


Article

# FPGA-Based Mechatronic Design and Real-Time Fuzzy Control with Computational Intelligence Optimization for Omni-Mecanum-Wheeled Autonomous Vehicles

Hsu-Chih Huang , Chin-Wang Tao, Chen-Chia Chuang and Jing-Jun Xu

Department of Electrical Engineering, National Ilan University, Yilan 26047, Taiwan; cwtao@niu.edu.tw (C.-W.T.); ccchuang@niu.edu.tw (C.-C.C.); yehvon4@gmail.com (J.-J.X.)

\* Correspondence: hchuang@niu.edu.tw

Received: 14 October 2019; Accepted: 7 November 2019; Published: 11 November 2019



**Abstract:** This study presents a field-programmable gate array (FPGA)-based mechatronic design and real-time fuzzy control method with computational intelligence optimization for omni-Mecanum-wheeled autonomous vehicles. With the advantages of cuckoo search (CS), an evolutionary CS-based fuzzy system is proposed, called CS-fuzzy. The CS's computational intelligence was employed to optimize the structure of fuzzy systems. The proposed CS-fuzzy computing scheme was then applied to design an optimal real-time control method for omni-Mecanum-wheeled autonomous vehicles with four wheels. Both vehicle model and CS-fuzzy optimization are considered to achieve intelligent tracking control of Mecanum mobile vehicles. The control parameters of the Mecanum fuzzy controller are online-adjusted to provide real-time capability. This methodology outperforms the traditional offline-tuned controllers without computational intelligences in terms of real-time control, performance, intelligent control and evolutionary optimization. The mechatronic design of the experimental CS-fuzzy based autonomous mobile vehicle was developed using FPGA realization. Some experimental results and comparative analysis are discussed to examine the effectiveness, performance, and merit of the proposed methods against other existing approaches.

**Keywords:** computational intelligence; evolutionary; real-time; Mecanum vehicle

## 1. Introduction

Real-time fuzzy control is a modern control strategy for a variety of challenging control applications because it provides a convenient method for constructing nonlinear controllers [1–3]. Compared with conventional controllers, this approach has significantly improved the performance by considering the control parameter tuning issue. The control parameters are self-tuned in this kind of real-time controller to achieve optimal performance using fuzzy theory. This approach provides a formal methodology for representing, manipulating, and implementing human heuristic knowledge to develop intelligent controllers [4–6].

Although the real-time fuzzy control method has been widely used in many real-world disciplines, there remain a number of drawbacks in the design stages. One major difficulty is the construction of the fuzzy model in terms of the center and width of the membership functions and fuzzy rules [5,6]. Although the identification procedure of a fuzzy model can be manually tuned, the tuning procedure is time-consuming for complex systems. In recent years, some auto-tuned fuzzy modeling methods have been presented, however, these works are computationally intensive and they do not provide a general optimization method [6].

In order to meet these challenges, artificial evolution can provide robustness and scalability. It is a well-established approach to the design and optimization of intelligent systems [7,8]. Biologically inspired computational intelligence has been identified as an efficient tool for the optimal construction of fuzzy systems. To date, some popular evolutionary algorithms have been employed to develop metaheuristic fuzzy systems for engineering optimization. For example, GA (genetic algorithm)-fuzzy, PSO (particle swarm optimization)-fuzzy, and ACO (ant colony optimization)-fuzzy are powerful hybrid swarm intelligences for fuzzy structure optimization [7–11]. This hybrid computational intelligence is a modern technology for solving real-world engineering problems and complex multidimensional optimization problems [7].

Metaheuristic algorithms are a subset of computational intelligence. This kind of computing paradigm has attracted attention over recent decades because of the algorithms' simplicity, flexibility, and local optimum avoidance [12–16]. In particular, some of them play an important role not only in academic society but also in many other practical engineering fields. It is an emerging field of artificial intelligence based on the behavior models of social insects in nature. This discipline deals with the natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization [17,18].

In the rapid development of metaheuristic algorithms, CS introduced by Yang and Deb is a population-based algorithm inspired by the brood parasitic behavior observed in the cuckoo bird species [19]. Compared with conventional optimization methods, these metaheuristics are potentially more generic and robust for many non-deterministic polynomial-time hard optimization problems. CS has been employed in diverse domains with great efficiency by exploiting its strong optimization ability, including controller design, chaotic systems, path planning, and sensor networks [19–21]. This study not only presents a CS-fuzzy optimization method, but also applies this intelligent system to the locomotion control of omni-Mecanum-wheeled autonomous vehicles.

Omni-Mecanum-wheeled vehicles are mobile vehicles equipped with four Mecanum omnidirectional wheels [22]. They are based on the characteristic of a central wheel with a number of rollers placed at an angle of  $45^\circ$  around the periphery of the wheel. They provide the omnidirectional ability to move instantaneously in any direction from any configuration without changing the direction of the wheel. This kind of mobile robot with four degrees of freedom (DOFs) has been used in vast applications instead of conventional vehicles due to its flexible mobility in performing difficult tasks in congested environments with complex obstacles [23,24]. To date, the omni-Mecanum-wheeled vehicles have been widely used in nursing-care robots, intelligent wheelchairs, industrial robots, and nursing-care robots [23,24].

Four-DOF Mecanum vehicles are categorized as redundant vehicles. They have quite different features compared with three-DOF holonomic and two-DOF non-holonomic mobile vehicles. Several studies [24,25] have been presented to address the mathematical vehicle modeling and nonlinear control problem of Mecanum vehicles. However, these control schemes neither addressed the optimal control problem, nor presented real-time fuzzy controllers. This work not only presents an evolutionary CS-fuzzy optimization, but also designs an FPGA-based real-time CS-fuzzy controller for omni-Mecanum-wheeled vehicles. With the advantages of FPGA realization [26–29], this study presents the FPGA-based mechatronic design for the omni-Mecanum-wheeled vehicle with CS-fuzzy computational intelligence optimization.

The objective of this study was to develop a hybrid CS-fuzzy optimization technique for application to evolutionary real-time fuzzy control omni-Mecanum-wheeled autonomous vehicles. With the advantages of FPGA, metaheuristic algorithm, and fuzzy theory, the proposed CS-fuzzy control method outperforms the traditional fuzzy controllers. Moreover, the proposed approach has significantly improved the performance by considering the control parameter tuning issue. The remainder of this paper is organized as follows. In Section 2, the hybrid CS-fuzzy optimization approach is introduced. Section 3 details the method for employing the proposed CS-fuzzy computational intelligence to design an intelligent real-time CS-fuzzy locomotion controller for Mecanum autonomous

vehicles. Section 4 presents the mechatronic design and FPGA implementation of the CS-fuzzy-based omni-Mecanum-wheeled autonomous vehicle. Section 5 discusses several experimental results and comparative works to demonstrate the performance and merit of the proposed methods. Concluding remarks are provided in Section 6.

## 2. CS-Fuzzy Computational Intelligence Optimization

### 2.1. Classical Fuzzy Control System

Fuzzy control systems are considered a kind of intelligent control scheme. They can utilize some human knowledge to construct logical inference rules. With the advantages of fuzzy theory, it has shown potential in designing controllers for complex nonlinear systems and ill-formulated systems. This practical control technology has been successfully used in solving many real-world optimization problems.

Fuzzy logic IF-THEN rules are utilized to describe fuzzy control systems. These rules have to be established based on human expert knowledge, and they must be implemented by performing rigorous logical operations. This study adopts a fuzzy model with  $i$  rules. The fuzzy inference system is expressed by

$$R_i : \text{If } x_1 \text{ is } A_{i1} \text{ AND } \dots \text{ AND } x_n \text{ is } A_{in} \text{ THEN } y_i \text{ is } a_i \quad (1)$$

where  $x = (x_1, \dots, x_n)$  is the input,  $y_i$  denotes the output of  $i$ th rule.  $A_{i1}, \dots, A_{in}$  are fuzzy set in the antecedent part and  $a_i$  is a real number in the consequent part of  $R_i$ . The output  $y$  is calculated by the weighted average defuzzification method as

$$y = \frac{\sum_{i=1}^K \Phi_i(x) a_i}{\sum_{i=1}^K \Phi_i(x)} \quad (2)$$

where  $K$  denotes the number of rules,  $\Phi_i(x)$  is the firing strength of the rule  $R_i$ :

$$\Phi_i(x) = \prod_{j=1}^n \mu_{A_{ij}}(x_j), \quad i = 1, 2, \dots, K \quad (3)$$

where  $\mu_{A_{ij}}(x_j)$  is the membership function of the fuzzy set  $A_{ij}$  in the antecedent of  $R_i$ .

### 2.2. CS Algorithm

Cuckoo Search is a nature-inspired optimization algorithm based on the cuckoo bird's parasitic breeding behavior of laying its eggs in the nests of different bird species. The cuckoo bird deposits her eggs in the nest of a host bird whose eggs closely resemble her own eggs and uses the services of host birds to hatch her own eggs. This parasitic behavior increases the chance of survival of the cuckoo's genes since the cuckoo need not expend any energy rearing its offspring. The cuckoo search algorithm simulates the cuckoo bird's intelligent search strategy of finding the best host nest to deposit her egg. It utilizes the brood parasitism behavior in order to explore the search space and search for optimal solutions.

In CS computational intelligence applied to solving optimization problems, a set of nests with one egg inside is randomly placed in the search space and each egg represents a candidate solution for optimization problems. The cuckoos traverse the search space and CS records the highest fitness values for the different candidate solutions. A special search pattern called Lévy flight is employed in real cuckoo birds and insects. This characteristic is more efficient for global search rather than random walks or some conventional motions. Each cuckoo  $i$  in CS lays an egg (the potential candidate solution) at a random location by performing the Lévy flight, which is characterized by

$$x_i(t + 1) = x_i(t) + \alpha \oplus Levy(\beta) \tag{4}$$

where  $t$  is the iteration number,  $x_i(t)$  is the location of  $i$ th nest at  $t$ th iteration.  $\alpha > 0$  is the step size and  $\oplus$  is an entry-wise multiplication.  $Levy(\beta)$  is a specialized random walk behavior, formulated by the form:

$$Levy(\beta) = \left| \frac{\Gamma(\beta + 1) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right|^{\frac{1}{\beta}} \tag{5}$$

where  $\Gamma$  is the gamma function  $\Gamma(x) = \int_0^\infty e^{-t}t^{x-1}dt$ .  $\beta$  denotes a constant ( $1 < \beta \leq 3$ ).

### 2.3. Hybrid CS-Fuzzy Optimization to Real-Time Control

CS has some attractive features for dealing with optimization problems, including great flexibility and robust ability. This section aims to develop a hybrid control scheme using the CS-fuzzy optimization technique. The CS algorithm is employed to determine the fuzzy structures. To allow easy implementation, in this study, the triangular membership functions in  $R_i$  are defined by the center  $c_i$ , the left vertex  $a_i$  and the right vertex  $b_i$ . The design parameters of the fuzzy structure for real-time control include the number of rules  $K$ ,  $c_i$ ,  $a_i$  and  $b_i$ . Figure 1 depicts the architecture of the proposed CS-fuzzy control system. Having the CS-optimized fuzzy system, the intelligent CS-fuzzy control strategy is applied to real plants to obtain an optimal feedback control system.

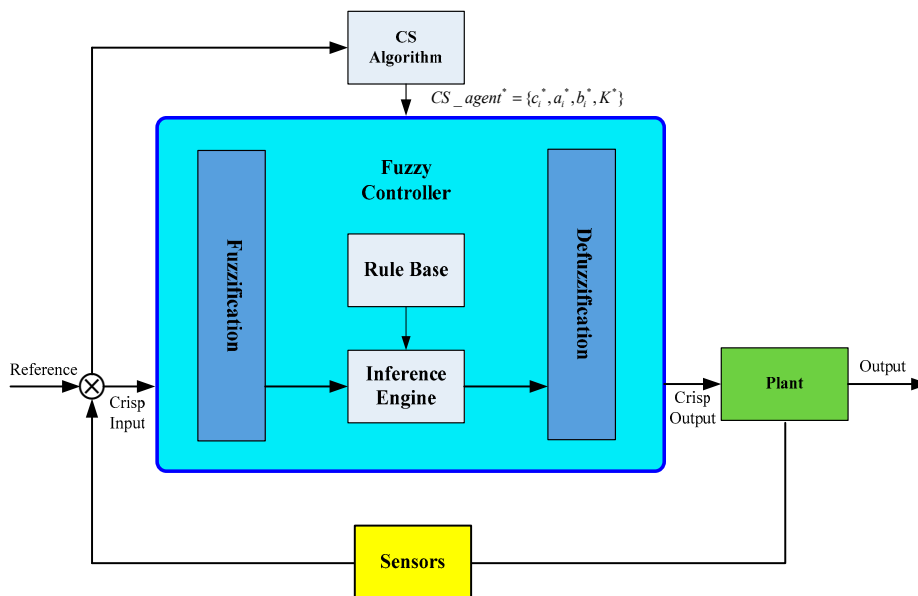


Figure 1. Architecture of the proposed cuckoo search (CS)-fuzzy control system.

In the CS-based approach for optimizing fuzzy modeling, each CS agent is defined as a specific fuzzy model, and the goal is to design an optimal fuzzy structure. In this work, this parameter optimization is addressed by using the evolutionary CS algorithm, thereby optimizing the fuzzy structure parameters. Compared with conventional fuzzy systems, the proposed biological CS-fuzzy optimization takes the advantages of metaheuristic algorithm and fuzzy theory. In the proposed CS-optimized fuzzy system, a CS agent is defined by the fuzzy parameters  $CS\_agent = \{c_i, a_i, b_i, K\}$ . This intelligent control system features feedback control and artificial intelligence. Based on the reference signals, the CS-fuzzy control system with evolved  $CS\_agent^* = \{c_i^*, a_i^*, b_i^*, K^*\}$  controls the plant to achieve the control goal. Some sensors are employed in the closed-loop control system to measure and convert the real-world signals.

With the measured signals, the system errors are updated at every sampling period. The proposed CS-fuzzy control scheme generates output commands to drive the plant by considering the plant model. Worthy of mention is that the parameters in the derived control laws are online-tuned via the CS-fuzzy optimization technique. This real-time control strategy is superior to the traditional fixed-parameter control methods and off-line approach because the control parameters are self-adjusting at every sampling period. To evaluate the performance of the CS-based fuzzy modeling, a fitness function (performance index) is defined to evaluate the agents. This fitness function can be predefined according to the optimization problems. Typically, the least square error is employed to evaluate the CS-fuzzy system, described by equations of the form

$$Fitness = \frac{1}{N_s} \sum_{i=1}^{N_s} (y_f - y_d)^2 \tag{6}$$

where  $y_f$  and  $y_d$  respectively denote the fuzzy model outputs and desired output.  $N_s$  denotes the number of sampling data.

### 3. Real-Time CS-Fuzzy Control for Mecanum Vehicles

#### 3.1. Robot Kinematics Analysis

Robot kinematics deals with the configuration of robots in their workspace or environment. It describes the relations between a robot’s geometric parameters and the workspace. The kinematic equations depend on the geometrical structure of the robot. Kinematics analysis is an important topic for the study of the stability and control of robots. In robotic research, direct and inverse kinematics are two key issues in vehicle modeling. Direct kinematics determines the vehicle motion when the angular velocities of the wheels are given. The inverse kinematic problem uses the velocities of the vehicle to determine the angular velocities of the wheels. Figure 2 presents the symmetrical layout structure of the omni-Mecanum-wheeled autonomous vehicle equipped with four wheels. The inverse kinematic equation is presented as follows:

$$V_W = TV_C \tag{7}$$

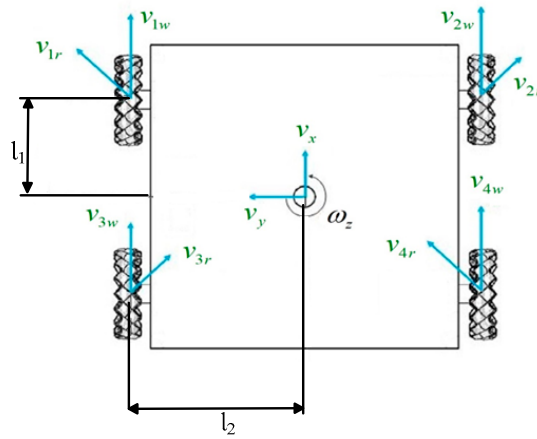
where  $V_W = [v_{1w} \ v_{2w} \ v_{3w} \ v_{4w}]^T$  represents the wheel’s velocity vector corresponding to the angular velocity,  $V_C = [v_x \ v_y \ \omega_z]^T$  is the velocity vector in Cartesian coordinates.  $T$  is a transformation matrix, expressed by

$$T = \begin{bmatrix} 1 & -1 & -(l_1 + l_2) \\ 1 & 1 & (l_1 + l_2) \\ 1 & 1 & -(l_1 + l_2) \\ 1 & -1 & (l_1 + l_2) \end{bmatrix}$$

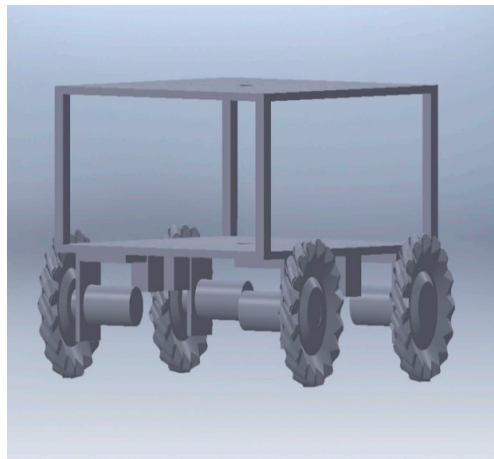
The direct kinematic equation vehicle velocity can be obtained from the wheel velocity using a pseudo inverse matrix approach as follows:

$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ \frac{-1}{(l_1+l_2)} & \frac{1}{(l_1+l_2)} & \frac{-1}{(l_1+l_2)} & \frac{1}{(l_1+l_2)} \end{bmatrix} \begin{bmatrix} R\dot{\theta}_1 \\ R\dot{\theta}_2 \\ R\dot{\theta}_3 \\ R\dot{\theta}_4 \end{bmatrix} \tag{8}$$

where  $v_{iw} = R\dot{\theta}_i, i = 1, 2, 3, 4, R$  denotes the radius of the Mecanum omnidirectional wheel.  $l_1$  and  $l_2$  are the distances between the center of gravity in length and width direction, respectively.



(a)



(b)

**Figure 2.** Structure of the omni-Mecanum-wheeled autonomous vehicle (a) bottom view (b) front view.

The velocities and the angular velocity of the vehicle in Equations (7)–(8) are derived in the moving frame. By considering the position and orientation of the omni-Mecanum-wheeled vehicle with the vector  $[x_w \ y_w \ \theta]^T$  in world frame, one obtains the following transformation matrix:

$$\begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} \tag{9}$$

Combining Equations (9) and (8), the direct kinematics model of the Mecanum vehicle in a world frame is formulated by

$$\begin{aligned} \begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\theta} \end{bmatrix} &= \frac{1}{4} \begin{bmatrix} (\cos \theta + \sin \theta) & (\cos \theta - \sin \theta) & (\cos \theta - \sin \theta) & (\cos \theta + \sin \theta) \\ -(\cos \theta - \sin \theta) & (\cos \theta + \sin \theta) & (\cos \theta + \sin \theta) & -(\cos \theta - \sin \theta) \\ \frac{-1}{(l_1+l_2)} & \frac{1}{(l_1+l_2)} & \frac{-1}{(l_1+l_2)} & \frac{1}{(l_1+l_2)} \end{bmatrix} \begin{bmatrix} v_{1w} \\ v_{2w} \\ v_{3w} \\ v_{4w} \end{bmatrix} \\ &= \frac{1}{4} \begin{bmatrix} \sqrt{2} \sin \vartheta & \sqrt{2} \cos \vartheta & \sqrt{2} \cos \vartheta & \sqrt{2} \sin \vartheta \\ -\sqrt{2} \cos \vartheta & \sqrt{2} \sin \vartheta & \sqrt{2} \sin \vartheta & -\sqrt{2} \cos \vartheta \\ \frac{-1}{(l_1+l_2)} & \frac{1}{(l_1+l_2)} & \frac{-1}{(l_1+l_2)} & \frac{1}{(l_1+l_2)} \end{bmatrix} \begin{bmatrix} v_{1w} \\ v_{2w} \\ v_{3w} \\ v_{4w} \end{bmatrix} = P^+(\theta) \begin{bmatrix} v_{1w} \\ v_{2w} \\ v_{3w} \\ v_{4w} \end{bmatrix} \end{aligned} \tag{10}$$

where  $\vartheta = \theta + \frac{\pi}{4}$ . The inverse kinematics model of the Mecanum-wheeled vehicle in a world frame is governed by

$$\begin{bmatrix} v_{1w} \\ v_{2w} \\ v_{3w} \\ v_{4w} \end{bmatrix} = \begin{bmatrix} \sqrt{2} \sin(\vartheta) & -\sqrt{2} \cos(\vartheta) & -(l+L) \\ \sqrt{2} \cos(\vartheta) & \sqrt{2} \sin(\vartheta) & (l+L) \\ \sqrt{2} \cos(\vartheta) & \sqrt{2} \sin(\vartheta) & -(l+L) \\ \sqrt{2} \sin(\vartheta) & -\sqrt{2} \cos(\vartheta) & (l+L) \end{bmatrix} \begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\theta} \end{bmatrix} = P(\theta) \begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\theta} \end{bmatrix} \quad (11)$$

where  $P^+(\theta)$  is the left pseudo inverse matrix of  $P(\theta)$ .

### 3.2. CS-Fuzzy Motion Control

#### 3.2.1. Control Law Design via Vehicle Kinematics

Motion control is a core issue in robotics research, aiming at generating the smooth movement of the robot by considering the vehicle’s kinematics model. This subsection aims at presenting a motion control scheme with trajectory tracking capability for omni-Mecanum-wheeled robots. The smooth and differentiable trajectories of the Mecanum vehicle can be planned via the control law to accomplish motion control. Defining the desired trajectory  $M_d(t) = [x_d(t) \ y_d(t) \ \theta_d(t)]^T$ , the trajectory tracking error of the vehicle is expressed by

$$M_e(t) = \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} = \begin{bmatrix} x_w(t) \\ y_w(t) \\ \theta(t) \end{bmatrix} - \begin{bmatrix} x_d(t) \\ y_d(t) \\ \theta_d(t) \end{bmatrix} \quad (12)$$

where  $[x_w(t) \ y_w(t) \ \theta(t)]^T$  is the position and orientation of Mecanum mobile vehicle at time stamp  $t$ . Thus, one obtains the time derivative of  $M_e(t)$  as follows

$$\dot{M}_e(t) = \begin{bmatrix} \dot{x}_e(t) \\ \dot{y}_e(t) \\ \dot{\theta}_e(t) \end{bmatrix} = \begin{bmatrix} \dot{x}_w(t) \\ \dot{y}_w(t) \\ \dot{\theta}(t) \end{bmatrix} - \begin{bmatrix} \dot{x}_d(t) \\ \dot{y}_d(t) \\ \dot{\theta}_d(t) \end{bmatrix} = P^+(\theta(t)) \begin{bmatrix} R\omega_1(t) \\ R\omega_2(t) \\ R\omega_3(t) \\ R\omega_4(t) \end{bmatrix} - \begin{bmatrix} \dot{x}_d(t) \\ \dot{y}_d(t) \\ \dot{\theta}_d(t) \end{bmatrix} \quad (13)$$

The design goal of control law is to determine the angular velocities  $U = [\omega_1(t) \ \omega_2(t) \ \omega_3(t)]^T$  of wheel motors such that the robotic system is globally and asymptotically stable. In this paper, the following trajectory tracking law is proposed

$$U(t) = \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \\ \omega_4(t) \end{bmatrix} = \frac{1}{R} P(\theta(t)) \left( -K_p \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} - K_I \begin{bmatrix} \int_0^t x_e(\tau) d\tau \\ \int_0^t y_e(\tau) d\tau \\ \int_0^t \theta_e(\tau) d\tau \end{bmatrix} + \begin{bmatrix} \dot{x}_d(t) \\ \dot{y}_d(t) \\ \dot{\theta}_d(t) \end{bmatrix} \right) \quad (14)$$

where  $K_p$  and  $K_I$  are control matrices; they are set as symmetric and positive definite. Substituting Equation (14) into Equation (13) leads to the closed-loop error system, governed by

$$\begin{aligned} \dot{M}_e(t) &= \begin{bmatrix} \dot{x}_e(t) \\ \dot{y}_e(t) \\ \dot{\theta}_e(t) \end{bmatrix} = P^+(\theta(t)) P(\theta(t)) \left( -K_p \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} - K_I \begin{bmatrix} \int_0^t x_e(\tau) d\tau \\ \int_0^t y_e(\tau) d\tau \\ \int_0^t \theta_e(\tau) d\tau \end{bmatrix} + \begin{bmatrix} \dot{x}_d(t) \\ \dot{y}_d(t) \\ \dot{\theta}_d(t) \end{bmatrix} \right) - \begin{bmatrix} \dot{x}_d(t) \\ \dot{y}_d(t) \\ \dot{\theta}_d(t) \end{bmatrix} \\ &= -K_p \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} - K_I \begin{bmatrix} \int_0^t x_e(\tau) d\tau \\ \int_0^t y_e(\tau) d\tau \\ \int_0^t \theta_e(\tau) d\tau \end{bmatrix} \end{aligned} \quad (15)$$

To prove the globally asymptotical stability of the proposed control scheme for the omni-Mecanum-wheeled system by means of Lyapunov stability theory, one selects the quadratic Lyapunov function as follows:

$$V(t) = \frac{1}{2} \begin{bmatrix} x_e(t) & y_e(t) & \theta_e(t) \end{bmatrix} \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \int_0^t x_e(\tau) d\tau & \int_0^t y_e(\tau) d\tau & \int_0^t \theta_e(\tau) d\tau \end{bmatrix} K_I \begin{bmatrix} \int_0^t x_e(\tau) d\tau \\ \int_0^t y_e(\tau) d\tau \\ \int_0^t \theta_e(\tau) d\tau \end{bmatrix} \quad (16)$$

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

$$\begin{aligned} \dot{V}(t) &= \begin{bmatrix} x_e(t) & y_e(t) & \theta_e(t) \end{bmatrix} \begin{bmatrix} \dot{x}_e(t) \\ \dot{y}_e(t) \\ \dot{\theta}_e(t) \end{bmatrix} + \begin{bmatrix} \int_0^t x_e(\tau) d\tau & \int_0^t y_e(\tau) d\tau & \int_0^t \theta_e(\tau) d\tau \end{bmatrix} K_I \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} \\ &= \begin{bmatrix} x_e(t) & y_e(t) & \theta_e(t) \end{bmatrix} \left( -K_p \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} - K_I \begin{bmatrix} \int_0^t x_e(\tau) d\tau \\ \int_0^t y_e(\tau) d\tau \\ \int_0^t \theta_e(\tau) d\tau \end{bmatrix} \right) + \begin{bmatrix} \int_0^t x_e(\tau) d\tau & \int_0^t y_e(\tau) d\tau & \int_0^t \theta_e(\tau) d\tau \end{bmatrix} K_I \begin{bmatrix} x_e(t) \\ y_e(t) \\ \theta_e(t) \end{bmatrix} \\ &= -\begin{bmatrix} x_e(t) & y_e(t) & \theta_e(t) \end{bmatrix} K_p \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} < 0 \end{aligned}$$

This result implies that the globally asymptotic stability of the proposed feedback robotic system is ensured, meaning that the position and orientation errors asymptotically approach zero as time tends to infinity.

### 3.2.2. Real-Time Motion Control Scheme Using CS-Fuzzy Computing

Although the nonlinear motion control law of the omni-Mecanum-wheeled vehicle is synthesized in Equation (14), there exists a control gain parameter tuning problem in this redundant robotic system. In other words, the control gain matrices  $K_p = \text{diag}[k_{xp} \ k_{yp} \ k_{\theta p}]$  and  $K_I = \text{diag}[k_{xi} \ k_{yi} \ k_{\theta i}]$  affect the performance of the closed-loop control system. To improve the diversity of motion and increase the performance requirements, this subsection aims to develop a CS-fuzzy real-time motion control methodology. This intelligent control strategy is superior to traditional control approaches because the gain matrices are online-optimized via the proposed evolutionary CS-fuzzy computational intelligence optimization.

The parameter tuning of control gains can be regarded as a metaheuristic optimization problem. This study employed the proposed CS-fuzzy optimization to design a self-tuning CS-fuzzy online controller, in which the control gain parameters are self-organising at every sampling period to achieve motion control of omni-Mecanum-wheeled vehicles. This evolutionary real-time control strategy is superior to traditional controllers using off-line tuning and hand-tuning methods because it provides the advantages of the CS, fuzzy theory, and real-time optimal control approach. The fitness function of the proposed CS-fuzzy optimal motion controller can be defined by the integral square error (ISE), expressed by

$$ISE = \int_0^t w_e(x_e^2(\tau) + y_e^2(\tau) + \theta_e^2(\tau)) + w_r K \quad (17)$$

where  $w_e$  and  $w_r$  are two weighting factors.  $x_e$  and  $y_e$  are the position errors in the Cartesian coordinates and  $\theta_e$  is the error of the vehicle's orientation.  $K$  is the number of rule bases. Notice that the proposed ISE-based intelligent CS-fuzzy control system is applicable to IAE (integral absolute error) and ITAE (integral time-weighted absolute error) based fuzzy systems by easily redefining the CS fitness function in Equation (17).

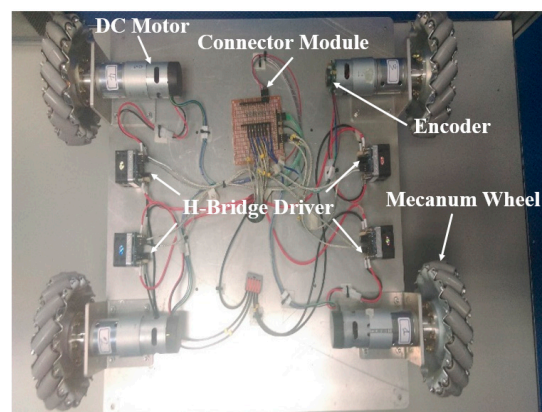
Having the predefined fitness function in the proposed omni-Mecanum-wheeled robotic system, the proposed CS computing paradigm aims to determine the optimal fuzzy structure  $CS\_agent^* = \{c^*_i, a^*_i, b^*_i, K^*\}$  in triangular membership functions. Both the structure of fuzzy



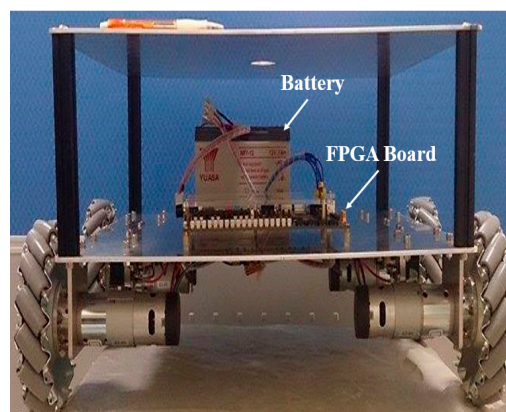
membership functions and the number of rule bases are included in the presented CS-fuzzy control system. Notice that this work presents a general CS-fuzzy optimization. Furthermore, this approach is applicable to solving other engineering optimization problems by easily redefining the fitness function in Equation (17). The control gain matrices  $K_P = \text{diag}[k_{xp} \ k_{yp} \ k_{\theta p}]$  and  $K_I = \text{diag}[k_{xi} \ k_{yi} \ k_{\theta i}]$  of the mobile vehicle are online self-tuned via the CS-fuzzy evolutionary process, thus obtaining optimal control performance.

#### 4. Mechatronic Design and FPGA Implementation

Figure 3 presents the FPGA-based mechatronic design of the proposed omni-Mecanum-wheeled autonomous vehicle. As shown in Figure 3a, the four Mecanum omnidirectional wheels were connected to the DC motors. A photo encoder was directly mounted on the DC motor to detect the QEP (quadrature encoder pulse) feedback signal. Four H-bridge power driver units were used to drive the DC motor via PWM (pulse-width modulation) methodology. A connector board was designed to serve as an interface between the driver units and the FPGA development kit. With the advantages of FPGA realization, observable in Figure 3b, all the hardware circuits and software components were integrated in one chip. This is an efficient approach to present a cost-effective Mecanum robotic system to examine the proposed real-time CS-fuzzy control scheme. The FPGA development board was employed to perform the proposed intelligent CS-fuzzy motion control strategy. The QEP and PWM hardware circuits were developed using Verilog hardware description language (HDL) to connect the physical world and robotic cyber world. An FPGA embedded processor was utilized to execute the cyber computing, including CS-fuzzy computing and kinematics model-based control.



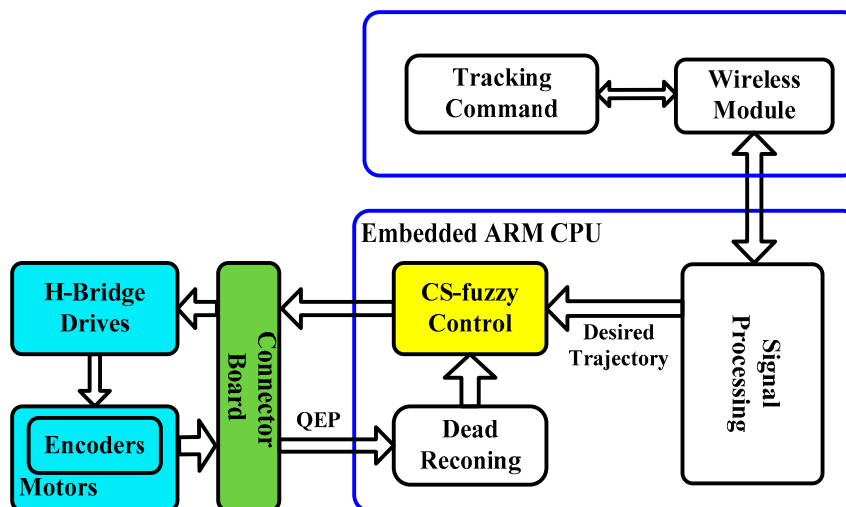
(a)



(b)

**Figure 3.** Field-programmable gate array (FPGA)-based mechatronic design of the proposed omni-Mecanum-wheeled autonomous vehicle (a) Bottom view (b) front view.

Figure 4 depicts the system structure of the omni-Mecanum-wheeled autonomous vehicle. When the FPGA embedded CPU received the feedback QEP signals from the motor's encoders, the dead reckoning technology [30–33] was applied to calculate the position and orientation of the Mecanum-wheeled vehicle. This FPGA-implemented approach is very useful in robotics research for determining pose information from sensors. The raw data captured from the physical world are converted to useful information in the cyber-world by means of this embedded processor. Possessing the current pose information and desired trajectory, the error vector is then obtained, thereby generating the control output based on the proposed intelligent CS-fuzzy control scheme. The control matrices are online-updated along with the sampled data at every sampling. Finally, the control commands are sent to PWM modules and H-bridge drivers to steer the motors in the physical world to achieve trajectory tracking task.



**Figure 4.** System structure of the omni-Mecanum-wheeled autonomous vehicle. (Quadrature encoder pulse = QEP).

In the FPGA realization of the omni-Mecanum-vehicle, a dual-core ARM Cortex-A9 CPU was employed to perform the cyber CS-fuzzy computing using C/C++ code for ultimate design flexibility. The model of the development kit is Altera DE1-SoC which presents a modern hardware design platform built around the Altera System-on-Chip (SoC) FPGA. The FPGA chip is a Cyclone V SoC device that integrates a hard processor system (HPS) consisting of processor, peripherals and memory interfaces tied seamlessly with the FPGA fabric. The DE1-SoC development board includes hardware such as high-speed DDR3 memory, video and audio capabilities and Ethernet networking. This hardware/software FPGA codesign methodology is regarded as a modern realization method to construct cost-effective mobile robots, because both the embedded processor and custom logics are integrated in one chip.

## 5. Experimental Results and Discussion

This section is devoted to discussing the experimental results and comparative analysis to highlight the contributions and merit of the proposed FPGA-based mechatronic design and intelligent CS-fuzzy real-time control for omni-Mecanum-wheeled mobile vehicles. The first experiment was conducted to examine the tracking performance of the CS-fuzzy online control for the omni-Mecanum-wheeled autonomous vehicle. The desired trajectory is mathematically expressed by  $\begin{bmatrix} x_d(t) & y_d(t) & \theta_d(t) \end{bmatrix}^T = \begin{bmatrix} r_d \cos \omega_r t \text{ (cm)} & r_d \sin \omega_r t \text{ (cm)} & 0 \text{ (rad)} \end{bmatrix}^T$ , where  $r_d = 64 + 40 \cos 4\omega_r t \text{ (cm)}$  and  $\omega_r = 0.05 \text{ (rad/sec)}$ . The initial pose in this experiment is set at the origin. Figure 5 presents the trajectory tracking result for this special curve and Figure 6 depicts the tracking errors of the orientation and position of the omni-Mecanum-wheeled vehicle. Through these

experimental results displayed in Figures 5 and 6, the proposed intelligent CS-fuzzy motion controller successfully steered the vehicle to track this curve. The position error and orientation are converged to zero in 10 s.

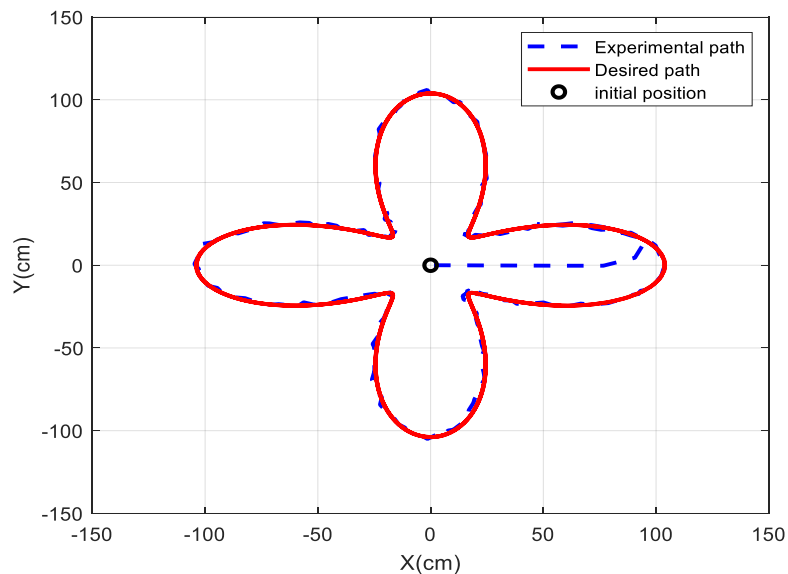


Figure 5. Experimental result of CS-fuzzy real-time trajectory tracking.

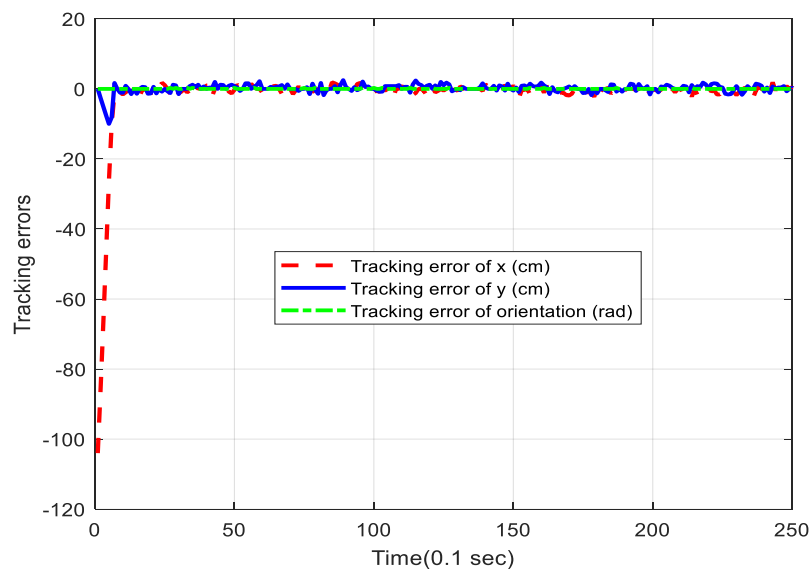


Figure 6. Experimental tracking errors of the position and orientation of the omni-Mecanum-vehicle.

In the proposed intelligent CS-fuzzy motion controller of the omni-Mecanum-vehicle, the predefined fitness value was used to evaluate the performance of the CS-fuzzy system. Both the pose error and the number of fuzzy rules are included in the fitness function. Figure 7 depicts the convergent behavior of the fitness value. As shown in Figure 7, the fitness value converges to constant successfully, meaning that the proposed CS-fuzzy intelligent controller online evolves the optimal control parameters. Figure 8 presents the output command of the CS-fuzzy controller to steer the Mecanum vehicle. The proposed CS-fuzzy generates smooth output commands for tracking the desired trajectory. To compare the proposed control scheme against other existing methods, this research performs the ACO-fuzzy and PSO-fuzzy tracking control tasks using the same experimental setting as in Figure 5. The fitness value in the ACO-fuzzy tracking controller is  $6.70 \times 10^4$  and the convergent fitness value using the PSO-fuzzy tracking approach is  $6.72 \times 10^4$ . These comparative works clearly

indicate that the proposed FPGA-based intelligent CS-fuzzy control scheme outperforms the traditional evolutionary fuzzy control systems.

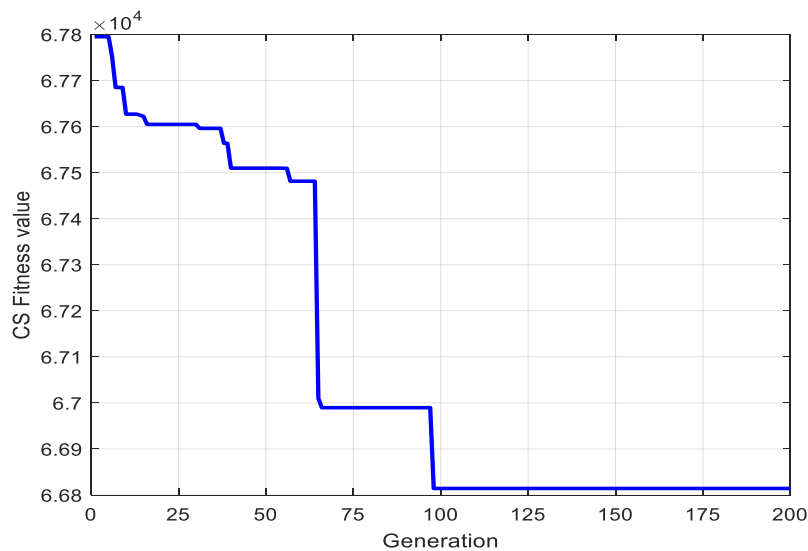


Figure 7. Behavior of the CS fitness value.

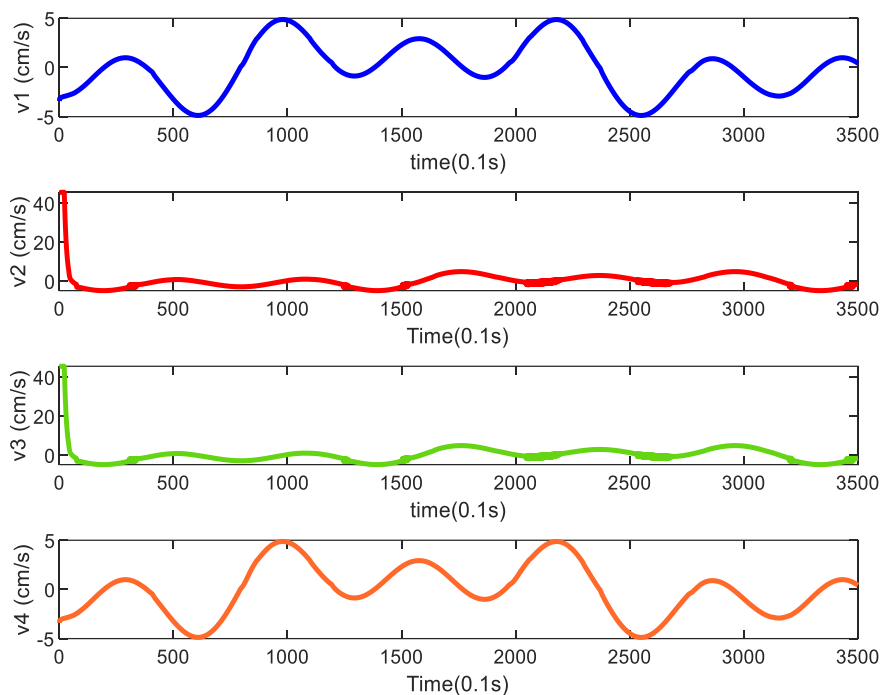


Figure 8. Output commands of the CS-fuzzy controller.

## 6. Conclusions

This study has presented a FPGA-based mechatronic design and real-time fuzzy control method with CS-fuzzy computational intelligence optimization for omni-Mecanum-wheeled autonomous vehicles. The proposed CS-fuzzy computing scheme was applied to synthesize an optimal real-time control for omni-Mecanum-wheeled autonomous vehicles with four wheels. The control gain matrices of the Mecanum fuzzy controller were online-adjusted to provide real-time capability. The mechatronic design of the experimental CS-fuzzy based autonomous mobile vehicle was developed using FPGA realization. Experimental results and comparative works were discussed to examine the effectiveness, performance and merit of the proposed methods against other existing approaches. The proposed

CS-fuzzy real-time control method can be applied to designing dynamics controllers of mobile vehicles by considering the forces and torques in future works.

**Author Contributions:** Conceptualization, methodology, analysis, writing—review and editing, H.-C.H., C.-W.T. and C.-C.C.; Experiments, J.-J.X.

**Funding:** This research was funded by the Ministry of Science and Technology, Taiwan, under grant number MOST 108-2221-E-197-030.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Pan, Y.; Er, M.J. Enhanced adaptive fuzzy control with optimal approximation error convergence. *IEEE Trans. Fuzzy Syst.* **2013**, *21*, 1123–1132. [[CrossRef](#)]
2. Mercorelli, P. Using fuzzy PD controllers for soft motions in a car-like robot. *Adv. Sci. Technol. Eng. Syst. J.* **2018**, *3*, 380–390. [[CrossRef](#)]
3. Goharimanesh, M.; Lashkaripour, A.; Shariatnia, S.; Akbari, A. Diabetic control using genetic fuzzy-PI controller. *Int. J. Fuzzy Syst.* **2014**, *16*, 133–138.
4. Hannan, M.A.; Ghani, Z.A.; Mohamed, A.; Uddin, M.N. Real-time testing of a fuzzy-logic-controller-based grid-connected photovoltaic inverter system. *IEEE Trans. Ind. Appl.* **2015**, *51*, 4775–4784. [[CrossRef](#)]
5. Mercorelli, P. Fuzzy based control of a nonholonomic car-like robot for drive assistant systems. In Proceedings of the 2018 19th International Carpathian Control Conference (ICCC), Szilvasvarad, Hungary, 28–31 May 2018; pp. 434–439.
6. Xu, S.S.; Huang, H.C.; Chiu, T.C.; Lin, S.K. Biologically-inspired learning and adaptation of self-evolving control for networked mobile robots. *Appl. Sci.* **2019**, *9*, 1034.
7. Fayaz, M.; Ullah, I.; Kim, D. An optimized fuzzy logic control model based on a strategy for the learning of membership functions in an indoor environment. *Electronics* **2019**, *8*, 132. [[CrossRef](#)]
8. Huang, H.C.; Chiang, C.H. Backstepping holonomic tracking control of wheeled robots using an evolutionary fuzzy system with qualified ant colony optimization. *Int. J. Fuzzy Syst.* **2016**, *18*, 28–40. [[CrossRef](#)]
9. Valdez, F.; Vazquez, J.C.; Gaxiola, F. Fuzzy dynamic parameter adaptation in ACO and PSO for designing fuzzy controllers: The cases of water level and temperature control. *Adv. Fuzzy Syst.* **2018**, *2018*, 1274969. [[CrossRef](#)]
10. Juang, C.F.; Jeng, T.L.; Chang, Y.C. An interpretable fuzzy system learned through online rule generation and multiobjective ACO with a mobile robot control application. *IEEE Trans. Cybern.* **2016**, *46*, 2706–2718. [[CrossRef](#)] [[PubMed](#)]
11. Chen, C.; Li, M.; Sui, J.; Wei, K.; Pei, Q. A genetic algorithm-optimized fuzzy logic controller to avoid rearend collisions. *J. Adv. Transp.* **2016**, *50*, 1735–1753. [[CrossRef](#)]
12. enavides-Álvarez, C.; Villegas-Cortez, J.; Román-Alonso, G.; Avilés-Cruz, C. Wiener-granger causality theory supported by a genetic algorithm to characterize natural scenery. *Electronics* **2019**, *8*, 726. [[CrossRef](#)]
13. Chao, K.H.; Hsieh, C.C. Photovoltaic module array global maximum power tracking combined with artificial bee colony and particle swarm optimization algorithm. *Electronics* **2019**, *8*, 603. [[CrossRef](#)]
14. Huang, H.C. Intelligent motion control for omnidirectional mobile robots using ant colony optimization. *Appl. Artif. Intell.* **2013**, *27*, 151–169. [[CrossRef](#)]
15. Liao, T.; Socha, K.; Oca, M.; T, S.; Dorigo, M. Ant colony optimization for mixed-variable optimization problems. *IEEE Trans. Evol. Comput.* **2014**, *18*, 503–518. [[CrossRef](#)]
16. Hasanien, H.M. Design optimization of PID controller in automatic voltage regulator system using Taguchi combined genetic algorithm method. *IEEE Syst. J.* **2013**, *7*, 825–831. [[CrossRef](#)]
17. Huang, H.C.; Xu, S.S.; Wu, C.H. A Hybrid swarm intelligence of artificial immune system tuned with Taguchi-genetic algorithm and its field-programmable gate array realization to optimal inverse kinematics for an articulated industrial robotic manipulator. *Adv. Mech. Eng.* **2016**, *8*. [[CrossRef](#)]
18. Wang, D.; Tan, D.; Liu, L. Particle swarm optimization algorithm: An overview. *Soft Comput.* **2017**, *22*, 387–408. [[CrossRef](#)]
19. Gandomi, A.; Yang, X.S.; Alavi, A. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **2013**, *29*, 17–35. [[CrossRef](#)]

20. Gandomi, A.H.; Talatahari, S.; Yang, X.S.; Deb, S. Design optimization of truss structures using cuckoo search algorithm. *Struct. Des. Tall Spec. Build.* **2013**, *22*, 1330–1349. [[CrossRef](#)]
21. Civicioglu, P.; Besdok, E. A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artif. Intell. Rev.* **2013**, *39*, 315–346. [[CrossRef](#)]
22. Song, J.B.; Byun, K.S. Design and control of a four-wheeled omnidirectional mobile robot with steerable omnidirectional wheels. *J. Robot. Syst.* **2004**, *21*, 193–208. [[CrossRef](#)]
23. Kim, J.; Woo, S.; Kim, J.; Do, J.; Kim, S.; Bae, S. Inertial navigation system for an automatic guided vehicle with mecanum wheels. *Int. J. Precis. Eng. Manuf.* **2012**, *13*, 379–386. [[CrossRef](#)]
24. Wang, Y.; Chang, D. Motion performance analysis and layout selection for motion system with four mecanum wheels. *J. Mech. Eng.* **2009**, *45*, 307–310. [[CrossRef](#)]
25. Kang, J.; Kim, B.; Chung, M. Development of omnidirectional mobile robots with mecanum wheels assisting the disabled in a factory environment. In Proceedings of the 8th International Conference on Control, Automation and Systems, Seoul, Korea, 14–17 October 2008; pp. 2070–2075.
26. Renteria-Cedano, J.; Rivera, J.; Sandoval-Ibarra, F.; Ortega-Cisneros, S.; Loo-Yau, R. SoC design based on a FPGA for a configurable neural network trained by means of an EKF. *Electronics* **2019**, *8*, 761. [[CrossRef](#)]
27. Wu, T.F.; Huang, H.C.; Chien, Y.-R. Development of an field-programmable gate arrays-based three-wheeled omnidirectional sensor mobile robot for the teaching of embedded robotics. *Sens. Lett.* **2013**, *11*, 2145–2148. [[CrossRef](#)]
28. Cavuslu, M.A.; Karakuzu, C.; Karakaya, F. Neural identification of dynamic systems on FPGA with improved PSO learning. *Appl. Soft Comput.* **2012**, *12*, 2707–2718. [[CrossRef](#)]
29. Zhu, W.H.; Lamarche, T.; Dupuis, E.; Jameux, D.; Barnard, P.; Liu, G. Precision control of modular robot manipulators: The VDC approach with embedded FPGA. *IEEE Trans. Robot.* **2013**, *29*, 1162–1179. [[CrossRef](#)]
30. Gulbudak, O.; Santi, E. FPGA-based model predictive controller for direct matrix converter. *IEEE Trans. Ind. Electron.* **2016**, *63*, 4560–4570. [[CrossRef](#)]
31. Pantel, L.; Wolf, L.C. On the suitability of dead reckoning schemes for games. In Proceedings of the 1st Workshop on Network and System Support for Games, Braunschweig, Germany, 16–17 April 2002; pp. 79–84.
32. Jirawimut, R.; Ptasincki, P.; Garaj, V.; Cecelja, F.; Balachandran, W. A method for dead reckoning parameter correction in pedestrian navigation system. *IEEE Trans. Instrum. Meas.* **2003**, *52*, 209–215. [[CrossRef](#)]
33. Wang, Y.; Zhao, H. Improved smartphone-based indoor pedestrian dead reckoning assisted by visible light positioning. *IEEE Sens. J.* **2019**, *19*, 2902–2908. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).