Research Article

Pareto Design of State Feedback Tracking Control of a Biped Robot via Multiobjective PSO in Comparison with Sigma Method and Genetic Algorithms: Modified NSGAI and MATLAB's Toolbox

M. J. Mahmoodabadi,¹ M. Taherkhorsandi,² and A. Bagheri³

¹ Department of Mechanical Engineering, Sirjan University of Technology, Sirjan, Iran

² Department of Mechanical Engineering, University of Texas at San Antonio, San Antonio, TX 78249, USA

³ Department of Mechanical Engineering, Faculty of Engineering, The University of Guilan, Rasht, Iran

Correspondence should be addressed to M. Taherkhorsandi; m.taherkhorsandi@gmail.com

Received 20 August 2013; Accepted 3 November 2013; Published 27 January 2014

Academic Editors: L. D. S. Coelho, K. Dincer, and M. H. Ghayesh

Copyright © 2014 M. J. Mahmoodabadi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An optimal robust state feedback tracking controller is introduced to control a biped robot. In the literature, the parameters of the controller are usually determined by a tedious trial and error process. To eliminate this process and design the parameters of the proposed controller, the multiobjective evolutionary algorithms, that is, the proposed method, modified NSGAI, Sigma method, and MATLAB's Toolbox MOGA, are employed in this study. Among the used evolutionary optimization algorithms to design the controller for biped robots, the proposed method operates better in the aspect of designing the controller since it provides ample opportunities for designers to choose the most appropriate point based upon the design criteria. Three points are chosen from the nondominated solutions of the obtained Pareto front based on two conflicting objective functions, that is, the normalized summation of angle errors and normalized summation of control effort. Obtained results elucidate the efficiency of the proposed controller in order to control a biped robot.

1. Introduction

Biped robots are one of crucial kinds of mobile robots since they are the most similar robots to humans and have the capabilities the same as humans, such as walking, speaking, and communicating [1–3]. Undeniably, they will be utilized in industry as an alternative to the skilled workforce doing high-risk activities in the near future. To control robots for a variety of challenging tasks, researchers have used efficient robust controllers, such as practical velocity tracking control [4], dynamic state feedback control [5], fuzzy PD control [6], and neural network control [7]. Specifically, Solís-Perales and Peón-Escalante [8] used robust adaptive tracking control for a class of robot manipulators having model uncertainties. Indeed, they utilized a linearizing-like control feedback and a high-gain estimator for a model

with four unknown parameters, that is, system parameters, nonlinear terms, external perturbations, and the friction effects in each robot joint. Akbari et al. [9] employed a fuzzy TSK controller to control a rotary flexible joint manipulator modeled by the use of a solenoid nonlinear spring. It has been illustrated that state feedback controllers are effectual controllers in the aspect of having acceptable tracking error and control effort. In particular, Montagner and Ribas [10] used state feedback control for tracking sinusoidal references to reject disturbances affecting the plant. They utilized three techniques, linear quadratic regulator, the pole placement, and the H_∞ control to control uninterruptible power supply systems. Chang and Fu [11] proposed dynamic state feedback formation control for achieving the realization of the multirobot formation system with respect to the problem of dilation of a formation shape and stabilization issue in

a nonholonomic system simultaneously. Oliveira et al. [12] proposed a state feedback technique based on the modification of states transition matrix and used Genetic Algorithm (GA) to eliminate the trial and error process. Solihin et al. [13] utilized state feedback control for tracking control of a flexible link manipulator. They also employed the adopting particle swarm optimization algorithm to omit the tedious trial and error process of determining heuristic parameters in state feedback control.

Particle swarm optimization (PSO) is regarded as a smart optimization evolutionary and simulating algorithm introduced by Kennedy and Eberhart [14] having outstanding qualities, such as a high convergence rate and the capability of solving complicated optimization problems. PSO was derived from human behavior and animal behavior and it is easy to implement owing to having few parameters to adjust and special characteristic of memory [15]. In that respect, it has been successfully promoted by a number of researchers and applied in a wide range of scientific fields, to name it a few, control, electronics, robotics, and economics [16–35]. Although having many excellent qualities, PSO suffers premature convergence due to losing of diversity. To solve the premature convergence, Wang et al. [36] proposed a hybrid PSO algorithm utilizing a diversity improving mechanism and neighborhood search strategies to reach a trade-off between exploration and exploitation abilities. Zhou et al. [37] employed the concept of the mutation by using random factors to augment the global search ability of particle swarm optimization. In order to enhance the search capabilities of PSO, it has been combined with other optimization algorithms. Idoumghar et al. [38] utilized particle swarm optimization combined with simulated annealing algorithms to avoid local optimal solutions and premature convergence of PSO. Qian et al. [39] combined particle swarm optimization with the simplex method to adopt a hierarchical and cooperative regime of global search and local search to optimize the objective function. Liu and Yang [40] utilized particle swarm optimization combined with the Nelder-Mead simplex method to enhance its potential for rapid convergence.

Undeniably, the objective functions of practical engineering problems conflict with each other. Hence, designers prefer to use multiobjective optimization algorithms to regard all objective functions based on the design criteria. To this end, several approaches, such as dynamic neighborhood PSO [41], dominated tree [42], sigma method [43], vector evaluated PSO [44], dynamic multiple swarms [45], dynamic population size, and adaptive local archives [46], have been proposed to develop the PSO algorithm to deal with multi-objective optimization problems. By comparing the above-mentioned techniques, it can be concluded that the main difference among those approaches is the leader selection techniques. When all particles are updated in the current iteration, some of nondominated solutions are similar in the objective function space. Keeping all of them in the archive needs a great deal of space. Moreover, it will also preclude uniform diversity of nondominated solutions. To overcome these drawbacks, a fuzzy approach is utilized in this study.

Recently, Hassanzadeh and Mobayen [47] utilized the genetic algorithm, particle swarm optimization, and ant colony to balance the pendulum in the rotational inverted position. They demonstrated the efficiency of the proposed controller with respect to parameter variations, noise effects, and load disturbances. Wang and Guan [48] successfully utilized optimal control based on particle swarm optimization for a parallel hybrid hydraulic excavator. Gao et al. [49] used a typical fractional order PID control strategy based upon the improved multi-objective differential evolutionary algorithm for the gun control system.

This investigation develops significantly authors' previous work [50, 51] as follows: Taherkhorsandi et al. [50] used a linear quadratic tracking controller to control a biped robot stepping on a flat surface; however, a nonlinear state feedback tracking controller is used here to control the biped robot walking on slope. Optimal state feedback tracking control using a multi-objective particle swarm optimization algorithm in comparison to three prominent optimization algorithms, modified NSGAI, Sigma method, and MATLAB's Toolbox MOGA is used here to design the parameters of the proposed controller, while, sliding mode control based upon a particle swarm optimization algorithm is utilized in [51] to control the biped robot.

2. The Dynamics of the Biped Robot

In the present study, a biped robot is walking in the lateral plane on slope [50]. To model this robot, a three-link planar is used according to Figure 1. The first link represents the stance leg on the ground, the second link signifies the head, arms, and trunk, and the third link is the swing leg. In fact, those links move freely in the lateral plane. The parameters of the biped robot are obtained from Table 1 for a humanoid robot having 171 (cm) height and 74 (kg) weight [53]. The distance between two legs of the model ($2d_2$) equals 32.7 (cm).

To obtain the dynamic equations of the biped robot, Newton-Euler method is employed to derive the dynamic equations of the model. θ_1 , θ_2 , and θ_3 are the angles between the first, second, and third links and assumed vertical line of these links, correspondingly. Hence, the equations of the model for θ_1 , θ_2 , and θ_3 are

$$\begin{aligned}
 I_1 \ddot{\theta}_1 = & u_1 - u_2 + h_1 m_1 g \sin \theta_1 \\
 & + l_1 \sin \theta_1 g (m_2 + m_3) + h_1 m_1 (-h_1 \ddot{\theta}_1) \\
 & + l_1 m_2 [-\ddot{\theta}_1 l_1 + \ddot{\theta}_2 \{d_2 \sin (\theta_2 - \theta_1) \\
 & - h_2 \cos (\theta_2 - \theta_1)\} \\
 & + \dot{\theta}_2^2 \{d_2 \cos (\theta_2 - \theta_1) + h_2 \sin (\theta_2 - \theta_1)\}] \\
 & + l_1 m_3 [-\ddot{\theta}_1 l_1 + \ddot{\theta}_2 \{2d_2 \sin (\theta_2 - \theta_1) \\
 & - \theta_3 (l_2 - h_3) \cos (\theta_3 - \theta_1)\} \\
 & + \dot{\theta}_2^2 2d_2 \cos (\theta_2 - \theta_1) \\
 & + \dot{\theta}_3^2 (l_3 - h_3) \sin (\theta_3 - \theta_1)],
 \end{aligned}$$

$$\begin{aligned}
I_2 \ddot{\theta}_2 &= u_2 - u_3 + m_2 d_2 g \cos \theta_2 \\
&+ m_2 h_2 g \sin \theta_2 + 2 m_3 d_2 g \cos \theta_2 \\
&- \ddot{\theta}_1 [\cos (\theta_2 - \theta_1) m_2 h_2 l_1 - \sin (\theta_2 - \theta_1) m_3 2 d_2 l_1 \\
&\quad - \sin (\theta_2 - \theta_1) m_2 d_2 l_1] \\
&+ \dot{\theta}_1^2 [\sin (\theta_1 - \theta_2) m_2 l_1 h_2 - \cos (\theta_2 - \theta_1) 2 m_3 d_2 l_1 \\
&\quad - \cos (\theta_2 - \theta_1) m_2 d_2 l_1] \\
&+ \ddot{\theta}_2 [-m_2 d_2^2 - 4 m_3 d_2^2 - m_2 h_2^2] \\
&+ \ddot{\theta}_3 [2 m_3 d_2 (l_3 - h_3) \sin (\theta_2 - \theta_3)] \\
&- \dot{\theta}_3^2 [2 m_3 d_2 (l_3 - h_3) \cos (\theta_2 - \theta_3)], \\
I_3 \ddot{\theta}_3 &= (l_3 - h_3) m_3 g \sin \theta_3 + u_3 \\
&- m_3 (l_3 - h_3) l_1 \cos (\theta_3 - \theta_1) \ddot{\theta}_1 \\
&+ m_3 (l_3 - h_3) l_1 \sin (\theta_1 - \theta_3) \dot{\theta}_1^2 \\
&+ 2 d_2 m_3 (l_3 - h_3) \sin (\theta_2 - \theta_3) \ddot{\theta}_2 \\
&+ 2 d_2 m_3 (l_3 - h_3) \cos (\theta_3 - \theta_2) \ddot{\theta}_2 \\
&- m_3 (l_3 - h_3)^2 \ddot{\theta}_3.
\end{aligned} \tag{1}$$

In this study, the biped robot passes two phases, double support phase (DSP) and single support phase (SSP). In DSP, both feet are on the ground; however, in SSP, the biped robot has one contact surface with the floor. The time of DSP is regarded as 20 percent of the whole time [54]. Moreover, the swing foot trajectory having the first-order continuity is generated and it would maintain the zero moment point on the inside of the support polygon. Then, the inverse kinematics is employed to acquire the desired trajectories of the joints. The desired trajectories should have first-order and second-order continuity. The first-order derivative continuity guarantees the smoothness of the joint velocity, while the second-order continuity guarantees the smoothness of the acceleration or torque on the joints.

3. Multiobjective Particle Swarm Optimization

3.1. Particle Swarm Optimization. Particle swarm optimization is a smart evolutionary and simulating algorithm motivated by the simulation of social behavior instead of survival of the fittest [14]. At first, PSO was proposed to tune weight functions in neural networks [55]; however, it is now utilized as an effectual optimization algorithm where the decision variables are real numbers [56, 57]. The candidates for solutions are named particles and the position of them changes based on every particle's experience and neighbors (velocity). Indeed, each candidate solution is associated with

FIGURE 1: The parameters of the robot based on the anthropometric table.

a velocity [58]. The governing equations for particles are as follows:

$$\begin{aligned}
\vec{x}_i(t+1) &= \vec{x}_i(t) + \vec{v}_i(t+1), \\
\vec{v}_i(t+1) &= W \vec{v}_i(t) + C_1 r_1 (\vec{x}_{pbest_i} - \vec{x}_i(t)) \\
&\quad + C_2 r_2 (\vec{x}_{gbest} - \vec{x}_i(t)),
\end{aligned} \tag{2}$$

where $\vec{x}_i(t)$ is the position of particle i and $\vec{v}_i(t)$ presents the velocity of particle i , at time step t . W is the inertia weight utilized to control the impact of the previous history of velocities on the current velocity of a given particle. C_1 is the cognitive learning factor illustrating the attraction which a particle has toward its own success. C_2 is the social learning factor representing the attraction which a particle has toward the success of the entire swarm. $r_1, r_2 \in [0, 1]$ are random values. $\vec{x}_{gbest}$ is the position of the best particle of the entire swarm and $\vec{x}_{pbest_i}$ is the best personal position of the particle i . Inertia weight is used to tune the global and local search ability and it has qualities which are reminiscent of the temperature parameter in the simulated annealing [58]. It is crucial to note that the large inertia weight makes a global search straightforward; however, the small inertia weight facilitates a local search. In that respect, by changing the inertia weight dynamically, the search ability is dynamically adjusted. Eberhart and Kennedy [59] illustrated that decreasing inertia weight linearly over the iterations enhances the performance of PSO. Particles are permitted to move around their best personal position ($\vec{x}_{pbest_i}$) by using a large value of C_1 and a small value of C_2 . Moreover, particles

TABLE 1: The anthropometric parameters of the model.

	First link	Second link	Third link	Unit
Mass	$m_1 = 13.75$	$m_2 = 46.5$	$m_3 = 13.75$	kg
Inertia	$I_1 = 1.4$	$I_2 = 3.25$	$I_3 = 1.4$	Kg.m ²
Length	$l_1 = 0.91$	$l_2 = 0.8$	$l_3 = 0.91$	m
CG	$h_1 = 0.50$	$h_2 = 0.27$	$h_3 = 0.50$	m

converge to the best particle of the whole swarm ($\vec{x}_{gbest}$) by utilizing a small value of C_1 and a large value of C_2 . By regarding the above-mentioned results, it was observed that best solutions were acquired when C_1 is decreased linearly and C_2 is increased linearly over the iterations [57]. In this study, the linear formulation for inertia weight and learning factors are employed as follows:

$$\begin{aligned} W &= W_1 - (W_1 - W_2) \times \left(\frac{t}{\text{maximum iteration}} \right), \\ C_1 &= C_{1i} - (C_{1i} - C_{1f}) \times \left(\frac{t}{\text{maximum iteration}} \right), \\ C_2 &= C_{2i} - (C_{2i} - C_{2f}) \times \left(\frac{t}{\text{maximum iteration}} \right), \end{aligned} \quad (3)$$

where W_1 and W_2 are the initial and final values of the inertia weight, respectively. C_{1i} and C_{2i} are the initial values of the learning factors C_1 and C_2 , respectively. C_{1f} and C_{2f} are the final values of the learning factors C_1 and C_2 , respectively. t is the current iteration number and maximum iteration is the maximum number of allowable iterations.

3.2. Multiobjective Particle Swarm Optimization. Multi-objective optimization is gaining a vector of decision variables satisfying constraints to give acceptable values to all objective functions [60]. It involves the vector of design variables and the vector of objective functions. Multi-objective minimization based on the Pareto technique can be conducted using some definitions [61].

The definition of Pareto optimality: a point $X^* \in \Omega$ (Ω is a feasible region in R^n) is Pareto optimal (minimal) if and only if there is not $X \in \Omega$ which is dominant over X^* . Alternatively, it can be readily restated as $\forall X \in \Omega, X \neq X^*, \exists i \in \{1, 2, \dots, k\} : f_i(X^*) < f_i(X)$.

The definition of Pareto dominance: a vector $\vec{U} = [u_1, u_2, \dots, u_n]$, is dominant over the vector $\vec{V} = [v_1, v_2, \dots, v_n]$ (denoted by $\vec{U} < \vec{V}$) if and only if $\forall i \in \{1, 2, \dots, n\}, u_i \leq v_i \wedge \exists j \in \{1, 2, \dots, n\} : u_j < v_j$.

The definition of Pareto front: for a given multi-objective optimization problem, the Pareto front PT^* is a set of vectors of objective functions gained by utilizing the vectors of decision variables in the Pareto set P^* ; that is, $PT^* = \{F(X) = (f_1(X), f_2(X), \dots, f_k(X)) : X \in P^*\}$.

The definition of Pareto set: for a given multi-objective optimization problem, a Pareto set P^* is a set in the decision variable space consisting of all the Pareto optimal vectors $P^* = \{X \in \Omega \mid \nexists X' \in \Omega : F(X') < F(X)\}$. Indeed, the Pareto front PT^* is a set of the vectors of objective functions mapped from P^* .

In multi-objective particle swarm optimization, a set of different leaders is devoted to each particle and one of the leaders could be chosen to update the position of a particle; however, one leader is utilized to update the positions of particles in single-objective optimization problems. In elaboration, one leader should be selected as $gbest$ in order to update the position of each particle and enhance the convergence and diversity of solutions. To this end, a leader selection approach based on density measures is utilized. A neighborhood radius $R_{neighborhood}$ is defined for all nondominated solutions. Two nondominated solutions are neighbors if the Euclidean distance measured in the objective domain between them is fewer than $R_{neighborhood}$. Hence the particle having fewer neighbors is preferred as a leader. On the other hand, by choosing an appropriate approach to find $\vec{x}_{pbest_i}$ for particle i th, the diversity within the swarm is maintained. Here, Sigma method is utilized to find the best personal positions of particles; however, this method was proposed to find the best local guides [43]. When σ_i and σ_j are devoted to each particle in the population and archive, respectively, the particle k in the archive is chosen as the best personal position of the particle i where the distance between σ_i and σ_k is minimized. By regarding a two-objective space, the parameter σ is defined as follows:

$$\sigma = \frac{f_1^2 - f_2^2}{f_1^2 + f_2^2}. \quad (4)$$

In the present study, a turbulence operator is employed to find more appropriate positions and avoid being trapped in a local minimum. N particles in the population are randomly chosen to add the turbulence factor to their position vectors:

$$\vec{x}_i(t) = \vec{x}_{min}(t) + \text{rand} \times (\vec{x}_{max} - \vec{x}_{min}), \quad (5)$$

where rand is a random number generated uniformly in the interval $[0, 1]$. $\vec{x}_{max}$ and $\vec{x}_{min}$ are upper bound and lower bound of the search space. In this paper, $N = P_m \times \text{number of particles}$ where P_m is the probability of the turbulence operator and set at $5/t$.

4. The Pareto Design of State Feedback Control

The stages of state feedback control are designed and constructed step by step as follows. To control the system, the

state variable vector is chosen as $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \\ \theta_3 \\ \dot{\theta}_3 \end{pmatrix}$. And the

errors could be defined as

$$\begin{aligned} E_p &= \theta_{dp} - \theta_p \quad (p = 1, 2, 3), \\ \dot{E}_q &= \dot{\theta}_{dq} - \dot{\theta}_q \quad (q = 1, 2, 3). \end{aligned} \quad (6)$$

Finally, the control effort is obtained by

$$u = \sum_{p=1}^3 K_p E_p + \sum_{q=1}^3 K'_q \dot{E}_q, \quad (7)$$

TABLE 2: The objective functions and their associated design variables for the optimum points of Figure 2.

Optimum design point	A	B	C
Normalized summation of angles errors	1.22×10^{-2}	4.14×10^{-1}	8.23×10^{-1}
Normalized summation of control efforts	9.67×10^{-1}	5.27×10^{-1}	1.09×10^{-1}
Design variable K_1	-1.80×10^2	-8.33×10^1	-8.44×10^1
Design variable K_2	-9.49×10^3	-8.23×10^3	-6.64×10^3
Design variable K_3	-1.90×10^2	-1.11×10^2	-1.09×10^2
Design variable K'_1	-9.39×10^3	-5.00×10^3	-4.89×10^3
Design variable K'_2	-1.80×10^2	-3.91×10^{-1}	-2.55×10^{-2}
Design variable K'_3	-9.00×10^2	-6.49×10^{-2}	-1.03×10^1

where K_p and K'_q (p and $q = 1, 2, 3$) are the state feedback parameters.

The proposed method is used to find the proper state feedback parameters and remove the tedious and repetitive trial and error process. Furthermore, the results are compared with three prominent algorithms. The performance of a controlled closed loop system is usually assessed by a variety of goals [62]. In this study, the normalized summation of angles errors and normalized summation of control effort are regarded as the objective functions. These objective functions are minimized at the same time. The vector $[K_1, K_2, K_3, K'_1, K'_2, K'_3]$ is the vector of selective parameters of state feedback control. These are positive constants. The normalized summation of angles errors and normalized summation of control effort are functions of this vector's components. In this regard, by choosing various amounts of the selective parameters, changes occur in the normalized summation of angles errors and normalized summation of control effort. This is an optimization problem with two objective functions (the normalized summation of angles errors and normalized summation of control effort) and six decision variables ($K_1, K_2, K_3, K'_1, K'_2, K'_3$). The regions of the selective parameters are

$$\begin{aligned}
 -200 < K_1, \quad K'_2 < 0, \quad -10000 < K_2 < -5000, \\
 -200 < K_3 < -100, \quad -10000 < K'_1 < -4000, \quad (8) \\
 -1000 < K'_3 < -10.
 \end{aligned}$$

The feasibility and efficiency of the proposed multi-objective algorithm are assessed in comparison with Sigma method [23], modified NSGAII [41], and MATLAB Toolbox MOGA. The Pareto front of this multi-objective problem is shown in Figure 2. The swarm size is 10 and the maximum iteration equals 500. The term $\tilde{v}_i(t)$ is limited to the range $[-v_{ave}, +v_{ave}]$ in which $v_{ave} = (x_{max} - x_{min})/2$. While the velocity violates this range, it will be multiplied by a random number between $[0, 1]$. $E_{constant}$ and $R_{neighborhood}$ are set at 25 and 0.02, respectively. Over iteration, the inertia weight W is linearly decreased from $W_1 = 0.9$ to $W_2 = 0.4$, C_1 is linearly decreased from $C_{1i} = 2.5$ to $C_{1f} = 0.5$, and C_2 is linearly increased from $C_{2i} = 0.5$ to $C_{2f} = 2.5$.

By regarding Figure 2, all the optimal points in the Pareto front are nondominated and could be selected to design the controller. However, it is crucial to note that selecting a better

FIGURE 2: The obtained Pareto fronts by using Sigma method [43], modified NSGAII [52], MATLAB's Toolbox MOGA, and the proposed algorithm regarding the optimal control design of the biped robot.

amount of any objective function causes a worse amount of another objective. In the Pareto front, there are three crucial points, A, B, and C. By regarding both objective functions of the normalized summation of angles errors and normalized summation of control effort, point B could be the trade-off optimum choice. Moreover, design variables and objective functions corresponding to the optimum design points A, B, and C are illustrated in Table 2. The real tracking trajectories and phase planes of the optimum design points A, B, and C are shown in Figures 3, 4, and 5.

5. Conclusions

In this study, an optimal robust state feedback controller is used to control biped robots walking in the lateral plane on slope. To this end, a biped robot is regarded and modeled

FIGURE 3: The desired trajectory of θ_1 and its tracking trajectory for the optimum design points A, B, and C shown in the Pareto front.

FIGURE 4: The desired trajectory of θ_2 and its tracking trajectory for the optimum design points A, B, and C shown in the Pareto front.

in that plane. State feedback control is employed as a robust controller to control heavy nonlinear dynamic equations of the robot. Moreover, a multi-objective particle swarm optimization algorithm is used to design the parameters of the proposed controller. In the proposed controller, effectual techniques, such as a fuzzy based approach which prunes the archive, the turbulence operator that causes particles to escape straightforwardly, and Sigma method which finds the best personal positions of particles, are used. The results elucidate that the proposed method performs effectively in designing the parameters of the controller in comparison to three well-known algorithms, modified NSGAI, Sigma

FIGURE 5: The desired trajectory of θ_3 and its tracking trajectory for the optimum design points A, B, and C shown in the Pareto front.

method, and MATLAB's Toolbox MOGA. Indeed, the proposed approach can be regarded as a promising approach to control various similar nonlinear systems, especially, biped robots. Furthermore, the normalized summation of angles errors and normalized summation of control effort are two conflicting objective functions. By using three points of the obtained Pareto front, six parameters of the controller are designed. The first point chosen in the Pareto front has the minimum normalized summation of angles errors, the third point has the minimum normalized summation of control effort, and the second point is the optimal point minimizing both objective functions simultaneously.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] A. Rahman Hafiz, F. Alnajjar, and K. Murase, "A novel bio-inspired vision system: a step toward real-time human-robot interactions," *Journal of Robotics*, vol. 2011, Article ID 943137, 9 pages, 2011.
- [2] K. Madani, D. M. Ramik, and C. Sabourin, "Multilevel cognitive machine-learning-based concept for artificial awareness: application to humanoid robot awareness using visual saliency," *Applied Computational Intelligence and Soft Computing*, vol. 2012, Article ID 354785, 11 pages, 2012.
- [3] V. Ivancevic and N. Beagley, "Brain-like functor control machine for general humanoid biodynamics," *International Journal of Mathematics and Mathematical Sciences*, vol. 2005, no. 11, pp. 1759–1779, 2005.
- [4] Z. Zhou, W. Meng, Q. Ai, Q. Liu, and X. Wu, "Practical velocity tracking control of a parallel robot based on fuzzy

- adaptive algorithm," *Advances in Mechanical Engineering*, vol. 2013, Article ID 574896, 11 pages, 2013.
- [5] C. F. Chang and L. C. Fu, "Dynamic state feedback control of robotic formation system," in *Proceeding of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 3574–3579, October 2010.
 - [6] J. Wu, W. Zhang, and S. Wang, "A two-wheeled self-balancing robot with the fuzzy PD control method," *Mathematical Problems in Engineering*, vol. 2012, Article ID 469491, 13 pages, 2012.
 - [7] P. K. Kim and S. Jung, "Experimental studies of neural network control for one-wheel mobile robot," *Journal of Control Science and Engineering*, vol. 2012, Article ID 194397, 12 pages, 2012.
 - [8] G. Solís-Perales and R. Peón-Escalante, "Robust adaptive tracking control of a class of robot manipulators with model uncertainties," *Modelling and Simulation in Engineering*, vol. 2012, Article ID 271705, 7 pages, 2012.
 - [9] M. E. Akbari, M. A. Badamchizadeh, and M. A. Poor, "Implementation of a fuzzy TSK controller for a flexible joint robot," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 279498, 21 pages, 2012.
 - [10] V. F. Montagner and S. P. Ribas, "State feedback control for tracking sinusoidal references with rejection of disturbances applied to UPS systems," in *Proceedings of the 35th Annual Conference of the IEEE Industrial Electronics Society (IECON '09)*, pp. 1764–1769, November 2009.
 - [11] C. F. Chang and L. C. Fu, "Dynamic state feedback control of robotic formation system," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 3574–3579, October 2010.
 - [12] P. S. Oliveira, L. S. Barros, and L. G. D. Q. Silveira Jr., "Genetic algorithm applied to state feedback control design," in *Proceedings of the IEEE/PES Transmission and Distribution Conference and Exposition: Latin America*, pp. 480–485, November 2010.
 - [13] M. I. Solihin, W. Wahyudi, A. Legowo, and R. Akmeliawati, "Comparison of LQR and PSO-based state feedback controller for tracking control of a flexible link manipulator," in *Proceedings of the 2nd IEEE International Conference on Information Management and Engineering (ICIME '10)*, pp. 354–358, April 2010.
 - [14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, December 1995.
 - [15] A. Khare and S. Rangnekar, "A review of particle swarm optimization and its applications in Solar Photovoltaic system," *Applied Soft Computing*, vol. 13, no. 5, pp. 2997–3006, 2013.
 - [16] X. Chen and Y. Li, "A modified PSO structure resulting in high exploration ability with convergence guaranteed," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 37, no. 5, pp. 1271–1289, 2007.
 - [17] Y. P. Chen, W. C. Peng, and M. C. Jian, "Particle swarm optimization with recombination and dynamic linkage discovery," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 37, no. 6, pp. 1460–1470, 2007.
 - [18] K. M. Christopher and K. D. Seppi, "The Kalman swarm: a new approach to particle motion in swarm optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '04)*, pp. 140–150, 2004.
 - [19] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proceeding of IEEE Congress on Evolutionary Computation*, vol. 3, pp. 1951–1957, 1999.
 - [20] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 35, no. 6, pp. 1272–1282, 2005.
 - [21] R. A. Krohling and L. Dos Santos Coelho, "Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 36, no. 6, pp. 1407–1416, 2006.
 - [22] F. van den Bergh and A. P. Engelbrecht, "A new locally convergent particle swarm optimiser," in *Proceeding of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 96–101, October 2002.
 - [23] W. J. Zhang and X. F. Xie, "DEPSO: hybrid particle swarm with differential evolution operator," in *Proceeding IEEE International Conference Systems, Man, and Cybernetics*, vol. 4, pp. 3816–3821, October 2003.
 - [24] Y. Ma, M. Zamirian, Y. Yang, Y. Xu, and J. Zhang, "Path planning for mobile objects in four-dimension based on particle swarm optimization method with penalty function," *Mathematical Problems in Engineering*, vol. 2013, Article ID 613964, 9 pages, 2013.
 - [25] A. Y. Alanis, L. J. Ricalde, C. Simetti, and F. Odone, "Neural model with particle swarm optimization Kalman learning for forecasting in smart grids," *Mathematical Problems in Engineering*, vol. 2013, Article ID 197690, 9 pages, 2013.
 - [26] Y. K. Lin, "Particle swarm optimization algorithm for unrelated parallel machine scheduling with release dates," *Mathematical Problems in Engineering*, vol. 2013, Article ID 409486, 9 pages, 2013.
 - [27] C. H. Chen and Y. P. Chen, "Convergence time analysis of particle swarm optimization based on particle interaction," *Advances in Artificial Intelligence*, vol. 2011, Article ID 204750, 7 pages, 2011.
 - [28] J. Li, J. Li, D. Niu, and Y. Wu, "A parallel adaptive particle swarm optimization algorithm for economic/environmental power dispatch," *Mathematical Problems in Engineering*, vol. 2012, Article ID 271831, 14 pages, 2012.
 - [29] S. Kok and J. A. Snyman, "A strongly interacting dynamic particle swarm optimization method," *Journal of Artificial Evolution and Applications*, vol. 2008, Article ID 126970, 9 pages, 2008.
 - [30] Y. Zhang, L. Wu, and S. Wang, "UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 705238, 9 pages, 2013.
 - [31] X. Su, W. Fang, Q. Shen, and X. Hao, "An image enhancement method using the quantum-behaved particle swarm optimization with an adaptive strategy," *Mathematical Problems in Engineering*, vol. 2013, Article ID 824787, 14 pages, 2013.
 - [32] R. Alwee, S. M. H. Shamsuddin, and R. Sallehuddin, "Hybrid support vector regression and autoregressive integrated moving average models improved by particle swarm optimization for property crime rates forecasting with economic indicators," *The Scientific World Journal*, vol. 2013, Article ID 951475, 11 pages, 2013.
 - [33] Q. Ni and J. Deng, "A new logistic dynamic particle swarm optimization algorithm based on random topology," *The Scientific World Journal*, vol. 2013, Article ID 409167, 8 pages, 2013.

- [34] K. S. Lim, Z. Ibrahim, S. Buyamin et al., "Improving vector evaluated particle swarm optimisation by incorporating non-dominated solutions," *The Scientific World Journal*, vol. 2013, Article ID 510763, 19 pages, 2013.
- [35] H. T. Yau, C. J. Lin, and Q. C. Liang, "PSO based PI controller design for a solar charger system," *The Scientific World Journal*, vol. 2013, Article ID 815280, 13 pages, 2013.
- [36] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J. S. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences*, vol. 223, pp. 119–135, 2013.
- [37] D. Zhou, X. Gao, G. Liu, C. Mei, D. Jiang, and Y. Liu, "Randomization in particle swarm optimization for global search ability," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15356–15364, 2011.
- [38] L. Idoumghar, M. Melkemi, R. Schott, and M. I. Aouad, "Hybrid PSO-SA type algorithms for multimodal function optimization and reducing energy consumption in embedded systems," *Applied Computational Intelligence and Soft Computing*, vol. 2011, Article ID 138078, 12 pages, 2011.
- [39] X. Qian, M. Cao, Z. Su, and J. Chen, "A hybrid particle swarm optimization (PSO)-simplex algorithm for damage identification of delaminated beams," *Mathematical Problems in Engineering*, vol. 2012, Article ID 607418, 11 pages, 2012.
- [40] A. Liu and M. T. Yang, "A new hybrid nelder-mead particle swarm optimization for coordination optimization of directional overcurrent relays," *Mathematical Problems in Engineering*, vol. 2012, Article ID 456047, 18 pages, 2012.
- [41] W. J. Zhang and X. F. Xie, "DEPSO: Hybrid particle swarm with differential evolution operator," in *Proceeding of the IEEE International Conference Systems, Man, and Cybernetics*, pp. 3816–3821, October 2003.
- [42] J. E. Fieldsend and S. Singh, "A multi-objective algorithm based upon particle swarm optimization and efficient data structure and turbulence," in *Proceeding of the Workshop on Computational Intelligence*, pp. 34–44, 2002.
- [43] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Proceedings of the IEEE Swarm Intelligence Symposium*, pp. 26–33, 2003.
- [44] K. E. Parsopoulos, D. K. Tasoulis, and M. N. Vrahatis, "Multi-objective optimization using parallel vector evaluated particle swarm optimization," in *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, pp. 823–828, February 2004.
- [45] G. G. Yen and W. F. Leong, "Dynamic multiple swarms in multiobjective particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 39, no. 4, pp. 890–911, 2009.
- [46] W. F. Leong and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 38, no. 5, pp. 1270–1293, 2008.
- [47] I. Hassanzadeh and S. Mobayen, "Controller design for rotary inverted pendulum system using evolutionary algorithms," *Mathematical Problems in Engineering*, vol. 2011, Article ID 572424, 17 pages, 2011.
- [48] D. Y. Wang and C. Guan, "Optimal control for a parallel hybrid hydraulic excavator using particle swarm optimization," *The Scientific World Journal*, vol. 2013, Article ID 831564, 6 pages, 2013.
- [49] Q. Gao, J. Chen, L. Wang, S. Xu, and Y. Hou, "Multi-objective optimization design of a fractional order PID controller for a gun control system," *The Scientific World Journal*, vol. 2013, Article ID 907256, 8 pages, 2013.
- [50] M. Taherkhorsandi, B. Miripour-Fard, and A. Bagheri, "Optimal tracking control of a biped robot walking in the lateral plane," in *Proceeding of the International Symposium on Innovations in Intelligent Systems and Applications (INISTA '11)*, pp. 560–564, IEEE, June 2011.
- [51] M. Alitavoli, M. Taherkhorsandi, M. J. Mahmoodabadi, A. Bagheri, and B. Miripour-Fard, "Pareto design of sliding-mode tracking control of a biped robot with aid of an innovative particle swarm optimization," in *Proceeding of the International Symposium on Innovations in Intelligent Systems and Application*, pp. 1–5, IEEE, 2012.
- [52] K. Atashkari, N. Nariman-Zadeh, M. Gölcü, A. Khalkhali, and A. Jamali, "Modelling and multi-objective optimization of a variable valve-timing spark-ignition engine using polynomial neural networks and evolutionary algorithms," *Energy Conversion and Management*, vol. 48, no. 3, pp. 1029–1041, 2007.
- [53] D. A. Winter, *Biomechanics and Motor Control of Human Movement*, Wiley, New York, NY, USA, 1990.
- [54] P. N. Mousavi, C. Nataraj, A. Bagheri, and M. A. Entezari, "Mathematical simulation of combined trajectory paths of a seven link biped robot," *Applied Mathematical Modelling*, vol. 32, no. 7, pp. 1445–1462, 2008.
- [55] R. C. Eberhart, R. Dobbins, and P. K. Simpson, *Computational Intelligence PC Tools*, Morgan Kaufmann Publishers, 1996.
- [56] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, 2002.
- [57] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, 2005.
- [58] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 611–616, Anchorage, Alaska, USA, 1998.
- [59] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, 1995.
- [60] C. A. Coello Coello, D. A. van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic, Dordrecht, The Netherlands, 2002.
- [61] A. Jamali, A. Hajiloo, and N. Nariman-zadeh, "Reliability-based robust Pareto design of linear state feedback controllers using a multi-objective uniform-diversity genetic algorithm (MUGA)," *Expert Systems with Applications*, vol. 37, no. 1, pp. 401–413, 2010.
- [62] R. Toscano, "A simple robust PI/PID controller design via numerical optimization approach," *Journal of Process Control*, vol. 15, no. 1, pp. 81–88, 2005.