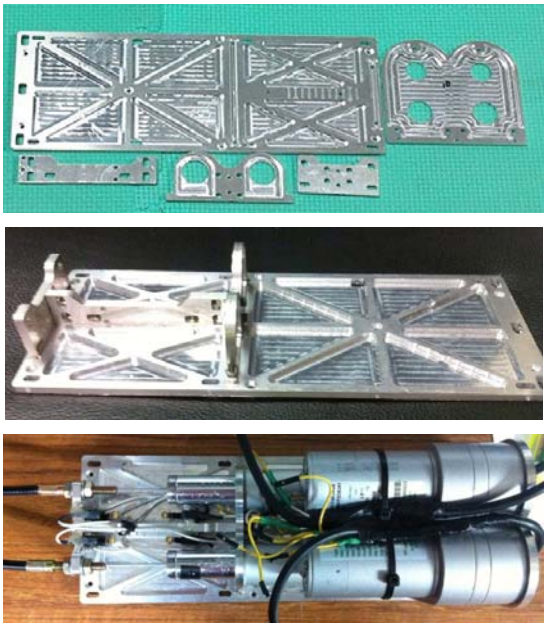


Fig. 4. Produced aluminum plates (up), assembly of BTJD frame and all assembled BTJD picture.

Overall, the component of each module consists of two DC motors HIWIN-LAS3-1, two switches made from plastic material attached on the stroke, four limit switches attached on the mechanical part and two tendon-driven which made from steel with 1.5 mm diameter. The mechanical parts made from aluminum are shown in the Figure 4.

According to the design of Figures 1 – 2, the tendon-driven is capable of controlling the joints of bipedal robot. However, to create an adult size humanoid robot, the problem of stiffness in the wire to get high torque is also a challenge. Hence, a kind of thrust wire was needed for constructing a tendon-driven system, which is to get high achievements to control the wire and the stiffness. The thrust wire should be consisted of inner wire and outer tube. The inner wire slides inside of the outer tube and the driving force can be transmitted from the screw motor to the end effector without too much force transmission loss. As referred to [17], this paper constructed the BTDJ module as shown in Figure 5. Moreover, the trust wire was formed with an inner linear spring and a plastic tube, i.e., spring-plastic tube, to get a better rigidity when a large tension appeared inside the tube, as shown in Figure 6.

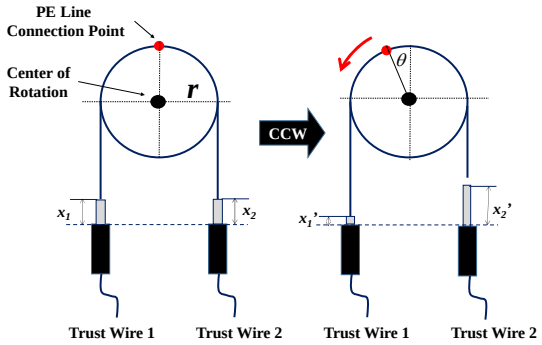



Fig. 5. Modeling of tendon-driven rotary actuator with thrust wires and PE line.

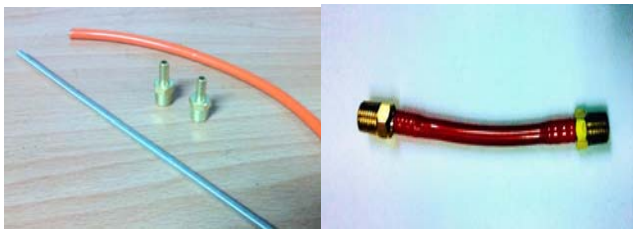


Fig. 6. Fabrication of a spring-plastic formed trust wire.

When thrust wire 1 moves by $x_1$-$x'_1$, the end effector rotates by a specific angle, $\theta$. On the other hand, thrust wire 2 moves by $x_2$-$x'_2$. The radius of the rotation pulley is $r$. Each thrust wire moves toward different directions by the same distance. The end effector is rotated by using the two screw motors. The kinematics of the rotary actuator can be derived as shown in (1) - (2), and defined as $f_i(\theta)$, where $i$ = 1 and 2.

$$f_1(\theta) = x_1 - x_1' = r(\theta) \quad (1)$$

$$f_2(\theta) = x_2 - x_2' = -r(\theta) \quad (2)$$

Jacobian matrix with regard to $\dot{x}$ and $\dot{\theta}$ is derived as eq. (3.7) by differentiating (1) and (2), as indicated in (3)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} r \\ -r \end{bmatrix} \dot{\theta} = J_{aco}\dot{\theta} \quad (3)$$

It is noted that the elasticity of the PE line is negligible because the length of the PE line is short, and it is tightly stretched with tension control.

## III. BTDJ Control System

The section is divided into two parts. The first part is a master board which has AVR AT-Mega 1281 microcontroller to implement the real-time gait pattern generations by inversing kinematics motion patterns. The input of the gait generator is a "gait decision", while the outputs are a set of synchronized motor position commands via universal asynchronous receiver transmitter (UART) communications. After that, the master board sends the gait decision which has the value of angle data to the slaves using the RS - 485 serial communications. Figure 7 shows the block diagram of the master board and its realization picture.

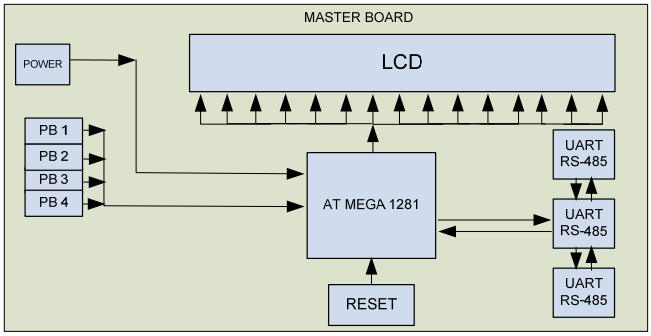



Fig. 7. Block diagram (top) and picture (down) of the master board.

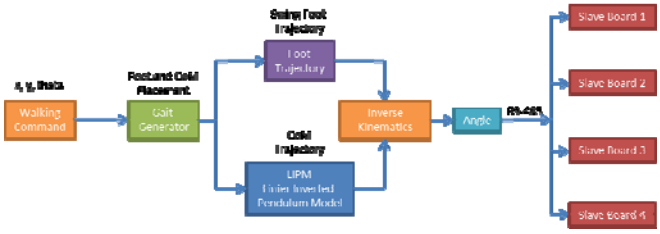
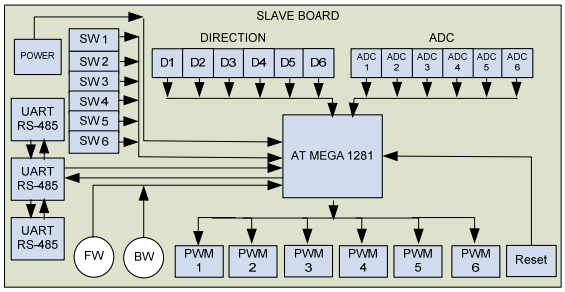



Fig. 8. Control flow chart of the bipedal locomotion in the master board.

Bipedal locomotion is a major function of the master board. Hence, the control of a bipedal robot can be achieved in terms of the flow chart addressed in Figure 8. It is noted that the details of the locomotion could be referred to our previous works [18], including gait generation, foot trajectory, center of mass (CoM) trajectory and inverse kinematics.

The second part is the slave board comprising of four slave boards to control DC-motor's driver and to drive the screw motor in each BTDJ module, as shown in Figure 9. The slave board is realized based on a microcontroller, AT-Mega 1281. This controller is capable of generating six PWM channels, collecting six limit switches' states and collecting six ADC channels. The each PWM channel had a duty cycle digital output pin and a direction pin so that the motor speed and direction can be dealt with. The ADC channel is used to read the voltage value from six motor's potentiometers. The six input pins are logic inputs, and they are used to know the limit switch status.



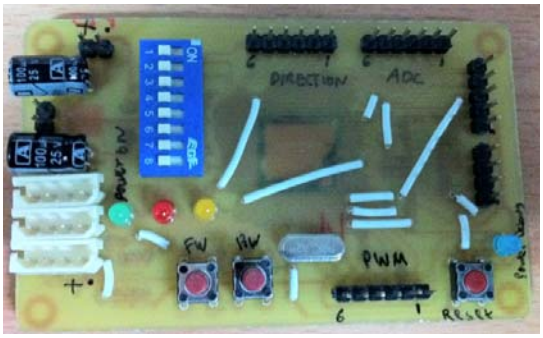



Fig. 9. Block diagram (top) and picture (down) of the slave board.

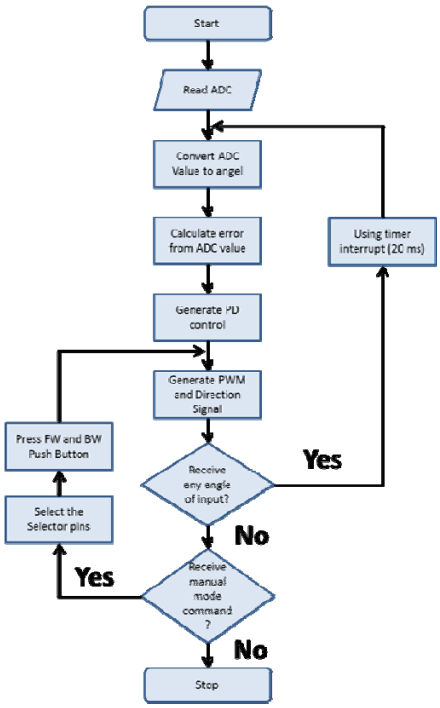



Fig. 10. Flowchart of the software codes implemented in the slave board.

The flowchart of the software design of the slave board is shown in Figure 10. According to Figure 9, the slave board generally works in an automatic mode. It runs the program in a loop. The slave board receives any angle input from the master board with a timer interrupt in 50 Hz. It was also activated in a manual operation mode to drive the linear actuator using selector pins to choose which motor should work and use the push button forward and backward to move it manually. It is noted that the control of DC screw motors used a conventional closed-loop proportional-differential (PD) controller.

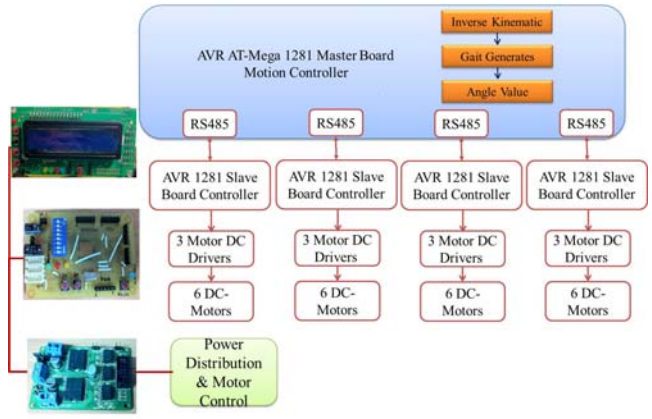



Fig. 11. Overall electric hardware architecture.

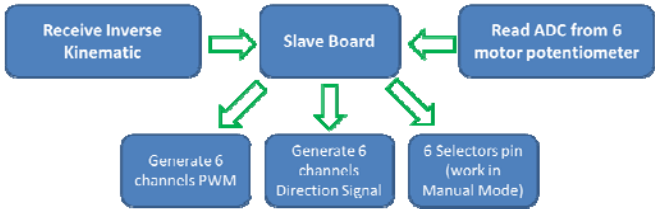



Fig. 12. Slave board electric hardware functions.

Fig. 13. Electric hardware assembly on the body of the robot.

In summary, one master board, four slave boards, twelve DC-motor's drivers and two power regulators which generated voltages 24 volt and 5 volts were employed in this tendon-driven biped robot project. The overall electric hardware architecture is shown in Figure 11. Moreover, Figure 12 explains the slave board architecture. As a consequence, Figure 13 shows the electric hardware assembly on the body of the robot.

## IV. BTDJ-BASED ADULT-SIZE BIPEDAL ROBOT

In this section, the implementation of BTDJ-based adult-size bipedal robot is elaborated. Based on the proposed BTDJs, a 12 degrees of freedom adult-size bipedal humanoid robot prototype was produced in our laboratory for preliminary test, as shown in Figure 14.

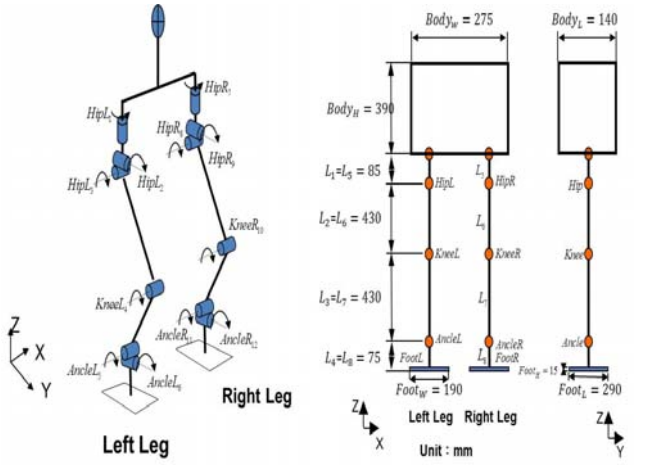



Fig. 14. Mechanical configuration of all the BTDJ joints on a humanoid robot body.

Fig. 15. Pictures of hip (left) and ankle (right) joints.

From the design of Figure 14, two 3 degrees of freedom hip joint, two 2 degrees of freedom ankle joint and two one degree of freedom knee joint were produced. The hip joint and ankle joint are formed with three BTDJs and two BTDJs, respectively, and they are shown in Figure 15.

Based on 12 BTDJs and the controller, an adult-size bipedal humanoid robot with 155 cm height and 37 kg weight was successfully produced, as shown in Figure 16. To make all electric wiring and more concise, all BTDJs controllers were placed inside the limbs. In addition, all the trust wires were arranged to eliminate the friction effects inside the spring-plastic tube in terms of preventing large turns of the trust wires as well as reducing the length of the trust wires.

## V. EXPERIMENTS AND RESULTS

The experiments were done by following four parts:

1. BTDJ PD position control experiments.
2. BTDJ load experiments.
3. Onboard angular position control experiments.
4. Locomotion trajectory experiments.

### *A. BTDJ PD Position Control Experiments*

The position servo control performance of a BTDJ module is important because the BTDJ control must be responsible for proving accurate joint angular position control. As mentioned in Figure 10, a conventional PD controller was desired. The PD control gains were manually tuned by following the approaches in [19]. Figure 17 shows the experiment setup. The experiments were done by applying a 2

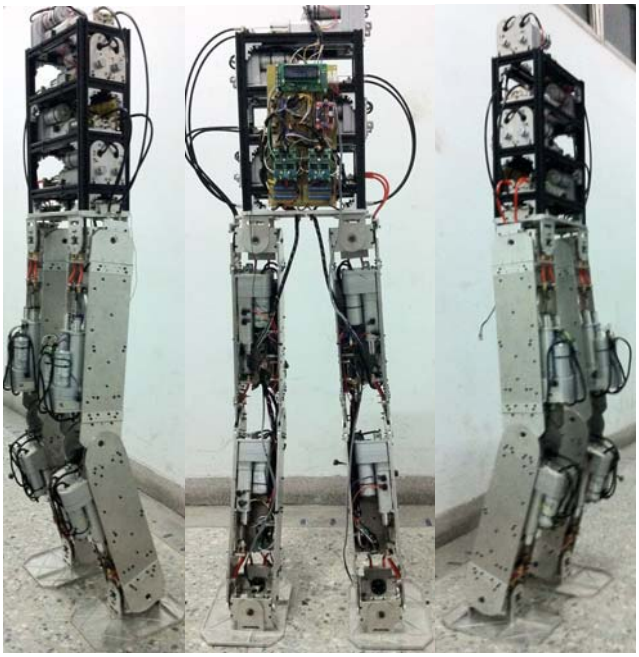


Fig. 16. Pictures of an adult-size BTDJ-based bipedal humanoid robot.

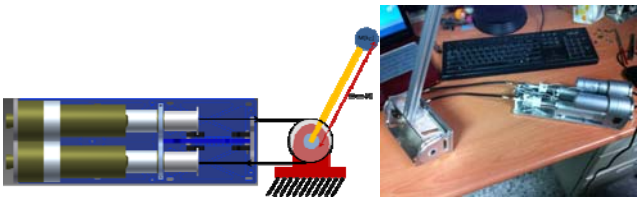


Fig. 17. BTDJ PD position control experiment setup.

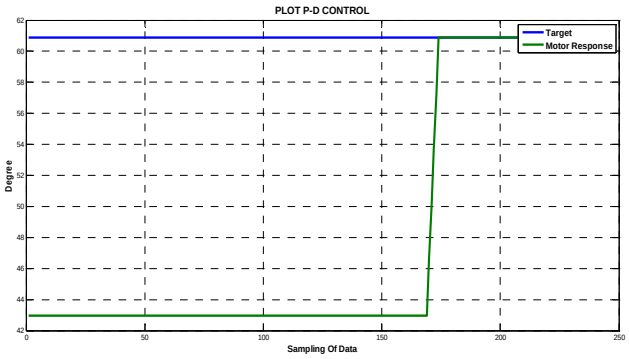



Fig. 18. Response of motor to reach target $K_p = 20$, $K_d = 3.0$.

kgf load at the end effector of a BTDJ module. In addition, a protractor ruler was used for ground truth. The arm length is 50 cm.

Several experiments with different gains' setting were done to reach the target 60.8° from the pulley's starting point position 42.8°. The best response happened when $K_p = 20$, $K_d = 3$. The response is shown in Figure 18, and clearly the response of the BTDJ reaches the target on the 175th sampling time. As a consequence, the PD gain setting of was used for the control of the following load and locomotion experiments.

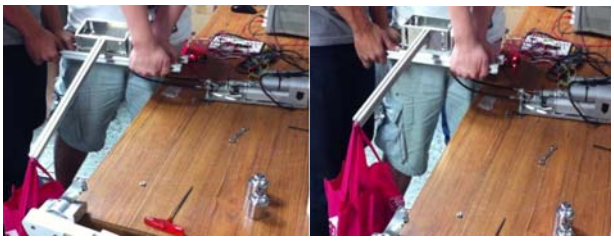


Fig. 19. Successful load test from a start position (left) to a target position (right)

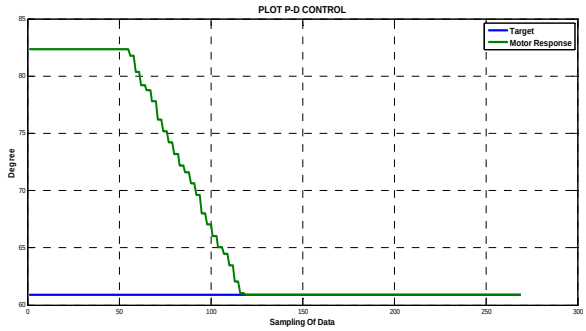



Fig. 20. Response of a load test from a start position (left) to a target position (right).

### *B. BTDJ Load Experiments*

The load experiment is to evaluate the load capability of the BTDJ for supporting the locomotion function of the 37 kg weight adult-size bipedal humanoid robot. From the robot weigh specification, the BTDJ is desired to provide 220 kg-cm torque. By considering the arm length of the experiment setup in Figure 19, the end effector must be capable of carrying 4.4 kg mass. Figure 19 shows the actual experiment situation from a start position to a target position based on $K_p = 20$, $K_d = 3$. Meanwhile, Figure 20 is the response of the angular positions, and the BTDJ reached the target on the 135th sampling time.

### *C. Onboard Angular Position Control Experiments*

The experiments on previous A and B subsections were done with a separate BTDJ experiment platform; however, they were not practical for final bipedal robot evaluation. Hence, onboard angular position control experiments were proposed. The onboard experiments measure the position control performance from the BTDJ modules assembled on the humanoid robot body. Figure 21 shows the ground truth verification of the knee and ankle (roll movement) joints from a start position to a target position. It is noted that the knee joint was desired to move from 14° to 34°; the ankle joint roll movement was desired to move from 90° to 105°. From the investigation of protractor ruler measurement, the knee and ankle can properly reach the desired angles.

### *D. Locomotion Trajectory Experiments*

The last experiments were done to evaluation the trajectory tracking performance with respect to a given locomotion command. The operation of this part follows the flow chart of

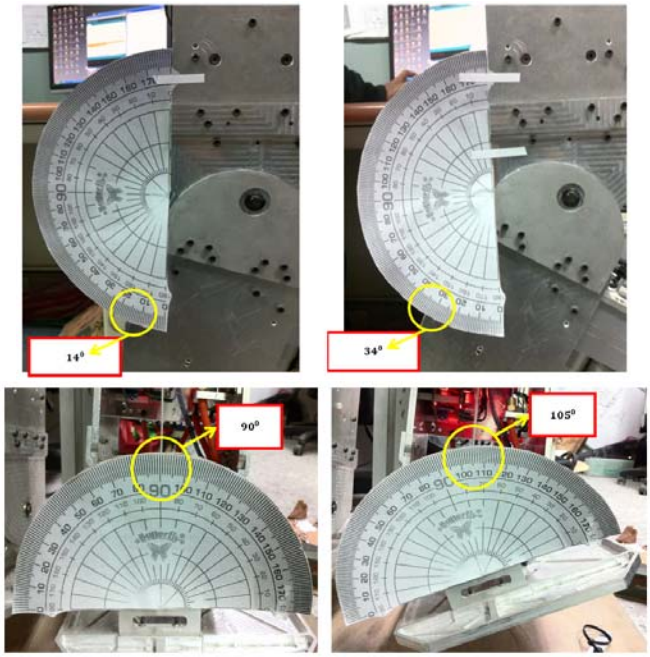



Fig. 21. Ground truth verification of the knee (top) and ankle/ roll movement (down) joints.

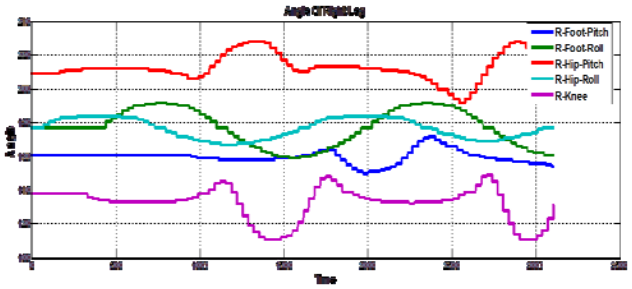



Fig. 22. Trajectory experiment recordings of 5 BTDJs on a bipedal leg based on a specific locomotion cycle command.

Figure 8. The trajectory experiment recordings of 5 BTDJs on a bipedal leg based on a specific locomotion cycle command were addressed in Figure 22. From the experiment results, the 5 BTDJs are capable of performing the desired locomotion commands. Figure 23 shows the snapshots of the hip yaw movement in a 1.5 s locomotion cycle.

## VI. Conclusion and Future Works

This paper presents a prototype implementation for an adult-size biped robot with BTDJ joints. The tendon driven-joints biped robot prototype is made based on antagonistically of two tendons-driven systems. The screw motors on each BTDJ joint in the robot's leg are connected with steel cable which acts as a tendon in the human body. They are controlled by the slave board using PD control in the BTDJ module. Each module also has its identity so that the motors are easier to be controlled, combined and integrated with the master board microcontroller via RS-485 communication. Although the fabrication problem in the linear screw dc-motor caused imperfect pitch movement on the robot's hips, the tendon driven biped robot prototype

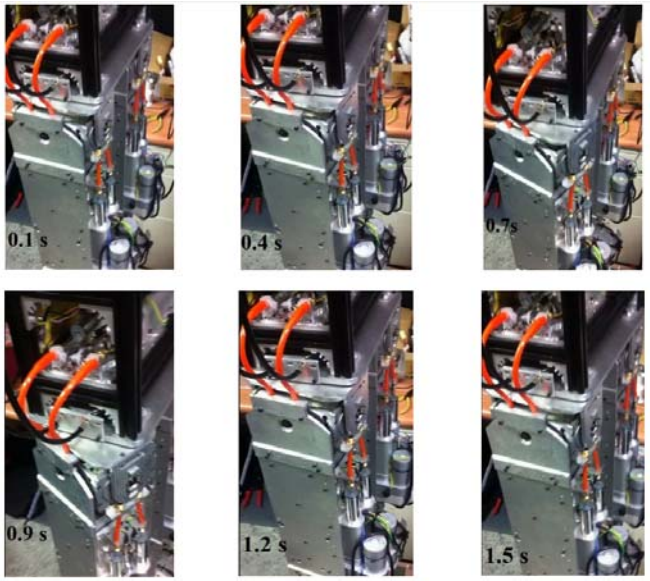



Fig. 23. Snapshots of the hip yaw movement in a 1.5 s locomotion cycle.

experiments showed that this robot has small backlash, move in precise angle and build with low cost components.

For the future works, it is necessary to specifically design the BTDJ for each robot leg joints, including motor specifications and control gains. This is very important because it greatly affects the ability of robot motion in each joint. In addition, lighter actuator with a stronger torque and better controller should be used to get better performances.

## References

[1] I. Mizuuchi, R. Tajima, T. Yoshikai, D. Sato, K. Nagashima, M. Inaba, Y. Kuniyoshi, and H. Inoue, "The design and control of the flexible spine of a fully tendon-driven humanoid Kenta," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol.3, pp. 2527-2532, 2002.

[2] D. Hobbelen, T. de Boer, and M. Wisse, "System overview of bipedal robots flame and tulip: Tailor-made for limit cycle walking," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2486-2491, 2008.

[3] Y. Ito, J. Urata, Y. Nakanishi, K. Okada, and M. Inaba, "Design and development of shock absorbable tendon driven ankles for high-powered humanoids," *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pp. 58-61, 2011.

[4] B. Dynamic. Petman, Available: http://www.bostondynamics.com/robot_bigdog.html, June 2, 2018.

[5] Y. Nakanishi, K. Hongo, J. Urata, I. Mizuuchi, and M. Inaba, "Muscle geometric topology estimation based on muscle length - joint angle nonlinearity in tendon-driven robot systems," *10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 263-268, 2010.

[6] N. Fukaya, S. Toyama, T. Asfour, and R. Dillmann, "Design of the TUAT/Karlsruhe humanoid hand," *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol.3, pp. 1754-1759, 2000.

[7] C. Y. Brown and H. H. Asada, "Inter-finger coordination and postural synergies in robot hands via mechanical implementation of principal components analysis," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2877-2882, 2007.

[8] S. Hyodo, Y. Soeda, and K. Ohnishi, "Verification of flexible actuator from position and force transfer characteristic and its application to bilateral teleoperation system," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 36-42, 2009.

[9] Y. Suzuki, T. Shimono, and K. Ohnishi, "An experimental performance evaluation of a tendinous actuation typed rotary actuator with thrust wires," *Journal of Interpolation and Approximation in Scientific Computing*, vol. 2, pp. 2-49, 2009.
[10] Y. Suzuki, H. Kuwahara, H. Tanaka, and K. Ohnishi, "Development and verification of tendon-driven rotary actuator for haptics with flexible actuators and a PE line," *11th IEEE International Workshop on Advanced Motion Control*, pp. 484-489, 2010.
[11] M. Grebenstein, M. Chalon, G. Hirzinger, and R. Siegwart, "Antagonistically driven finger design for the anthropomorphic dlr hand arm system," *10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 609-616, 2010.
[12] I. Mizuuchi, Y. Nakanishi, Y. Sodeyama, Y. Namiki, T. Nishino, N. Muramatsu, J. Urata, K. Hongo, T. Yoshikai, and M. Inaba, "An advanced musculoskeletal humanoid Kojiro," *7th IEEE-RAS International Conference on Humanoid Robots*, pp. 294-299, 2007.
[13] I. Mizuuchi, T. Yoshikai, Y. Nakanishi, Y. Sodeyama, T. Yamamoto, A. Miyadera, T. Niemela, M. Hayashi, J. Urata, and M. Inaba, "Development of muscle-driven flexible-spine humanoids," *5th IEEE-RAS International Conference on Humanoid Robots*, pp. 339-344, 2005.
[14] Y. Nakanishi, Y. Namiki, J. Urata, I. Mizuuchi, and M. Inaba, "Design of tendon driven humanoid's lower body equipped with redundant and high-powered actuators," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3623-3628, 2007.
[15] D. Nakiri, T. Nakamura, and H. Kino, "Deformation analysis of belt-formed pulley to adjust joint stiffness of tendon robot," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1-6, 2007.
[16] Y. Yangyi, C. Weihai, W. Xingming, and C. Quanzhu, "Stiffness analysis of 3-dof spherical joint based on cable-driven humanoid arm," *The 5th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 99-103, 2010.
[17] Y. Suzuki, K. Sugawara, and K. Ohnishi, "Achievement of precise force control for a tendon-driven rotary actuator with thrust wires and a pe line," *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pp. 1430-1435, 2010.
[18] C.H. Kuo, F. Zal, and S.L. Wu, "Development of fuzzy logic controllers for controlling bipedal robot locomotion on uneven terrains with IMU feedbacks," *Indian Journal of Science and Technology*, vol. 9, no, 28, DOI: 10.17485/ijst/2016/v9i28/98449, 2016.
[19] Y. Li, K.H. Ang, and G.C.Y. Chong, "PID control system analysis and design," *IEEE Control Systems*, vol. 26, no. 1, pp. 32- 41, 2006.

**Lukman Arif Kurniawan** received the master degree in electrical engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2014. He is currently an engineering at the Lite-On Automotive Electronics, Singapore. His research interests include humanoid robots, mechatronics design and mechatronics integration.

**Yu-Cheng Kuo** received the Ph.D. degree in electrical engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2017. He is currently a postdoctoral fellow at the Department of Electrical Engineering, National Taiwan University of Science and Technology. His research interests include humanoid robots, human-machine interface and smart automation.

**Ming-Yu Yan** received the master degree in industrial design from Harbin Institute of Technology, Harbin, China, in 2015. She is currently a research assistant at the Department of Electrical Engineering, National Taiwan University of Science and Technology. Her research interests include human factor design, human-machine interface and humanoid robot design.

**Chung-Hsien Kuo** received the Ph.D. degree in mechanical engineering from National Taiwan University in 1999. Since 2007 he has been with the Department of Electrical Engineering, National Taiwan University of Science and Technology (Taiwan TECH) and is currently a professor of Taiwan TECH. His research interests include humanoid robots, autonomous mobile robots, smart automation, medical instrumentation and brain-computer interface.

# A Sub-Optimal Route Planning for the SCARA Operation on a 3D Object

Wen-June Wang, Sheng-Kai Huang, Tzu-Cheng Kuo, Hsiang-Chieh Chen and Van-Phong Vu

***Abstract*—This paper proposes a sub-optimal route planning for a Selective Compliance Assembly Robot Arm (SCARA) to perform operations on an irregular three-dimension (3D) work-piece. The scenario of this study is that a sub-optimal route is proposed such that the SCARA's end can move to visit a number of (say N) preset holes, possibly in order to insert and tighten a screw in each hole, by following the proposed route on a 3D work-piece. The considered 3D work-piece is built by a number of stacked wooden blocks. To find the sub-optimal route, the following requirements must be satisfied:1. all holes are visited by the SCARA's end at the desired positions respectively; 2. each position is visited only once; 3. the total distance travelled by the robot arm to visit all holes must be as short as possible. In order to satisfy the above three requirements, this paper plans a sub-optimal 3D route based on a solution of the modified Travelling Salesman Problem with particle swarm optimization (PSO) and 2-Opt algorithms. Furthermore, when the SCARA operates on the 3D work-piece, obstacle avoidance must also be guaranteed. Before the operation of the robot arm, a Kinect camera is used to establish a 3D model of the work-piece where N preset holes are also detected. Finally, a completely real experiment will be performed to show that the SCARA's end can visit all preset holes while satisfying the above three requirements and avoiding any obstacle in the operation route.**

***Keywords*—SCARA, Travelling salesman problem, 3D route planning, Visibility graph.**

## I. INTRODUCTION

In order to compete in global markets, the factories of tomorrow will need complete production lines which will include applying many automation technologies and use various kinds of robots [1, 2]. The Selective Compliance Robot Arm (SCARA), which is one of the robots commonly used in modern factories, was invented by Makino [3]. The robot has high positioning accuracy, high repeated positioning accuracy, and good performance of acceleration characteristics. Therefore SCARA has been used in a lot of automated factories for automatic production around the world.

The problem for computing the shortest path between two points in 2D and 3D space has been studied for decades. The visibility graph (V-graph) was invented by Lozano-Perez and Wesley in 1979 [4] and is always used to find the shortest path between two points while avoiding obstacles. To improve the processing speed of the visibility graph, the paper [5] proposed a fast dynamic visibility graph to construct a reduced roadmap of convex polygonal obstacles. The minimum spanning tree method was widely used to find the shortest path. In paper [6], the authors proposed a weighted coding genetic algorithm with degree-constraints to calculate the minimum spanning tree which helps to find a shorter route than those found by several other competing algorithms. The authors in [7] proposed a modified priority-based encoding with a heuristic operator for reducing the possibility of loop-formation in a Particle Swarm Optimization algorithm so that the optimal path was found with good success rates. The paper [8] compared PSO algorithm and Genetic Algorithm (GA) and found that the computational efficiency superiority of the PSO over the GA is statistically proven in some cases. The paper [9] applied the Ant Colony Algorithm with an artificial potential field to solve a 3D path planning problem for a flying robot flying in a complex environment. A special technique for navigation planning in a 3D space for a combination mobile platform and robot arm was proposed by [10] in which the method was derived using the field based navigation techniques applied to the robot end and was applicable in the 3D space for single and multiple obstacles environments.

The traveling salesman problem (TSP) seems like a simple problem if there are not too many cities to be visited. The more cities, the higher the computational complexity becomes. When there are less than 12 cities, [11] claimed that the TSP path found by PSO is the optimal solution. However, if there are more than 12 cities, the path found by PSO is still a sub-optimal solution. The difference between the Generalized TSP (GTSP) and the TSP is that the TSP considers n cities as independent n nodes, but the GTSP considers n nodes as m clusters first. The paper [12] used an ant colony optimization method to change the TSP to a GTSP, and then used a mutation process and a 2-Opt exchange method to prevent the solution becoming locked into a local minimum situation. A hybrid model consisting of a PSO, GA, and mutation process was proposed by [13] to find the optimal solution for the TSP, in which the uniform crossover replaced the one point crossover to avoid invalid tours. A modified hybrid simulated annealing genetic algorithm was proposed in [14] in which the

W.J. Wang, S.K. Huang, T.C. Kuo and V.P. Vu are with the Department of Electrical Engineering, National Central University, Jhong-Li City 32001, Taiwan.

H.C. Chen is with the Department of Electrical Engineering, National United University, Miaoli County 36003, Taiwan.

V.P. Vu is also with the Faculty of Electrical and Electronic Engineering, Ho Chi Minh City University of Technology and Education, Vietnam.

Corresponding author: Wen-June Wang (wjwang@ee.ncu.edu.tw)

algorithm adopts the elite selection operator and places the simulated annealing algorithm in the evolutionary process of a genetic algorithm to optimize its performance and efficiency simultaneously. An approach combining fuzzy logic with ant colony optimization and PSO was provided by [15] for solving the TSP to find the optimal solution and reduce the computation time.

In the above papers, there have been many authors studying the problems of shortest path planning in a 2D plane, optimal solution of the TSP, and 3D path planning for a flying robot. However, it is hard to find a paper which studies route planning for SCARA operation on a 3D object. Route planning for a SCARA to operate on a 3D object such that minimization of the total distance covered by the SCARA's end and obstacle avoidance are achieved simultaneously is the main purpose of this study. Here, the 3D object to be considered is a three-dimensional (3D) set of stacked wooden blocks and the operation is visiting in preset holes on the 3D object.

The organization of this study is as follows. Section II describes the main purpose of the paper and the 3D environment to be worked on. Section III presents the main techniques to solve the problem and summarizes the whole process as a route planning procedure. The real experiment of a SCARA operation on a 3D work-piece is presented in Section IV to show the effectiveness of the proposed method. Lastly, the conclusion and discussion are presented in the final section.

## II. Problem Description

In industrial factories, automated production lines are a trend today's world. In order to reduce labor costs, the use of robots is getting much more popular. In particular, the use of robot arms is highly necessary for high precision or/and high repetition operations. Therefore, the industrial applications of robotic arms are very extensive, and include operations such as drilling, dispensing, and screwing. The robot considered in this study is the Selective Compliance Assembly Robot Arm (SCARA). The SCARA has advantages such as high accuracy and fast speed. It is well suited to the work of multi-position operations. The main work of the SCARA in automated factories is performing operations on a preset path on a two-dimensional (2D) plane. However, this paper investigates the problem of three-dimensional (3D) route planning in a 3D environment while satisfying the following conditions: The end of the SCARA must visit all of the preset positions only once and then return to its starting point having covered the shortest possible distance. In other words, it will be a sub-optimal route planning problem for a SCARA operation on a 3D object. In order to narrate the design process more easily, we use the SCARA to visit all the preset screw holes on a 3D object as an example. The considered 3D object here is a set of stacked wooden blocks. It should be mentioned that before the operation of the robot arm, a Kinect camera is used to establish a 3D model of the stacked wooden blocks and N preset holes are detected on the 3D model. Finally, there is a real experiment to show the result is workable. This study takes into account the fact that the height of each of the detected holes may be different, and any two neighboring holes may be separated by obstacles. Considering the problems mentioned here, this study actually solves a modified 3D Travelling Salesman Problem with obstacle avoidance.

## III. The Proposed Method

### 3.1 Basic concepts

After the use of the Kinect camera, the 3D model of the stacked wooden blocks is built and the positions of the N preset holes are detected. We intend to find a sub-optimal route for the SCARA to visit all of the holes while not only moving the shortest possible distance but also avoiding hitting any obstacles. To achieve this goal, we will complete the task in three steps. The first uses a visibility graph [4], [16] to find the shortest path between any two points on the 2D plane and satisfy the obstacle avoidance. The second step extends the visibility graph to a 3D environment and finds the minimum cost (the shortest distance) between any two points. The final step solves the 3D Travelling Salesman Problem using certain optimization algorithms.

Before using the visibility graph, let us define the "visibility". For any two points, say A and B, if point A can be connected to point B using a straight line without being blocked by an obstacle we call these two points visible to each other. Otherwise, the two points are invisible to each other.

### 3.2 Finding the shortest path on a 2D plane

Suppose there are two points on a 2D flat plane, let us take an aerial view of the plane. If there is an obstacle lying between the two points, these two points are not visible to each other. Then let the corners of the obstacle and the two points be regarded as nodes on the 2D plane, and let any two visible nodes be connected to each other so that a path network is formed as in Fig. 1. Suppose one of the two points is called the starting point and the other is the goal. Based on Dijkstra's algorithm [17], the shortest path from the starting point to the goal without hitting any obstacles can be found. For the sake of safety, in general, if we dilate the contour of each obstacle slightly, the proposed path from the starting point to the goal can avoid obstacles more safely (see Fig. 2).

### 3.3 Finding the shortest path on a 3D object

The above visibility graph and Dijkstra's algorithm are used to find the optimal path on a 2D plane. However, this paper considers the SCARA operating on a 3D object (a set of stacked wooden blocks). There are a number of holes distributed on the 3D object and those holes may be located at different heights. Now, the above visibility graph and Dijkstra's algorithm need to be applied to the 3D

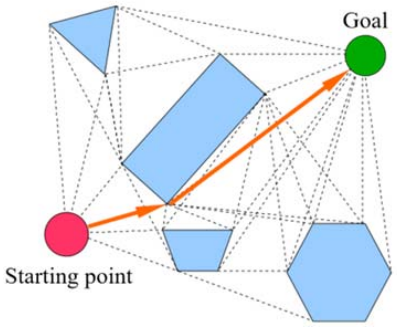



Fig. 1. The path network with visibility

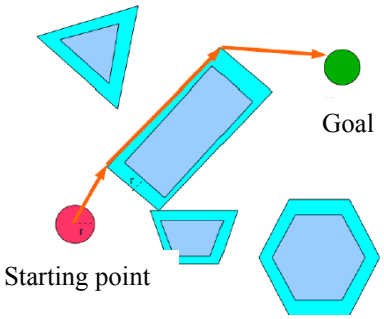



Fig. 2. Obstacle contour dilation

environment. First, let us consider two holes with different heights (a lower and a higher holes), one hole is the starting point and the other is the goal. Suppose the higher hole is the goal and is located on the top of a block stack and the lower hole is the starting point and is located on the lowest level (see Fig. 3). Now, we need to find the shortest distance between the two holes even obstacles may lie between them. Then, in order to guarantee the SCARA's end does not hit the block stack before it arrives at the goal, we need to add an additional node (called E point) that is at the same height as the goal, but positioned closer to the starting hole and little away from the block stack (see Fig. 4). Next, we check the visibility from the lower hole to E point. If it is visible, there is no obstacle on the straight line connecting the two holes. Otherwise, there must be an obstacle on the straight line. In the former case, the SCARA's end can be moved in a direct line from the lower hole to E point. In the latter case, we will make a 2D profile at the height of the lower point as shown in Fig. 3. The corners of the obstacle on the profile, E point, and the lower hole will be the nodes on the 2D profile. Based on the visibility graph and Dijkstra's algorithm, the shortest path on the 2D profile from the lower hole to E point without hitting any obstacles can be found (such as shown in Fig. 2). Suppose that the position coordinates of the three nodes on the shortest path are ($x_s$, $y_s$) (starting hole's coordinates), ($x_2$, $y_2$) (corner's coordinates), and ($x_E$, $y_E$) (additional node's coordinates), respectively, on the 2D profile. It is noted that, here, we are only finding the shortest path on the 2D profile, it will be modified for the 3D environment.

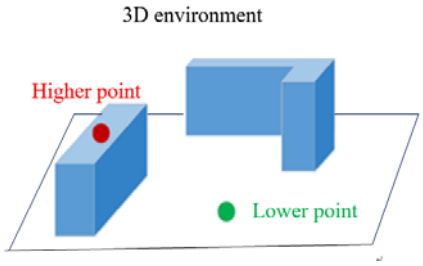



Fig. 3. Two points on the 3D environment

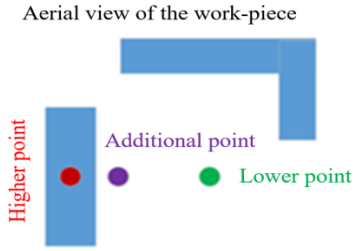



Fig. 4. Additional point away from the higher hole

However, in the 3D environment, E point, the obstacle corners and the lower hole may be at different heights. Now, we will choose the route, whose projection is the shortest path on the 2D profile, as the shortest route on the 3D environment. Then on the route on the 3D environment, the corner's height should be calculated based on the interpolation method as (1).

$$h_c = \frac{\ell_1}{\ell_1 + \ell_2} h_E \tag{1}$$

where $h_c$ is the height of the corner of the obstacle which interrupts the route, $h_E$ is the height of E point, $\ell_1$ and $\ell_2$ are the distances between the lower hole and the corner and between the corner and E point, respectively, in the shortest route in the 3D environment. Therefore, the desired route for the SCARA's end from the lower hole (starting point) to E point without hitting any obstacles will go through ($x_s$, $y_s$, $z_s$) (starting point's coordinates), ($x_2$, $y_2$, $z_s+h_c$) (corner's coordinates), and ($x_E$, $y_E$, $z_s+h_E$) (E point's coordinates) in sequence.

For the sake of convenience, let the above route be called "route A". However, there may be another suitable route between the lower hole and E point. That is, the SCARA's end rises from the lower hole to the height $z_h$ and then, moving at height $z_h$, crosses over the obstacle and arrives at E point. In other words, the SCARA's end will go through ($x_s$, $y_s$, $z_s$), ($x_s$, $y_s$, $z_h$), and ($x_E$, $y_E$, $z_s+h_E$) in sequence, where

$$z_h = \begin{cases} z_s + h_E, & \text{if the obstacle is not higher than E point.} \\ z_s + h_o + \in, & \text{otherwise.} \end{cases} \tag{2}$$

$h_o$ is the height of the obstacle, and $\in$ is a small value. This route is called "route B". Now we compare route A and route B and choose the shorter one to be the desired route ~~one~~. After the SCARA's end arrives at E point, it will then move at the same height to the goal. Thus the shortest route from the lower hole to the goal without hitting obstacles is completed. If the above process is repeated for each pair of holes in the 3D environment, we can find the shortest distance between each pair of holes without hitting any obstacles. For instance, if $N$ is the number of all holes to be visited on the 3D object, there will be $C_2^N$ shortest routes between all pair of holes, such as $N$=10, then there are in total 45 shortest routes connecting all of the pairs of holes. It is noted that if the starting hole is higher than the goal, the E point of the higher point (starting hole) is still necessary and the visibility still needs to be checked. Then the process will be the inverse of the above process from the lower hole to the goal.

### 3.4 Traveling Salesman Problem

In subsection 3.3, we have found the shortest distance between each pair of holes. Now, finding the shortest total distance from the starting hole to the goal and satisfying the conditions mentioned in Section II is the next task. It is obvious that the problem to be solved is a 3D Traveling Salesman Problem (TSP). The TSP is a famous problem which has been studied for decades. The TSP is defined as follows. "Given the distance between each pair of cities, solve the shortest loop that visits each city once and returns to the starting city." Let us define the shortest route of each pair of holes to be the "cost" of the pair of holes. Then the problem to be solved in this study becomes a TSP on a 3D environment. In other words, we must find the minimum total cost (distance) for the SCARA's end such that all preset screw holes are visited only once, and the end finally returns to the starting hole. Of course, the route also needs to avoid any obstacles.

In order to solve this TSP, we use two phases to complete the task. In Phase 1: we used Particle Swarm Optimization (PSO) to find the sub-optimal solution. In Phase 2: to avoid the solution in Phase 1 being the local optimal solution, the 2-Optimal exchange method (2-Opt) [18] is added after the solution of the PSO is found until the solution with the minimum cost is achieved. The detailed process is described as follows.

In Phase 1: Let each particle of the PSO be a vector with $N$ dimension, where $N$ is the number of preset screw holes. In a vector, the value of each entry is the weight of the hole. Each entry's weight of the vector denotes the priority of the hole to be visited. The bigger the weight, the earlier the hole should be visited. After the PSO algorithm, we obtain the final vector which is the sub-optimal solution of the PSO. The fitness function of the particle is the total cost (distance) of all holes being visited. The algorithm of the PSO is stated below

$$\begin{bmatrix} a=0.5 \\ b=0.8 \\ c=0.2 \\ d=0.7 \\ e=0.1 \end{bmatrix} \overset{PSO}{\Rightarrow} \begin{bmatrix} a=0.9 \\ b=0.2 \\ c=0.7 \\ d=0.8 \\ e=0.4 \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ d \\ c \\ e \\ b \end{bmatrix}$$

Fig. 5. The order evolution of the five holes

$$\{a \to d \to c \to e \to b \to a\} \to \{a \to d \to b \to e \to c \to a\}$$

Fig. 6. The 2-opt for five holes with terminals $c$ and $b$.

$$V_i(t+1) = \delta \bullet V_i(t) + c_1 \bullet \varphi \bullet (P_i(t) - X_i(t)) + c_2 \bullet \varphi \bullet (P_g(t) - X_i(t)) \quad (3)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (4)$$

and the fitness function for the particle $X_i(t)$ is defined as

$$F_i(t) = \frac{1}{[\sum_{j=1}^{N} q_{ij}(t)]} \quad (5)$$

where $V_i(t)$ is the velocity of the particle $X_i$ at time $t$, $P_i(t)$ is the best position of $X_i$ till the present time, φ is a random value between 0 and 1, $c_1$ and $c_2$ are learning rates, $P_g(t)$ is the best position of all particles till the present time. Furthermore, $q_{ij}(t)$ is the cost from the order $j^{th}$ city (hole) to the order $(j+1)^{th}$ city (hole) in the particle $X_i(t)$, and $q_{iN}(t)$ is the cost from the last city (hole) to the starting city (hole).

For instance, we have a vector with 5 holes [*a, b, c, d, e*] to be visited and each hole has an initial weight created arbitrarily as shown in the left of Fig. 5. After PSO, the final best vector is shown in the middle of Fig. 5. According to the weights in the middle, the order of visitation is shown in the right of Fig. 5.

In Phase 2: After Phase 1, we obtained the final solution of the TSP. However, we are not sure whether it is a local optimal solution or not. Therefore the 2-Optimal exchange method [18] is added to give one more chance to find a better solution. For instance, the final hole visitation order solution in Fig. 5 is $\{a \to d \to c \to e \to b \to a\}$. Then we add the 2-Opt exchange method to modify the solution. Let us select two holes $c$ and $b$ randomly as two terminals of a part of a sequence, then exchange the holes' order between the two terminals such that $\{c \to e \to b\}$ is exchanged to $\{b \to e \to c\}$, so that another solution is created as $\{a \to d \to b \to e \to c \to a\}$. The sequence after the exchange is shown in Fig. 6.

Then let us compare the new sequence and old sequence and choose the shorter one. For instance, if we have 40 cities to be visited the final route of the TSP after 295 iterations of PSO is shown in Fig. 7a, where the total cost is 242.8923. When the 2-Opt exchange method is added after PSO, the final route is shown in Fig. 7b, where the total cost is 140.8574 [19]. It is seen that the result from PSO+2-Opt is better than that from PSO alone.

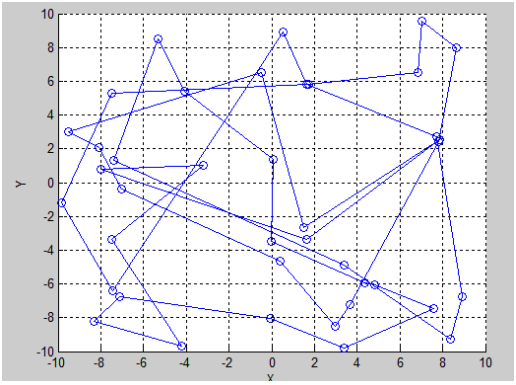


Fig. 7a. The route obtained by PSO only.

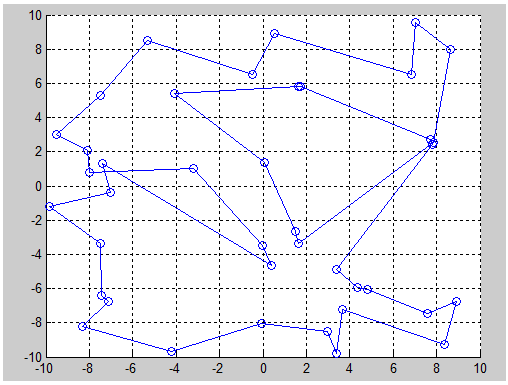


Fig. 7b. The route obtained by PSO+2-opt.

Consequently, we use PSO in Phase 1 to find the sub-optimal solution for *N* screw holes to be visited by the SCARA's end. Then the 2-Opt exchange method is used in Phase 2 to compare the solution in Phase 1 until the solution of the minimum fitness function is found. Finally, the final solution (route) after these two phases will be followed by the SCARA's end to operate on the 3D work-piece.

### 3.5 Overall algorithm for the route planning

Here, we will summarize the proposed sub-optimal route plan as follows.

Step 1: Using a Kinect camera to detect all present screw holes on the 3D object.

Step 2: Using a visibility graph and Dijkstra's algorithm to find the shortest distance for each pair of holes on the 2D profile.

Step 3: Using the interpolation method (1) to find the height of the obstacle corner and then finding the shortest obstacle free route between two holes in the 3D environment.

Step 4: Comparing the distances between the route A obtained from Step 3 and the route B, and choosing the shorter one. Then doing the same job for all pairs of holes.

Step 5: Setting the calculated distance in Step 4 as the cost for each pair of holes. Then using Phase 1 to find the sub-optimal route (or the sub-optimal order of all holes to be visited) for the TSP.

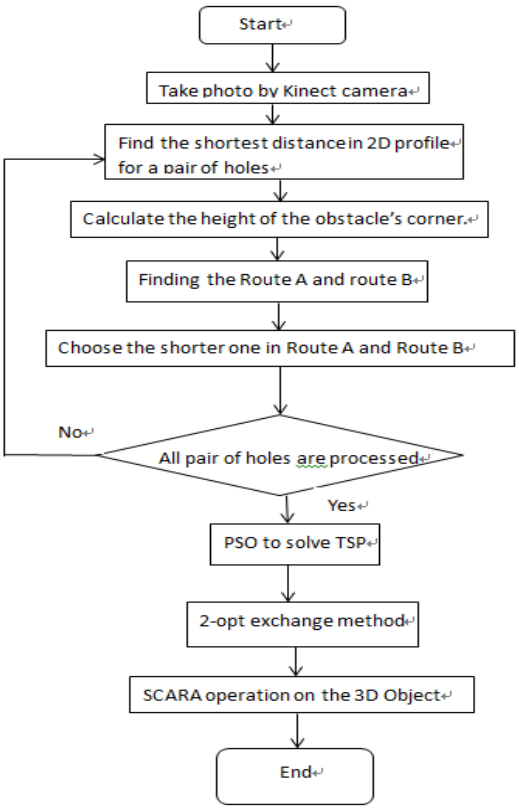



Fig. 8. Flow-chart of the route planning.

Step 6: Using Phase 2 to check the solution in Phase 1 to get the final sub-optimal solution.

Step 7: Applying the sub-optimal route to SCARA to operate on the 3D work-piece.

The flow-chart for the above route planning is shown as Fig. 8.

## IV. Experiment

For this experiment, a SCARA is working on a 1.5m 1.5m platform and the Kinect is set up 70 cm above the platform as shown in Fig. 9. The work-piece model in this experiment has 10 holes at various heights as shown in Fig. 10. After a certain image process, the positions of all the holes are detected. Before the real SCARA operation, we used Matlab to compute and simulate the proposed algorithm, and then plot the route by 3D Graphical User Interface (GUI) as shown in Fig. 11, where the blue line is the proposed route and the blue boxes are the holes in an aerial view. It is noted that there are many colors in Fig. 11 and Fig. 12. The different colors represent different heights as follows; deep blu When there are less than 12 cities e denotes the bottom level, light blue is the first level, light green is the second level, yellow is the third level, orange is the fourth level and brown is the top level with the heights 13, 26, 39, 52, and 67 mm respectively. Fig. 12 shows the route again but from a different angle. Finally, the SCARA's end visits all preset holes on the 3D work-pieces. The video of the SCARA in operation with the proposed method can be seen on Youtube [20].

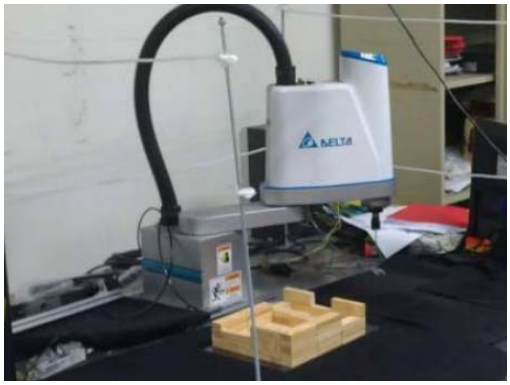


Fig. 9. Experiment environment

Fig. 10. Work-piece and 10 screw holes

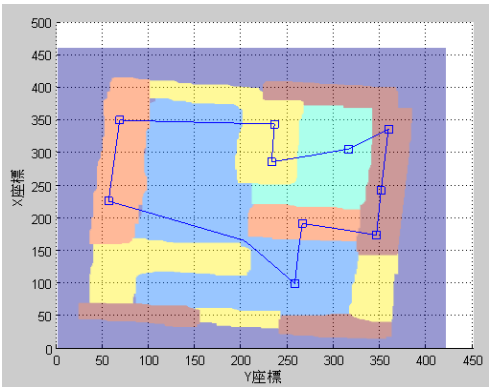



Fig. 11. The proposed sub-optimal route by aerial view.

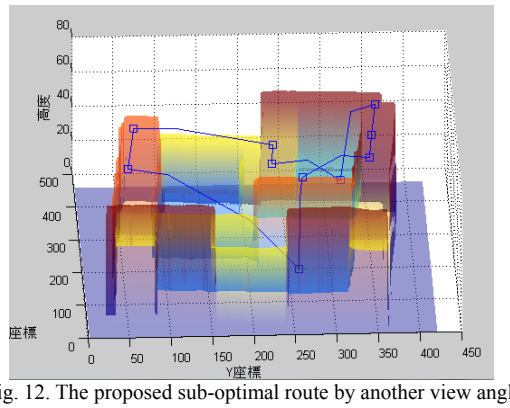



Fig. 12. The proposed sub-optimal route by another view angle.

## V. CONCLUSIONS

We have to emphasize that this paper is studying a practical problem. Using a SCARA to operate on a 3D work-piece is a very common task in an automated factory. Obtaining the shortest obstacle free route for a SCARA to operate on a 3D work-piece is an important issue. If the robot can perform its function in the most efficient way possible then time will be saved and electricity costs will be reduced simultaneously. In this study, a Kinect camera is used to detect the spatial information of the work-piece, but the details of the process used by the Kinect to form the image is of no concern in this study. The shortest route planning is the main purpose of the paper. Many techniques such as visibility graphing, Dijkstra's algorithm, PSO for the TSP, and the 2-Opt exchange method *etc*. have been utilized to perform the sub-optimal route planning. The detailed procedure has been proposed and a video of the actual experimental SCARA operation has been made available for viewing on Youtube. It is seen that the proposed route planning is effective and successful.

However, we have to admit there are still some improvements that can be made in future studies. For instance, if the obstacle between two invisible holes has a complex shape, such as narrower in the upper part and wider in the lower part, then the method we used to locate the corner of the obstacle on the 2D profile would no longer be applicable since the position of the corner is obtained from a 2D profile based on the lower hole. On this lower 2D plane, the corner will be much more prominent than it is at height $h_c$ of the obstacle. Therefore a route which goes around the corner as projected on the lower 2D profile will be longer than that going around the corner at the height $h_c$. Moreover, the additional point (E point) is a little away from but at the same height as the higher hole. We did not prove that an E point at the same height as the higher hole is suitable. This was simply an intuitive placement. There may be a much better position than E point to be found in the future. Furthermore, the 2-Opt exchange method may be not the best modification for the PSO solution. Because of the above concerns, we must title this study "A sub-optimal route planning……." instead of "An optimal route planning…..". The considered 3D object in this study is a number of stacked wooden blocks which is not too complex. If the shape of the 3D object is curved or has multiple curved facets, the 3D route planning would be a big challenge for engineers.

## REFERENSCES

[1] M. R. Pedersen *et al.*, "Robot skills for manufacturing: From concept to industrial deployment," *Robotics and Computer-Integrated Manufacturing,* vol. 37, pp. 282-291, 2016.

[2] K. Feng, X. Zhang, H. Li, and Y. Huang, "A dual-camera assisted method of the SCARA robot for online assembly of cellphone batteries," *Intelligent Robotics and Applications*, 2017, pp. 576-587: Springer International Publishing.

[3] N. Furuya, K. Soma, E. Chin, and H. Makino, "Research and development of selective compliance assembly robot arm (2nd

report) : Hardware and software of SCARA controller," *Journal of the Japan Society of Precision Engineering*, vol. 49, no. 7, pp. 835-841, 1983.

[4] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. ACM*, vol. 22, no. 10, pp. 560-570, 1979.

[5] H. Han-Pang and C. Shu-Yun, "Dynamic visibility graph for path planning," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 2004, vol. 3, pp. 2813-2818.

[6] G. R. Raidl and B. A. Julstrom, "A weighted coding in a genetic algorithm for the degree-constrained minimum spanning tree problem," *Proceedings of the 2000 ACM symposium on Applied computing*, Como, Italy, 2000.

[7] A. W. Mohemmed, N. C. Sahoo, and T. K. Geok, "Solving shortest path problem using particle swarm optimization," *Applied Soft Computing*, vol. 8, no. 4, pp. 1643-1653, 2008.

[8] R. Hassan, B. Cohanim, O. de Weck, and G. Venter, "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," *The 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2005.

[9] J. Tan, C. Wang, Y. Wang, Y. Chenxie, Y. Zhang, and Y. Wang, "Three-dimensional path planning based on ant colony algorithm with potential field For rotary-wing flying robot," *IEEE International Conference on Information and Automation*, 2015, pp. 2592-2597.

[10] S. D. Wickramaratna, A. S. Udage, R. T. Jayasekara, D. A. Kariyapperuma, W. A. S. P. Abeysiriwardhana, and A. M. H. S. Abeykoon, "Field based navigation for 3D obstacle avoidance," *Moratuwa Engineering Research Conference (MERCon)*, 2016, pp. 391-396.

[11] S. Lin, "Computer solutions of the traveling salesman problem," *The Bell System Technical Journal*, vol. 44, no. 10, pp. 2245-2269, 1965.

[12] J. Yang, X. Shi, M. Marchese, and Y. Liang, "An ant colony optimization method for generalized TSP problem," *Progress in Natural Science*, vol. 18, no. 11, pp. 1417-1422, 2008.

[13] K. Shaukat Ali, A. Sohail, and S. Fong, "Modeling TSP with Particle Swarm Optimization and Genetic Algorithm," *The 6th International Conference on Advanced Information Management and Service (IMS)*, 2010, pp. 455-459.

[14] G. Ye and X. Rui, "An improved simulated annealing andgenetic algorithm for TSP," *The 5th IEEE International Conference on Broadband Network & Multimedia Technology*, 2013, pp. 6-9.

[15] W. Elloumi, N. Baklouti, A. Abraham, and A. M. Alimi, "Hybridization of Fuzzy PSO and Fuzzy ACO applied to TSP," *The 13th International Conference on Hybrid Intelligent Systems (HIS 2013)*, 2013, pp. 105-110.

[16] E. Welzl, "Constructing the visibility graph for n-line segments in O(n2) time," *Information Processing Letters*, vol. 20, pp. 167- 171 , 1985.

[17] T. H. Cormen, C.E. Leiserson, R. L. Rivest, "Section 24.3: Dijkstra's algorithm". *Introduction to Algorithms (Second ed.)*. MIT Press and McGraw–Hill. pp. 595–601, 2001.

[18] P. Wu, K. J. Yang, and H. C. Fang, "A revised EM-Like algorithm + K-OPT method for solving the traveling salesman problem," *First International Conference on Innovative Computing, Information and Control, (ICICIC'06)*, Vol. 1, pp. 546-549, 2006.

[19] T. C. Kuo, "The study of 3D route optimization for the SCARA," *MS thesis*, NCU, Taiwan, June, 2017.

[20] Youtube video, available at https://www.youtube.com/watch?v=WlU8gKvxQz0, June, 2018.

**Wen-June Wang** received the B.S. degree in the Department of Control Engineering from National Chiao-Tung University, Hsin-Chu, Taiwan in 1980; and M.S. degree in the Department of Electrical Engineering from Tatung University, Taipei, Taiwan in 1984. Moreover, he received the Ph.D. degree in the Institute of Electronics from National Chiao-Tung University of Taiwan in 1987.

Prof. Wang is presently a Chair Professor of Department of Electrical Engineering. He was the Dean of College of Electrical Engineering and Computer Science, National Central University, Chung-Li City, Taiwan. He was also a Chair Professor and the Dean of the Research and Development Office of National Taipei University of Technology, Taipei, Taiwan in 2007~2009. In 2005~2007, he was the Dean of College of Science and Technology, National Chi-Nan University, Puli, NanTou, Taiwan. He was a visiting scholar for one year in the Department of Mechanical Engineering, Georgia Institute of Technology, USA in 1994 and a half year at the University of Louisville, Kentucky, USA, from September 2016 to February 2017. Moreover, Prof. Wang obtained the honor of IEEE Fellow in 2008 and IFSA Fellow in 2016.

Prof. Wang has authored or coauthored over 167 refereed journal papers and 165conference papers in the areas of fuzzy systems and theorems, robust and nonlinear control in large scale systems, and neural networks etc. His most significant contributions are the design of fuzzy systems and the development of robotics. His other research interests include the areas of robot control, neural networks, artificial intelligent, and pattern recognition etc.

**Sheng-Kai Huang** received the Bachelor's and Master's degree in the Mechanical and Electro-Mechanical Engineering from Tamkang University, Tamsui, Taiwan in 2014 and 2016, respectively. Now he is studying the Ph.D. degree in Department of Electrical Engineering, National Central University, Taiwan. Mr. Huang's research interests include robot control, image processing, system integration and deep learning. He also learned the fuzzy control systems, and neural networks from his advisor, Prof. Wen-June Wang.

**Tzu-Cheng Kuo** received the Bachelor's and Master's degrees in the Department of Electrical Engineering from National Central University, Chung-Li City, Taiwan respectively in 2015 and 2017. During Mr. Kuo studied the Master program, he has taken many courses, such as robotic control, fuzzy control, neural networks, and image processing etc. Under the supervision of Prof. W.-J. Wang, Mr. Kuo has finished his Master thesis which is titled "The study of 3D route optimization for the SCARA".

**Hsiang-Chieh Chen** received his B.S. degree in engineering science from the National Cheng Kung University, Tainan, Taiwan, in 2002 and his M.S. and Ph.D. degrees in electrical engineering from the National Central University, Taoyuan, Taiwan, in 2005 and 2009. From Aug. 2010 to Sep. 2014, he was with the Industrial Technology Research Institute, where he worked on computer vision and computational intelligence. From Sep. 2014 to Mar. 2015, he was with the Altek Semiconductor and focused on stereo vision R&D. From Mar. 2015 to Jul. 2016, he served as a technical leader of computer vision with the Coretronic Corporation and was in charge of innovative technology development. Since Aug. 2016, he joined the faculty of the Department of Electrical Engineering, National United University, Miaoli, Taiwan, where he is currently an assistant professor. His research interests include computer vision, computational intelligence, and robotics.

**Van-Phong Vu** received the B.S. degree in the Department of Automatic Control from Hanoi University of Sciences and Technology, Hanoi, Vietnam in 2007; and M.S. degree in the Department of Electrical Engineering from Southern Taiwan University of Sciences and Technology, Tainan, Taiwan in 2010. Moreover, he received the Ph.D. degree in the Department of Electrical Engineering from National Central University, Jhongli, Taiwan, in 2017.

Dr. Vu is a Lecturer with the Ho Chi Minh City University of Education and Technology, Ho Chi Minh city, Vietnam since 2012. He is also a Postdoctoral Research Fellow in the Department of Electrical Engineering, National Central University, Taiwan from Nov. 2017 to July 2018. He has authored or coauthored of six SCI papers, three of them are published on the IEEE Transactions on Fuzzy Systems journal. Moreover, he also published the some conference papers on the international conferences. He also received the best paper wards of the IFuzzy Conference in 2015 and ICSSE in 2017. His research interests include the fuzzy system, intelligent control, observer, and controller design for the uncertain system, polynomial system, fault estimation, large-scale system.

# 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 2 JUNE, 2018

# CONTENTS

## REGULAR PAPERS



## TECHNICAL NOTE