出國報告（出國類別：國際會議）

參加 IEEE SMC 2011 國際研討會心得報告

服務機關：國立宜蘭大學電機系
姓名職稱：黃旭志助理教授
派赴國家：美國
出國期間：100/10/4~100/10/14
報告日期：100/11/30

目次

一、摘要

本次心得報告針對參加在美國所舉辦的IEEE SMC 2011國際研討會做介紹。從參加此研討會的目的，過程及心得與建議做報告。

二、目的

藉由參與這種大型國際研討會會議，促進國際交流。

三、過程

2011 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2011)是由IEEE 舉辦的研討會。研討會的地點在美國阿拉斯加，這個大型研討會所刊登論文的領域非常廣，也有各種不同的領域的報告場次。在研討會中，大會除了安排各個場次的專題報告外，更有國外大師級的精彩演講。除了可以和每個作者討論外，更能在最短的時間，看到目前國外最新的研發成果。而在會後也有機會和國內外的學者共同討論，除了促進交流之外，也多認識了各個單位的專家學者，參加這次 IEEE SMC 2011 國際研討會，收穫非常豐富。未來也希望能夠多多的參加這類的大型研討會。

四、心得及建議

2011 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2011)是非常大型的國際研討會，在為期四天的研討會中，大會除了安排大師級的專業演講外，更有將近 50 個特別議程，包括很多領域，Systems Science & Engineering，Human-Machine Systems 和 Cybernetics。這次遠赴阿拉斯加參加這次的研討會，收穫非常多，也打開了國際視野，認識了國內外專家學者，共同參與此盛會，並進行討論與交流。然而這種大型的國際會議，註冊費比較高，加上旅費的花費，政府應增加補助，鼓勵發表研討會論文，增加台灣國際能見度。

Intelligent Motion Controller Design for Four-Wheeled Omnidirectional Mobile Robots Using Hybrid GA-PSO Algorithm

Hsu-Chih Huang
Department of Electrical Engineering
National Ilan University.
Ilan, Taiwan
vinus_huang@yahoo.com.tw

***Abstract*—This paper presents an intelligent motion controller based on genetic algorithm (GA)-particle swarm optimization (PSO) hybrid metaheuristic algorithm for four-wheeled omnidirectional mobile robots. The optimal parameters of motion controller are obtained by minimizing the performance index using the proposed GA-PSO hybrid algorithm. GA has been combined with PSO in evolving new solutions by applying crossover and mutation operators on solutions constructed by particles. These optimal parameters are used in the GA-PSO motion controller to obtain better performance for four-wheeled omnidirectional mobile robots to achieve both trajectory tracking and stabilization. Simulation results are conducted to show the effectiveness and merit of the proposed hybrid GA-PSO intelligent motion controller for four-wheeled omnidirectional mobile robots.**

Keywords-GA, kinematic, mobile robot, optimal control, PSO

I. INTRODUCTION

Recently, omnidirectional mobile robots have attracted much attention in the field of robotics. Comparing with several car-like robots [1-3], the type of omnidirectional mobile mechanism has the superior agile capability to move towards any position and to attain any desired orientation. Modeling and control of omnidirectional mobile robots have been investigated in [4-7]. However, these studies did not cope with the controller parameter optimization problems.

There are many methods to address the optimal problem of mobile robots [8-12]. Among these approaches, GAs [13-14] and PSOs [15-17] have been regarded as effective metaheuristic algorithms in finding optimal solutions for difficult combinatorial problems. GA was introduced by Holland [13] based on evolutionary principles and has been proven powerful in finding the optimal solution by exploiting its strong optimization ability [14]. Such ability hinges on the advantages of both deterministic and probabilistic schemes to improve solutions using simple operators, such as reproduction, crossover and mutation. On the other hand, PSO algorithm proposed by Kennedy and Eberhart [15] is a probabilistic approach to solve optimal problems. PSO algorithm has been regarded as another powerful means to solve for optimization problems. This computational paradigm and its model are directly related to the organization and behavior of animals which live in groups such as flock of birds or swarm of insects. The particle swarm optimization is a powerful and robust method, useful for global optimization problems in a wide variety of applications [15-17].

Although GA and PSO are both widely used intelligent optimization algorithms, they have their own advantages and disadvantages [18]. GA performs crossover and mutation operations to recombine chromosomes. It has strong global search capability, but its convergence speed is slow because no memory mechanism is applied. The searching experience is discarded once the population changes. PSO has much more powerful intelligent background because the knowledge of good solutions is retained by particles. It has constructive cooperation between particles, namely that particles in the swarm share their searching experiences [18]. Some GA-PSO hybrid methods [18-20] have been proposed to circumvent the problem. However, to date, no attempt has been made to using GA-PSO hybrid algorithm for developing intelligent motion controller of four-wheeled omnidirectional mobile robots.

The rest of this paper is organized as follows. In Section II, the kinematic control law is proposed to achieve stabilization and trajectory tracking for the four-wheeled omnidirectional mobile robots. Section III elaborates the GA-PSO hybrid algorithm and its application to controller parameter tuning. Section IV conducts several simulations to show the merit of the proposed method. Section V concludes this paper.

II. KINEMATIC CONTROL

This section is devoted to briefly describing the kinematic model of a four-wheeled omnidirectional mobile robot with four independent driving wheels. With the kinematic model, a kinematic controller is proposed to achieve stabilization and trajectory tracking of the mobile robot.

A. Kinematic model

The omnidirectional mobile robot is equipped with four independent driving wheels equally spaced at 90 degrees from one another. Fig. 1 depicts the structure and geometry of the four-wheeled omnidirectional driving configuration with respect to a world frame. Due to structural symmetry, the vehicle has the property that the center of geometry coincides with the center of mass.

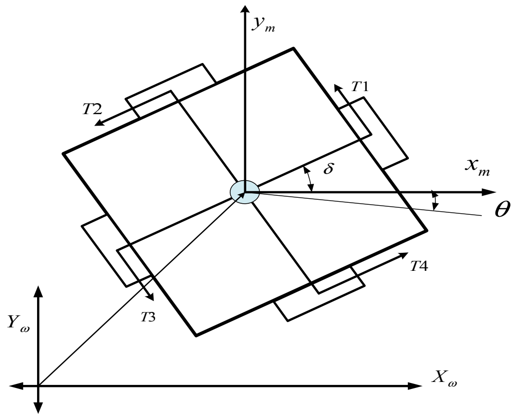

Fig. 1. Structure and geometry of the four-wheeled omnidirectional mobile robot.

In what follows describes the kinematic model of this kind of robot, where θ represents the vehicle orientation which is positive in the counterclockwise direction. Note that θ also denotes the angle between the moving frame and the world frame. On the basis of the method proposed by [12], it is easy to obtain the following inverse kinematic model of the four-wheeled omnidirectional mobile platform in the world frame.

$$\upsilon(t)=\begin{bmatrix}\upsilon_1(t)\\ \upsilon_2(t)\\ \upsilon_3(t)\\ \upsilon_4(t)\end{bmatrix}=\begin{bmatrix}r\omega_1(t)\\ r\omega_2(t)\\ r\omega_3(t)\\ r\omega_4(t)\end{bmatrix}=P(\theta(t))\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix} \tag{1}$$

where

$$P(\theta(t))=\begin{bmatrix}-\sin(\delta+\theta) & \cos(\delta+\theta) & L\\ -\cos(\delta+\theta) & -\sin(\delta+\theta) & L\\ \sin(\delta+\theta) & -\cos(\delta+\theta) & L\\ \cos(\delta+\theta) & \sin(\delta+\theta) & L\end{bmatrix} \tag{2}$$

and $\omega_i(t), i=1,2,3,4$ respectively denotes the angular velocity of each wheel; r denotes the radius of each wheel; L represents the distance from center of the platform to the center of each wheel. Note that although the matrix $P(\theta(t))$ is singular for any θ, but its left inverse matrix can be found, i.e., $P^{\#}(\theta(t))P(\theta(t))=I$, and expressed by

$$P^{\#}(\theta(t))=\begin{bmatrix}\frac{-\sin(\delta+\theta)}{2} & \frac{-\cos(\delta+\theta)}{2} & \frac{\sin(\delta+\theta)}{2} & \frac{\cos(\delta+\theta)}{2}\\ \frac{\cos(\delta+\theta)}{2} & \frac{-\sin(\delta+\theta)}{2} & \frac{-\cos(\delta+\theta)}{2} & \frac{\sin(\delta+\theta)}{2}\\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L}\end{bmatrix} \tag{3}$$

B. Kinematic control

With the kinematic model in (1), this subsection is devoted to designing two kinematic controllers to achieve point-to-point stabilization and trajectory tracking for the omnidirectional mobile robot in Fig. 1. Furthermore, a unified nonlinear control approach is also presented as below.

B.1 Point stabilization

The control goal of the point stabilization is to find the controlled angular velocity vector $[\omega_1(t)\quad \omega_2(t)\quad \omega_3(t)\quad \omega_4(t)]^T$ to steer the mobile robot from any starting pose $[x_0\quad y_0\quad \theta_0]^T$ to any desired destination pose $[x_d\quad y_d\quad \theta_d]^T$. Note that the current pose of the mobile robot is $[x(t)\quad y(t)\quad \theta(t)]^T$. To design the controller, one defines the pose error which is the difference between the present pose and the desired destination pose, that is,

$$\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}=\begin{bmatrix}x(t)\\ y(t)\\ \theta(t)\end{bmatrix}-\begin{bmatrix}x_d\\ y_d\\ \theta_d\end{bmatrix} \tag{4}$$

which gives

$$\begin{bmatrix}\dot{x}_e(t)\\ \dot{y}_e(t)\\ \dot{\theta}_e(t)\end{bmatrix}=\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix}=P^{\#}(\theta(t))\begin{bmatrix}r\omega_1(t)\\ r\omega_2(t)\\ r\omega_3(t)\\ r\omega_4(t)\end{bmatrix} \tag{5}$$

To asymptotically stabilize the system, the following stabilization law is proposed. Note that the matrices K_P and K_I are symmetric and positive definite, i.e., $K_P=diag\{K_{p1},K_{p2},K_{p3}\}=K_P^T>0,\ K_I=diag\{K_{i1},K_{i2},K_{i3}\}=K_I^T>0.$

$$\begin{bmatrix}\omega_1(t)\\ \omega_2(t)\\ \omega_3(t)\\ \omega_4(t)\end{bmatrix}=\frac{1}{r}P(\theta(t))\left(-K_p\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}-K_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix}\right) \tag{6}$$

Taking (6) into (5), the dynamics of the closed-loop error system becomes

$$\begin{bmatrix}\dot{x}_e(t)\\ \dot{y}_e(t)\\ \dot{\theta}_e(t)\end{bmatrix}=-K_p\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}-K_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix} \tag{7}$$

For the asymptotical stability of the closed-loop error system, a radially unbounded Lyapunov function candidate is chosen as follows:

$$V_1(t)=\frac{1}{2}[x_e(t)\quad y_e(t)\quad \theta_e(t)]\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}+\frac{1}{2}\left[\int_0^t x_e(\tau)d\tau\quad \int_0^t y_e(\tau)d\tau\quad \int_0^t \theta_e(\tau)d\tau\right]K_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix} \tag{8}$$

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

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

$$=-[x_e(t)\quad y_e(t)\quad \theta_e(t)]K_p\begin{bmatrix}x_e\\ y_e\\ \theta_e\end{bmatrix}<0 \tag{9}$$

Since $\dot{V}$ is negative semidefinite, Barbalat's lemma implies that $[x_e(t) \;\; y_e(t) \;\; \theta_e(t)]^T \to [0 \;\; 0 \;\; 0]^T$ as $t \to \infty$.

B.2 Trajectory tracking

This subsection considers the trajectory tracking problem. Unlike all nonholonomic conventional mobile robots, the trajectories of the omnidirectional mobile robots can not be generated using their kinematic models, i.e., any smooth and differentiable trajectories for the omnidirectional robots can be arbitrarily planned. Given the smooth and differentiable trajectory $[x_d(t) \;\; y_d(t) \;\; \theta_d(t)]^T \in C^1$, one defines the following tracking error vector

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

Thus, one obtains

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

Similarly, the control goal is to find the motors' angular velocities $[\omega_1(t) \;\; \omega_2(t) \;\; \omega_3(t)]^T$ such that the closed-loop error system is globally asymptotical stable. In doing so, one proposes the following trajectory tracking law such that

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

where the matrices, K_P and K_I , are symmetric and positive definite. Substituting (12) into (11) leads to the underlying closed-loop error system governed by

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

Similar to point stabilization, the Lyapunov function candidate can be chosen as equation (8), and from (9) one can easily prove that the closed-loop error system for trajectory tracking control is asymptotically stable.

Worthy of mention is that the point stabilization and trajectory tracking control problems can be simultaneously achieved by the control law (12). The control law (12) becomes a point stabilization one if the desired pose $[x_d(t) \;\; y_d(t) \;\; \theta_d(t)]^T$ can be either the time-dependent trajectory or the fixed destination posture.

III. GA-PSO HYBRID ALGORITHM FOR MOTION CONTROLLER

Both GA and PSO are population-based search algorithms to solve complex optimal problems. These metaheuristic algorithms are widely used for solving optimal problems. However, GA does not have the feedback mechanism in the system and PSO has the disadvantage of converging prematurely to a solution [18]. To circumvent this problem, PSO algorithm has been hybridized with GA operators, including crossover and mutation operations to generate new populations. In the proposed GA-PSO hybrid algorithm, PSO and GA search alternately and cooperatively in the solution space.

A. Genetic algorithm

Genetic algorithm is an adaptive and heuristic search method. Its main idea is to construct a fitness according to the objective function to evaluate all chromosomes in a population. After a certain number of iterations, a chromosome which has the best fitness is chosen as the solution of the specified problem. This algorithm starts with a set of randomly selected chromosomes as the initial population and the two genetic operators, crossover and mutation creates new chromosomes (offsprings). The selection operator is used to create populations from generation to generation. The chromosomes with better fitness values have higher probabilities to be selected in the next generation. In what follows, the genetic operations of GA are introduced.

A.1 Reproduction (Selection)

The primary objective of the reproduction is to duplicate good solutions and eliminate bad solutions in a population, thus keeping the population size constant. This operation is applied to select individuals from the population so that these chosen individuals can be sent to the crossover and mutation modules in order to attain new offsprings. The selection policy is ultimately responsible for ensuring survival of the best fitted individuals.

A.2 Crossover

Crossover is the fundamental mechanism of genetic rearrangement and is applied next to the selection in GA. The crossover site is randomly determined and some portions of the strings are exchanged between the two solutions to create new solutions.

A.3 Mutation

Mutation is a process that consists of making small alterations to the bits of the chromosomes by applying some kind of randomized changes. This operator is necessary for maintaining certain diversity in the population, thus avoiding quick convergence to a local minimum.

A.4 Fitness function

The fitness function is application-specific and is always designed according to the problem to be optimized. The fitness of new chromosomes from genetic operations should be evaluated based on the fitness function.

B. Particle swarm optimization

The PSO algorithm is a population based optimization method. This algorithm can be applied to solve for the optimal problems with the multimodal function $f(x) = f(x_1, x_2, \ldots x_n)$ by using a population of particles. The fitness of each particle is given by the fitness function $f(x)$. Each particle represents a solution of the optimal problem. Fig. 2 depicts the geometrical illustration of PSO algorithm, the particles are moving through a multidimensional search space, where the

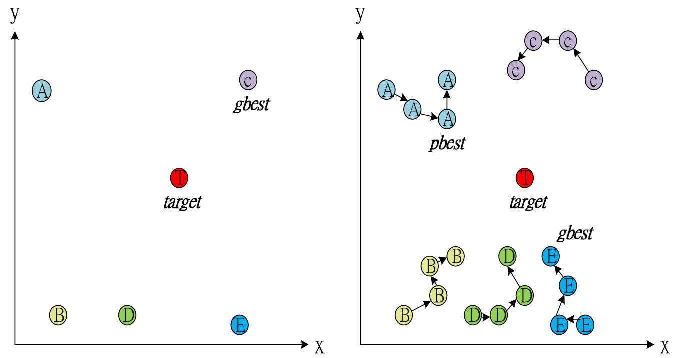

Fig. 2. Geometrical illustration of the PSO algorithm.

position of each particle is adjusted according to its own experience and that of its neighbors.

In PSO algorithm, $x_i(t)$ denote the position of particle *i* in the search space at discrete time steps *t*. The position of the particle is changed by adding a velocity, $v_i(t)$ to the current position, given by

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (14)$$

with a 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. The experimental knowledge of a particle is generally referred as the cognitive component that is proportional to the distance of the particle from its own best position found since the first time step. The socially exchanged information is referred to as the social component of the velocity equation.

In the beginning, each particle is randomly set in a random position and traverses the multidimensional space with a random velocity. The agent is free to wander inside the pre-defined *n*-dimensional space, within the constraints imposed by the boundary conditions that fix the search space of the algorithm. Hence, the value of the parameters during the optimization process is determined. The agents modify their trajectories following two types of stochastic attractions. The best position reached by the single particle (*gbest*) is responsible for the former type of attraction, also referred to as the "cognitive rate", since it determines how much the agent is affected by the memory of the best place he has found. On the other hand, the best location found by the rest of the swarm (*pbest*) is the second factor, also termed the "social rate", which indicates the influence of the swarm on the single particle. Each particle *i* move around the search space, and renew its velocity using its past experience (personal best) and the population's experience (global best) given by Shi and Eberhart [18]

$$v_{ij}(t+1) = wv_{ij}(t) + c_1\varphi_1(p_{best} - x_{ij}(t)) + c_2\varphi_2(g_{best} - x_{ij}(t)) \quad (15)$$

where $v_{ij}(t)$ is the velocity of particle *i* in dimension *j* at time step *t*, $x_{ij}(t)$ is the position of particle *i* in dimension *j* at time step *t*, c_1 and c_2 are positive acceleration constants used to respectively scale the contribution of the cognitive and social components. φ_1 and φ_2 are uniform random numbers with the range [0,1]. These two random values introduce a stochastic element to the PSO algorithm. w is called inertia weight which control the momentum of the particle by weighting the contribution of the previous velocity, namely that it controls how much memory of the previous flight direction will influence the new velocity.

C. Application GA-PSO to motion controller design

Although the kinematic controller for four-wheeled omnidirectional mobile robot was synthesized in (12), the two control matrices K_P and K_I were not optimally chosen to obtain optimal performance. This subsection aims to employ the GA-PSO hybrid algorithm to design an optimal motion controller for omnidirectional mobile robots. The control parameters $K_P = diag\{k_{p1}, k_{p2}, k_{p3}\}$ and $K_I = diag\{k_{i1}, k_{i2}, k_{i3}\}$ in (12) are optimized via GA-PSO hybrid algorithm to achieve trajectory tracking and stabilization for mobile robots. Each PSO particle is composed of the sequence k_{p1}, k_{p2}, k_{p3}, k_{i1}, k_{i2}, k_{i3} defined in Section 2. The optimal configuration of the mobile robot k_{p1}, k_{p2}, k_{p3}, k_{i1}, k_{i2}, and k_{i3} will be evolved by the efficient GA-PSO hybrid algorithm. Note that the performance index of the GA-PSO motion controller is integral square error (ISE). Each particle represents a solution set $k_{p1}, k_{p2}, k_{p3}, k_{i1}, k_{i2}, k_{i3}$. The GA-PSO algorithm for motion controller design is described by the following steps.

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

Step 2: (1) Randomly generates particles.
(2) Initialize the position and velocity of the particles.

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

Step 4. GA 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 (15).
(2)Update the position using (14).

Step 7. Check the stop criterion. If the stop criterion is not matched, go to Step 3 and set $t = t+1$, otherwise, output the optimal path and its corresponding controller parameter $k_{p1}, k_{p2}, k_{p3}, k_{i1}, k_{i2}, k_{i3}$ and stop the algorithm.

IV. SIMULATION RESULTS AND DISCUSSION

The aims of the simulations are to examine the effectiveness and performance of the proposed GA-PSO control law (12) to the omnidirectional mobile platform. These simulations are performed with the parameters: $w = 0.7$, $c1 = c2 = 1.2$. In the simulations, GA-PSO algorithm was coded using Matlab, and the fitness function was given by ISE. The crossover probability is 0.6, and the mutation probability is 0.1.

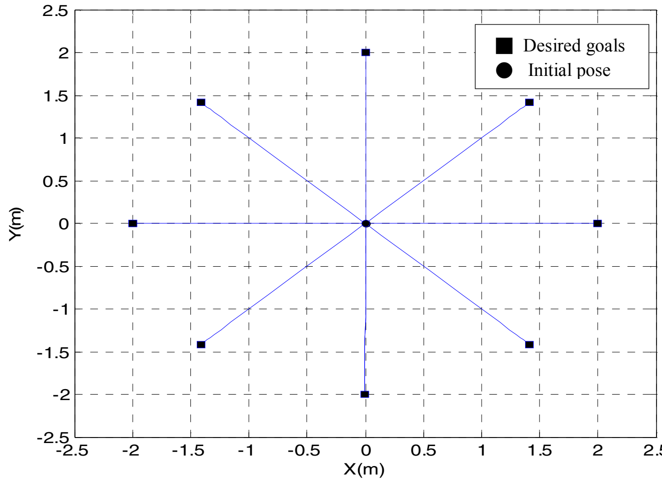

Fig. 3. Simulated trajectories of the proposed GA-PSO motion controller for achieving point stabilization.

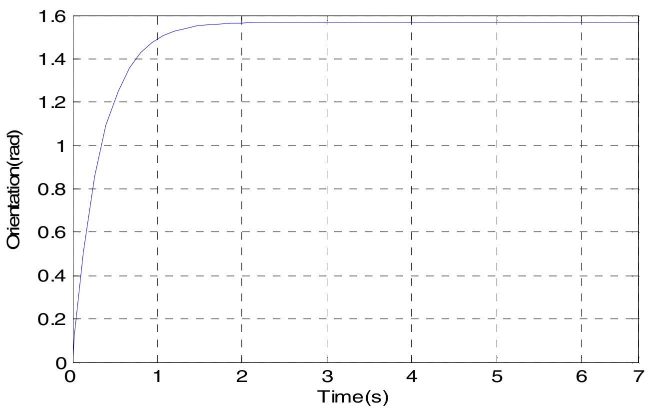

Fig. 4. Illustration of the orientation behavior moving towards the desired orientation of $\pi/2$ in the case n=1.

A. Point stabilization

The first simulation was conducted to investigate the regulation performance of the proposed GA-PSO control law (12). The initial pose of the omnidirectional mobile platform was assumed at the origin, i.e., $[x_0 \quad y_0 \quad \theta_0]=[0\text{ m} \quad 0\text{ m} \quad 0\,\text{rad}]$, and the desired final 8 goal postures are located on the unit circle, given by $\left[2\cos(\frac{n\pi}{4})\text{ m} \quad 2\sin(\frac{n\pi}{4})\text{ m} \quad \frac{\pi}{2}\text{ rad}\right]^T$, $n=0,1,...,7$. Fig. 3 depicts all the simulated trajectories of the omnidirectional mobile robot from the origin to the goal poses, and Fig. 4 shows the heading behavior of the proposed stabilization law for the platform moving towards the desired orientation $\pi/2$ in the case n=1. Through simultion results, the mobile robot with the proposed GA-PSO stabilization method has been shown capable of reaching the desired postures. Fig. 5 presents the total cost of the GA-PSO controller to achieve stabilization.

B. Elliptic trajectory tracking

The elliptic trajectory tracking simulation is aimed to explore how the proposed controller (12) steers the mobile platform to exactly track an elliptic trajectory described

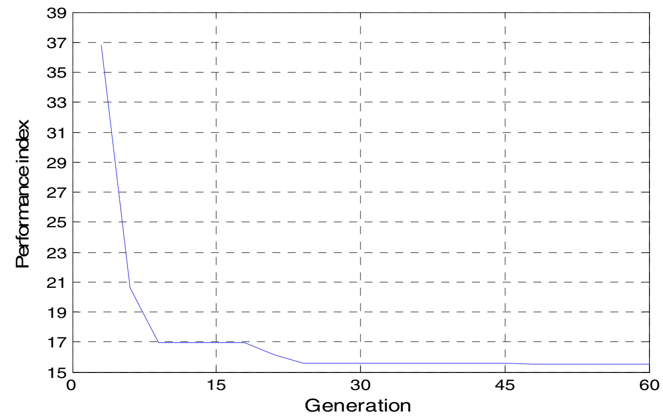

Fig. 5. Performance index of the proposed GA-PSO controller to achieve stabilization.

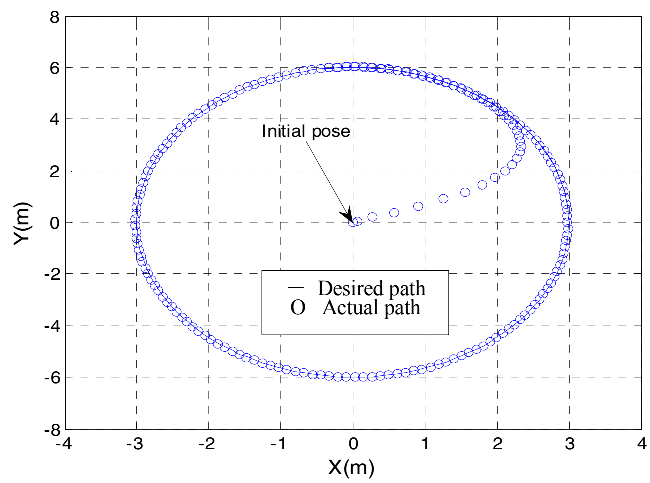

Fig. 6. Simulation result of the elliptic trajectory tracking.

Fig. 7. Performance index of the proposed GA-PSO controller to achieve trajectory tracking.

by $[x_r \quad y_r \quad \theta_r]=[3\cos w_r t(\text{m}) \quad 6\sin w_r t(\text{m}) \quad 0(\text{rad})], w_r=0.2$ rad/sec. The simulation assumed that the platform got started at $[x_0 \quad y_0 \quad \theta_0]=[0\text{m} \quad 0\text{m} \quad 0\,\text{rad}]$. Fig. 6 presents the simulation result for elliptic trajectory tracking of the mobile robot. Fig. 7 presents the total cost of the proposed GA-PSO controller to achieve trajectory tracking. These results indicate that the proposed GA-PSO kinematic controller (12) is capable of successfully steering the four-wheeled omnidirectional mobile robot to track the elliptic trajectory.

V. CONCLUSION

This paper has presented an optimal motion controller based on GA-PSO hybrid algorithm for four-wheeled omnidirectional mobile robots to achieve both trajectory tracking and stabilization. Based on the kinematic model, the optimial motion controller has been synthesized via the hybrid GA-PSO algorithm to achieve both trajectory tracking and stabilization. Through simulation results, the proposed GA-PSO motion control method has been shown to achieve stabilization and trajectory tracking for the four-wheeled omnidirectional mobilr robots.

ACKNOWLEDGMENT

The authors gratefully acknowledge financial support from the National Science Council, Taiwan, R.O.C., under grant NSC99-2628-E-241-003.

REFERENCES

[1] Z. P. Jiang and H. Nijmeijer, "A recursive technique for tracking control of nonholonomic systems in chained form," *IEEE Transactions on Automatic Control,* vol. 44, no. 2, pp. 265-279, February 1999.

[2] T. C. Lee, K. T. Song, C. H. Lee and C. C. Teng, "Tracking control of unicycle-modeled mobile robots using a saturation feedback controller," *IEEE Transactions on Control Systems Technology,* vol. 9, no. 2, pp. 305-318, March 2001.

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

[4] F. G. Pin and S. M. Killough, "A new family of omnidirectional and holonomic wheeled platforms for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 480-489, August 1994.

[5] M. J. Jung, H. S. Kim, S. Kim and J. H. Kim, "Omnidirectional mobile base OK-II," *Proceeding of the 2000 IEEE International Conference on Robotics and Automation*, pp. 3449-3454, San Francisco, April 2000.

[6] K. Watanabe, Y. Shiraishi, S. Tzafestas, J. Tang and T. Fukuda, "Feedback control of an omnidirectional autonomous platform for mobile service robots," *Journal of Intelligent and Robotic Systems*, vol. 22, pp. 315-330, 1998.

[7] H. C. Huang and C. C. Tsai, "FPGA implementation of an embedded robust adaptive controller for autonomous omnidirectional mobile platform," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 5, pp. 1604-1616, 2009.

[8] T. W. Manikas, K. Ashenayi and R. L. Wainwright, "Genetic algorithms for autonomous robot navigation," *IEEE Instrumentation & Measurement Magazine*, vol. 10, no. 6, pp. 26-31, December 2007.

[9] K. S. Tang, F. M. Kim, C. Guanrong and S. Kwong, "An optimal fuzzy PID controller," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 4, pp. 757-765, 2001.

[10] J. Kan, W. Li and J. Liu, "Fuzzy immune self-tuning PID controller and its simulation," *IEEE Conference on Industrial Electronics and Applications,* pp. 625-628, June 3-5, 2008.

[11] I. K. Jung, K. B. Hong, S. K. Hong and S. C. Hong, "Path planning of mobile robot using neural network," *Proceedings of the IEEE International Symposium on Industrial Electronics,* vol. 3, pp. 979-983, Slovenia 1999.

[12] T. Kalmár-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, January 2004.

[13] J. H. Holland, "Adaptation in natural and artificial systems," *University of Michigan Press*, Ann Arbor, 1975.

[14] J. C. Gallagher, S. Vigraham and G, Kramer, "A family of compact genetic algorithms for intrinsic evolvable hardware," *IEEE Transactions on Evolutionary Computing*, vol.8, no.2, pp.111-126, 2004.

[15] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, pp.1942-1948, November 1995.

[16] M. I. Solihin, W. Kamal and A. Legowo, "Optimal PID controller tuning of automatic gantry crane using PSO algorithm," *Proceeding of the 5th International Symposium on Mechatronics and its Applications (ISMA08)*, Amman, Jordan, May 27-29, 2008.

[17] X. Wang, Y. Wang, H. Zhou and X. Huai, "PSO-PID: a novel controller for AQM routers," *2006 IFIP International Conference on Wireless and Optical Communications Networks.*

[18] R. Nie and J. Yue, "A GA and particle swarm optimization based hybrid algorithm," *IEEE Congress on Evolutionary Computation,* pp. 1047-1050, 2008.

[19] S. W. Chen, J. S. Chiou, Y. C. Hu, K. Y. Wang and M. Y. Shieh, "Application of the GA-PSO with the fuzzy controller to the robot soccer," *Proceedings of SICE Annual Conference,* pp. 2083-2087, 2010.

[20] S. Du, W. Li and K. Cao, "A learning algorithm of artificial neural network based on GA-PSO," *The Sixth World Congress on Intelligent Control and Automation,* vol. 1, pp. 3633-3637, 2006.