Amirkabir University of Technology
(Tehran Polytechnic)
Vol. 45 - No. 2 - Fall 2013, pp. 31- 40

Amirkabir International Journal of Science & Research
(Modeling, Identification, Simulation & Control)
(AIJ-MISC)

Pareto Optimal Design Of Decoupled Sliding Mode Control Based On A New Multi-Objective Particle Swarm Optimization Algorithm

M. Bisheban[1] and M.J. Mahmoodabadi[2]*

1-School of Mechanical and Aerospace Engineering, the George Washington University, Washington DC, USA.
2-Department of Mechanical Engineering, Sirjan University of Technology, Sirjan, Iran.

Abstract

One of the most important applications of multi-objective optimization is adjusting parameters of practical engineering problems in order to produce a more desirable outcome. In this paper, the decoupled sliding mode control technique (DSMC) is employed to stabilize an inverted pendulum which is a classic example of inherently unstable systems. Furthermore, a new Multi-Objective Particle Swarm Optimization (MOPSO) algorithm is implemented for optimizing the DSMC parameters in order to decrease the normalized angle error of the pole and normalized distance error of the cart, simultaneously. The results of simulation are presented which consist of results with and without disturbances. The proposed Pareto front for the DSMC problem demonstrates that the Ingenious-MOPSO operates much better than other multi-objective evolutionary algorithms.

KEYWORDS

Decoupled Sliding Mode Control, Multi-objective Algorithm, Particle Swarm Optimization, Inverted Pendulum System.

*Corresponding Author, Email: mahmoodabadi@sirjantech.ac.ir

1. INTRODUCTION

Particle Swarm Optimization (PSO), first introduced by Kennedy and Eberhart, is one of the modern heuristic algorithms [1]. It was developed through simulation of simplified social systems, and has been found to be robust in solving nonlinear optimization problems [2]. The PSO technique can generate a high quality solution with short calculation time and a more stable convergence characteristic compared to other evolutionary methods [3,4].

In the recent years, several approaches have been proposed to extend the PSO algorithm to deal with multi-objective optimization problems. For instance, a dynamic neighborhood strategy is used to locate the Pareto front [5]. In [6], the dominated tree approach is used to select global and personal best position for each particle. In [7], The Sigma method is implemented to find global best positions. The parallel implementation of vector evaluated PSO is investigated in [8]. In [9], a diversity parameter and time variant inertia weight and acceleration coefficients are used. In [10], cluster operation is used to avoid the excessive retention of similar particles. Also, proportional distribution and jump improved operation mechanism are used to maintain the diversity and the solution searching abilities. In [11], preference order is used to design some Pareto solutions superior to others instead of dominance method when the size of the non-dominated solution set is very large and a new updating equation for the velocity is introduced. In [12], the idea of $\varepsilon_{elimination}$ is used to fix the size of archive. The influence of this method in comparison to the clustering method has been studied. In [13], an extra objective is added to the ε-dominance approach preserve some solutions which are normally lost when using original ε-dominance technique. The $\varepsilon-\text{dominance}$ approach is applied not only on PSO methods, but also on other evolutionary algorithms such as genetic algorithms. For example, this approach is used to replace the crowding distance assignment approach in NSGAII [14]. The authors previously proposed adaptive $\varepsilon_{elimination}$ to retain more leaders in the archive in the initial iterations and this increases the convergence of PSO algorithm [15]. In [16], a new fuzzy elimination technique is proposed to prune the archive of the non-dominated solutions. In this reference, the proposed algorithm is aimed at increasing the rate of convergence and the diversity of solutions in the population together by four easy techniques: (1) use of new ε_{Fuzzy} for pruning archive; (2) using Sigma method [7] for finding personal best positions of particles; (3) use a leader selection technique based on density measures; and (4) use of turbulence operator.

In the existing literature, several previous works have considered the evolutionary algorithms for control design. For an overview of evolutionary algorithm in control engineering, Reference [17] is appropriate. In particular, pole placement for designing a discrete-time regulator for the single-input single output plant [18] and for a classical observer-based feedback controller [19] was formulated as a multi-objective optimization problem and solved with genetic algorithms. More recently, PSO was used to tune the linear controller gains for optimal design of PI controllers for doubly fed induction generators driven by wind turbines [20] and for optimal design of PID controller in AVR system [21]. These works have shown that PSO is a fast and reliable tool for control optimization, and also outperforms other evolutionary algorithms.

The inverted pendulum is one of the most commonly studied systems in the control area that is a classic example of an inherently unstable system. It is an excellent test bed for learning and testing various control techniques. There are many control techniques that have been used to investigate the control behaviors of an inverted pendulum. Sliding mode control technology is also used to stabilize the inverted pendulum due to its inherent robustness with respect to plant model uncertainty and external disturbance which are unavoidable in the environment. It is very important to note that for design of the sliding mode controller, the determination of the sliding surface parameters is a significant problem. This point is very crucial for the performance of the control system [22]. This problem can be solved using evolutionary optimization techniques [23].

Sliding mode controller is a powerful robust control strategy to treat the model uncertainties and external disturbances, provided that the bounds of these uncertainties and disturbances are known [24,25]. In [26], the parameters of switching function and exponential reaching law of the sliding mode controller for an inverted pendulum system are optimized with a modified single objective particle swarm optimization. In [27], two decoupled sliding mode control configurations have been designed for a scale model of an oil platform supply ship while the single objective genetic algorithm is used for optimization. In [28], multiobjective genetic algorithm is implemented for Pareto design of decoupled sliding-mode controllers for nonlinear systems. In [29], a novel MOPSO is applied for Pareto optimal design of the decoupled sliding mode controller for an inverted pendulum system and its stability is simulated via the Java programming. The most distinguishing characteristic between [28,29] and this work is the type of the multi-objective optimization algorithm employed for Pareto design of the controller. In other words, in this work, an intelligent decoupled sliding mode control scheme based on an improved multi-objective particle swarm optimization is proposed. Using this optimization algorithm, the important parameters of the decoupled sliding mode controller are optimized to decrease the normalized angle error of the pole and normalized distance error of the cart, simultaneously. The results obtained from this study illustrate the benefits of using the Ingenious-MOPSO to optimize the decoupled sliding mode control for an inverted pendulum system.

2. Preliminaries

A. Particle Swarm Optimization

Particle swam optimization is a population-based evolutionary algorithm and is similar to other population based evolutionary algorithms. PSO is motivated by the simulation of social behavior instead of survival of the fittest [1]. Although originally adopted for balancing weights in neural networks, PSO soon became a very popular global optimizer, mainly in problems in which the decision variables are real numbers [30].

In PSO, each candidate solution is associated with a velocity [30]. The candidate solutions are called particles and the position of each particle is changed according to its own experience and that of its neighbors (velocity). It is expected that the particles will move towards better solution areas. Mathematically, the particles are manipulated according to the following equations.

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \tag{1}$$

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

where $\vec{x}_i(t)$ and $\vec{v}_i(t)$ denote the position and velocity of particle i, at time step t. $r_1, r_2 \in [0,1]$ are random values. C_1 is the cognitive learning factor and represents the attraction that a particle has toward its own success. C_2 is the social learning factor and represents the attraction that a particle has toward the success of the entire swarm. W is the inertia weight which is employed to control the impact of the previous history of velocities on the current velocity of a given particle. The personal best position of the particle i is $\vec{x}_{pbest_i}$ and $\vec{x}_{gbest}$ is the position of the best particle of the entire swarm. Inertia weight is used to balance the global and local search ability.

The inertia weight has characteristics that are reminiscent of the temperature parameter in the simulated annealing [3]. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search. By changing the inertia weight dynamically, the search ability is dynamically adjusted. Experimental results indicated that the linearly decreasing inertia weight over the iterations improve the performance of PSO [31]. With a large value of C_1 and a small value of C_2, particles are allowed to move around their personal best position ($\vec{x}_{pbest_i}$). With a small value of C_1 and a large value of C_2, particles converge to the best particle of the entire swarm ($\vec{x}_{gbest}$). From the results, it was observed that best solutions were determined when C_1 is linearly decreased and C_2 is linearly increased over the iterations [32]. Hence, in this paper, the following linear formulation for inertia weight and learning factors are used.

$$W = W_1 - (W_1 - W_2) \times (\frac{t}{maximum\ iteration}) \tag{3}$$

$$C_1 = C_{1i} - \left(C_{1i} - C_{1f}\right) \times (\frac{t}{maximum\ iteration}) \tag{4}$$

$$C_2 = C_{2i} - \left(C_{2i} - C_{2f}\right) \times (\frac{t}{maximum\ iteration}) \tag{5}$$

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.

B. Multi-Objective Optimization Definitions

Multi-objective optimization which is also called multi-criteria optimization or vector optimization has been defined as finding a vector of decision variables satisfying constraints to give acceptable values to all objective functions [33]. It could be introduced as find X^* to optimize $F(X)$, therefore; is the vector of design variables and is the vector of objective functions. Such multi-objective minimization based on Pareto approach can be conducted using some definitions [34]:

- **Definition of Pareto dominance:** A vector $\vec{U} = [u_1, u_2, \dots, u_n]$, is dominance to vector $\vec{V} = [v_1, v_2, \dots, v_n]$ (denoted by $\vec{U} \prec \vec{V}$) if and only if

$$\forall i \in \{1,2,\dots,n\}, u_i \le v_i \wedge \exists j \in \{1,2,\dots,n\} : u_j < v_j$$

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

- **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 $P^* = \{X \in \Omega \,|\, \nexists X' \in \Omega: F(X') \prec \mathrm{F(X)}\}$.

- **Definition of Pareto front:** For a given multi-objective optimization problem, the Pareto front PT^* is a set of vectors of objective functions which are obtained using the vectors of decision variables in the Pareto set P^*, that is $PT^* = \{F(X) = (f_1(X), f_2(X), \ldots, f_k(X)): X \in P^*\}$.

C. Sliding And Decoupled Sliding Mode Control

Sliding mode control has been widely applied to the robust control of nonlinear systems [35]. In this section, the general concepts of sliding mode control for a second order dynamic system is discussed. Suppose a non-linear system is defined by the general state space equation:

$$\dot{x} = f(x, u, t) \tag{6}$$

where $x \epsilon R^n$ is the state vector, $u \epsilon R^m$ is the input vector, n is the order of the system and m is the number of inputs. Then the sliding surface $s(e,t)$ is given by:

$$s(e,t)=\{e: H^T e = 0\} \tag{7}$$

where $H \epsilon R^n$ represents the coefficients or slope of the sliding surface. Here,

$$e = x - x_d \tag{8}$$

is the negative tracking error vector.

Usually a time-varying sliding surface $s(t)$ is simply defined in the state-space R^n by the scalar equation, given by

$$s(e,t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} e = 0 \tag{9}$$

where λ is a strictly positive constant that can also be explained as the slope of the sliding surface. For instant, if $n = 2$ (for a second order system)

$$s = \dot{e} + \lambda e \tag{10}$$

And hence, s is a weighted sum of the position and velocity error. The n^{th}-order tracking problem is now being replaced by a first-order stabilization problem in which the scalar s is to be kept at zero by a governing reaching condition. By choosing Lyapunov function $V(x) = \frac{1}{2}s^2$, the following equation can guarantee the reaching condition [36].

$$\dot{V}(x) = s\dot{s} < 0 \tag{11}$$

It can be seen that the sliding mode of the system response would be chatter along the $s = 0$. From Equation (11), the existence and convergence condition can be re-written as:

$$s\dot{s} \leq -\eta\,|\,s\,| \tag{12}$$

This equation permits a non-switching region. Here, η is strictly positive constant, its value is usually chosen based on some knowledge of disturbances or system dynamics in terms of some known amplitudes.

In this control method, by changing control law according to certain predefined rules which depend on the position of the error states of the system with respect to sliding surfaces, the states are switched between stable and unstable trajectories until they reach the sliding surface.

It can be shown that the sliding condition of Equation (11) is always satisfied by:

$$u = u_{eq} - k\, sgn(s) \tag{13}$$

where u_{eq} is called equivalent control input which is obtained by $\dot{s} = 0$ and k is a design parameter or a function of $x(t)$ such that $k = k(x)$ and $k \geq \eta$.

The discontinuity in control law is the result of sgn function. This function makes the high frequency chattering in the control command. By properly defining a thin boundary layer around the sliding surface as:

$$B(t) = \{x; |s(x)| \leq \Phi\} \tag{14}$$

and therefore smoothing the discontinuity in control law across the boundary layer, the chattering can be eliminated. Linear interpolation across the boundary layer is one method for removing the discontinuous portion of control law. This is accomplished by defining a boundary layer of thickness Φ and replacing function sgn with the function sat defined as follow and shown in Figure 1.

$$sat\left(\frac{s}{\Phi}\right) = \begin{cases} sgn\left(\frac{s}{\Phi}\right) & if\ \left|\frac{s}{\Phi}\right| \geq 1 \\ \left(\frac{s}{\Phi}\right) & if\ \left|\frac{s}{\Phi}\right| < 1 \end{cases} \tag{15}$$

The aim of sliding mode control approach is to define asymptotically stable surfaces such that all system trajectories converge to these surfaces and slide along them until achieving the origin at their intersection [37]. Furthermore, the basic idea of decoupled sliding mode control is to design a control law such that the single input u simultaneously controls the two subsystems to accomplish the desired performance.

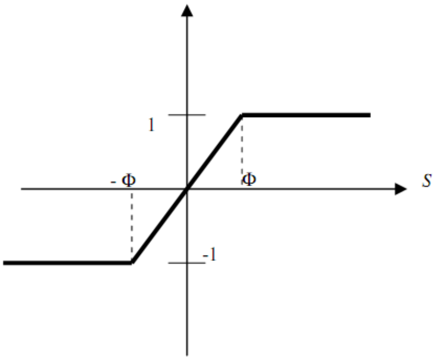

Fig. 1. Definition of function $Sat\left(\frac{s}{\Phi}\right)$ to eliminate the discontinuity in control law.

To achieve this goal, the following sliding surfaces are defined:

$$s_1(x) = \lambda_1(x_2 - x_{2d} - z) + x_4 - x_{4d} = 0 \qquad (16a)$$

$$s_2(x) = \lambda_2(x_1 - x_{1d}) + x_3 - x_{3d} = 0 \qquad (16b)$$

Here, z is a value proportional of s_2 with respect to a proper range from x_2.

$$u_1 = \hat{u}_1 - G_{f_1} sat\left(s_1(x) b_2(x)\, G_{s_1}\right), G_{f_1}, G_{s_1} > 0 \qquad (17)$$

with

$$\hat{u}_1 = -b_1^{-1}(x)\left(f_2(x) - \ddot{x}_{2d} + \lambda_1 x_4 - \lambda_1 \dot{x}_{2d}\right) \qquad (18)$$

So, let the control law for Equation (16b) be a sliding mode mimicking a sliding mode with boundary layer:

$$z = sat\left(s_2\, G_{s_2}\right) G_{f_2}\quad 0 < G_{f_2} < 1 \qquad (19)$$

where G_{s_2} represents the inverse of the width of the boundary layer to s_2, G_{f_2} transfers s_2 to the proper range of x_2 [38].

D. Inverted Pendulum System

In this section, a model of an inverted pendulum is investigated. A pole, hinged to a cart moving on a track, is balanced upwards by motioning the cart via a DC motor. Moreover, the cart has to track a varying reference position (Figure 2). The state vector, $x = [x_1, x_2, x_3, x_4]^T$, is the system observable state vector. It includes the cart horizontal distance from the track centre, the pole angular distance from the upwards equilibrium point and their derivatives, respectively. The force applying on the cart may be expressed as $F = \alpha u$, where u is the limited motor supply voltage. The mathematical model of the system is:

$$\begin{cases} \dot{x}_1 = x_3 \\ \dot{x}_2 = x_4 \\ \dot{x}_3 = f_1(x) + b_1(x)u \\ \dot{x}_4 = f_2(x) + b_2(x)u \end{cases} \qquad (20)$$

where

$$f_1(x) = \frac{a(-T_c - \mu x_4^2 \sin x_2) - l\cos(x_2)(\mu g \sin x_2 - f_p x_4)}{J + \mu l \sin^2 x_2} \qquad (21)$$

$$b_1(x) = \frac{a\alpha}{J + \mu l \sin^2 x_2} \qquad (22)$$

$$f_2(x) = \frac{l\cos(x_2)\,(-T_c - \mu x_4^2 \sin x_2) + \mu g \sin x_2 - f_p x_4}{J + \mu l \sin^2 x_2} \qquad (23)$$

$$b_2(x) = \frac{l\cos x_2 \alpha}{J + \mu l \sin^2 x_2} \qquad (24)$$

that $l = \frac{L m_p}{2(m_c + m_p)}$, $a = l^2 + \frac{J}{m_c + m_p}$ and $\mu = (m_c + m_p) l$.

The cart and pole masses are respectively m_c and m_p, g represents the gravity acceleration, L is half the pole length, J is the cart and pole overall moment of inertia with respect to the system centre of mass, f_p is the pole rotational friction coefficient and T_c is the horizontal friction acting on the cart, which is a nonlinear function of the cart speed x_3 . Note that in Equation (20) $b_1(x)$ and $b_2(x)$ are bounded.

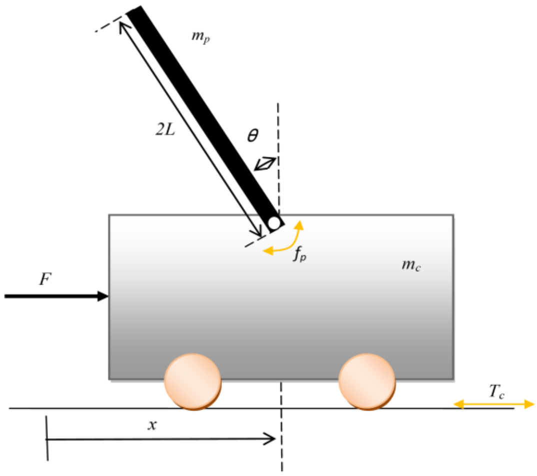

Fig. 2. The structure of an inverted pendulum system.

3. Multi-Objective Particle Swarm Optimization

In this paper, a new MOPSO namely Ingenious-MOPSO introduced in [16] will be applied to find the Pareto frontier of non-commensurable objective functions for the optimal design of the decoupled sliding mod control of the inverted pendulum system. The details of the algorithm are introduced as the following.

A. Global Best Position

In [16], a leader selection technique based on density measures was described. In fact, a neighborhood radius $R_{neighborhod}$ is defined for all non-dominated solutions. Two non-dominated solutions are neighbors if their Euclidean distance (measured in the objective domain) between them is less than $R_{neighborhod}$. Using this definition, the number of neighbors of each non-dominated solution is calculated and the particle which has the fewer neighbors is preferred as the leader.

B. Personal Best Position

Via choosing a good technique for finding $\vec{x}_{pbest_i}$ for particle *i*, the diversity within the swarm is maintained. Introduced approach in [7] (named Sigma method) is implemented to find the personal best position of the particle [16].

C. Turbulence Operator

In order to avoid being trapped in local minima and have the opportunity to discover other positions, the turbulence operator is used [16]. For this purpose, *N* particles of the population are randomly replaced by the new generated positions.

$$\vec{x}_i(t) = \vec{x}_{min}(t) + rand \times (\vec{x}_{max} - \vec{x}_{min}) \qquad (25)$$

where $rand$ is a random number which generated uniformly in the interval [0,1] and $\vec{x}_{max}$ and $\vec{x}_{min}$ are upper and lower bounds of the search space. It is proposed in [16] that $N = P_m \times number\ of\ particles$ that P_m is the probability of the turbulence operator and here considered as $\frac{5}{t}$.

D. Fuzzy Elimination Technique

The fuzzy elimination technique proposed in [16] is used to prune the archive. In this approach, each particle in the archive has an elimination radius equal to $\varepsilon_{\text{Fuzzy}}$ and if its Euclidean distance (in the objective function space) with another particle is less than $\varepsilon_{\text{Fuzzy}}$, then one of them will be omitted.

E. Details Of The Multi-Objective Algorithm

The Ingenious-MOPSO algorithm is briefly described in the following. Initially, the population should be randomly generated and the inertia weight, the learning factors, and turbulence probability would be assigned. In the each iteration, after calculation of the fitness values of all particles, the archive, $\vec{x}_{pbest_i}$ and $\vec{x}_{gbest}$ would be updated. Then, for each particle, a random number $\rho \in [0,1]$ would be allocated. If a particle has $\rho < (turbulence\ probability)$ then a new particle will be produced by the turbulent operator. Other particles that are not selected for turbulent operation will be enhanced by PSO. This cycle should be repeated until achieving the maximum iteration criterion.

4. Optimal Decoupled Sliding Mode Control

In fact, the heuristic sliding parameters are required to be chosen properly. Here, this problem is solved via Ingenious-MOPSO and these results are compared with three prominent algorithms.

The performance of a controlled closed loop system is usually evaluated by variety of goals [39-40]. In this paper, normalized angle error of the pole and normalized distance error of the cart are considered as the objective functions. These objective functions have to be minimized, simultaneously.

The vector $[G_{f_1}, G_{s_1}, \lambda_1, G_{f_2}, G_{s_2}, \lambda_2]$ is the vector of selective parameters (design variables) of sliding mode control. The regions of the design variables are as G_{f_1} , G_{s_1} and λ_1 belong to [0,10] and G_{f_2}, G_{s_2} and λ_2 belong to [0,1]. When solving proposed multi-objective problem, the population size is set at 100 and also, the maximum iteration is set at 200.

The initial values for the inverted pendulum system are as following: $x_1(0) = 0, x_2(0) = \frac{\pi}{6}, x_3(0) = 0, x_4(0) = 0$. The system parameters and constants used in simulation are given in Table 1. When the multi-objective optimization algorithms are applied to this problem, a Pareto front of normalized angle error of the pole and normalized distance error of the cart would be achieved that is demonstrated in Figure 3. Although the performances of all algorithms are competitively good over this problem, the most interesting result is that the proposed algorithm has more uniformity and diversity. In Figure 3, points A and C stand for the best normalized angle error of the pole and normalized distance error of the cart, respectively.

It is clear from Figure 3 that all the optimum design points in the Pareto front are non-dominated and choosing a better value for any objective function in the Pareto front would cause a worse value for another objective. The corresponding decision variables (vector of sliding-mode control parameters) of the Pareto front shown in Figure 3 are the best possible design points. Point C could be a trade-off optimum choice when considering minimum values of both of the normalized angle error of the pole and normalized distance error of the cart. Design variables and objective functions corresponding to the optimum design points A, B, and C are illustrated in Table 2. The time response of these optimum design points are shown in Figures 4 and 5. Also, this problem is carried out with disturbance (a unit impulse torque is applied to the pole in $Time = 10\,S$). The simulation results with disturbance are shown in Figures 6 and 7 that demonstrate the effectiveness and robustness of the proposed strategy to handle disturbances.

TABLE 1. THE INVERTED PENDULUM PARAMETERS.

The mass of the pendulum	m_p	$0.5\ kg$
The mass of the cart	m_c	$2\ kg$
The half length of the pendulum	L	$0.5\ m$
The inertia moment of the cart and pendulum	J	$0.4\ kg\ m^2$
The friction constant of the pendulum	f_p	*0.1*
The friction constant of the cart	T_c	*0.25*
The gravity acceleration	g	$9.81\ \frac{m}{s^2}$
The force coefficient	α	3

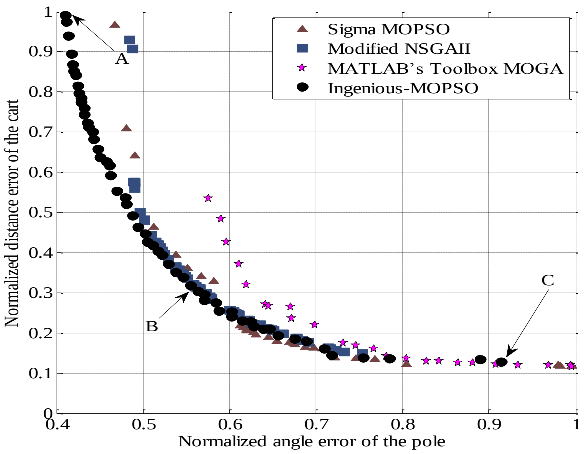

Fig. 3. The obtained Pareto fronts by using Sigma method [7], modified NSGAII [14], MATLAB's Toolbox MOGA and Ingenious-MOPSO for optimal decoupled sliding mode control design of the inverted pendulum system.

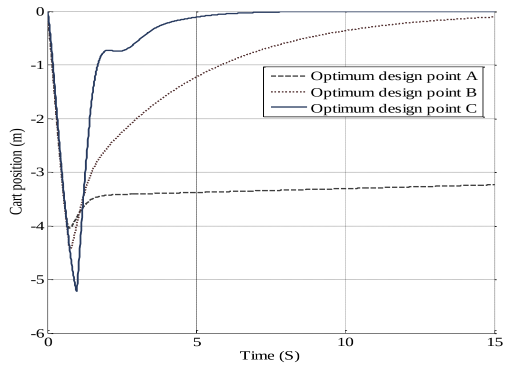

Fig. 4. Simulations of the cart position correspond to the optimum design points A, B, and C shown in the Pareto front.

The objective functions and their associated design variables for the optimum points of Figure 3.

Optimum design point	*A*	*B*	*C*
Normalized angle error of the pole	4.100×10^{-1}	5.553×10^{-1}	9.137×10^{-1}
Normalized distance error of the cart	9.905×10^{-1}	3.186×10^{-1}	1.128×10^{-1}
Design variable G_{f_1}	9.816×10^{0}	9.720×10^{0}	9.621×10^{0}
Design variable G_{s_1}	9.122×10^{0}	9.532×10^{0}	9.148×10^{0}
Design variable λ_1	1.346×10^{0}	1.392×10^{0}	1.432×10^{0}
Design variable G_{f_2}	9.996×10^{-1}	9.997×10^{-1}	9.997×10^{-1}
Design variable G_{s_2}	1.979×10^{-1}	1.914×10^{-1}	1.790×10^{-1}
Design variable λ_2	4.296×10^{-3}	2.178×10^{-1}	5.942×10^{-1}

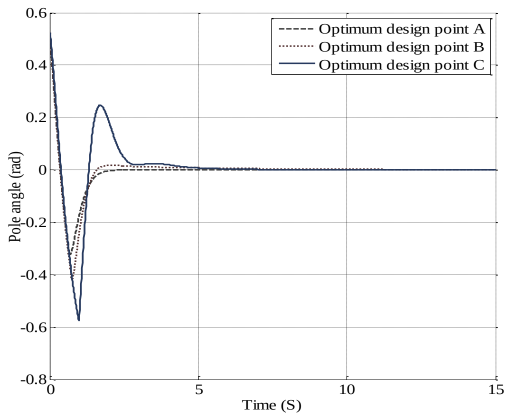

Fig. 5. Simulations of the pole angle correspond to the optimum design points A, B, and C shown in the Pareto front.

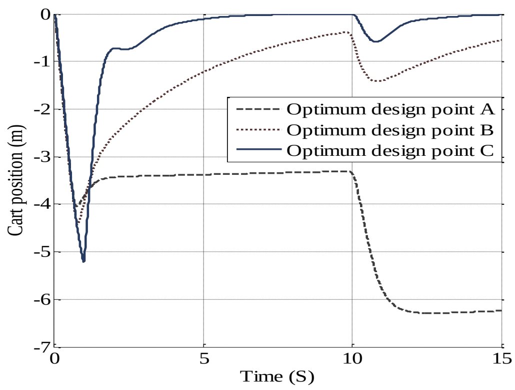

Fig. 6. Simulations of the cart position with a unit impulse disturbance correspond to the optimum design points A, B, and C shown in the Pareto front.

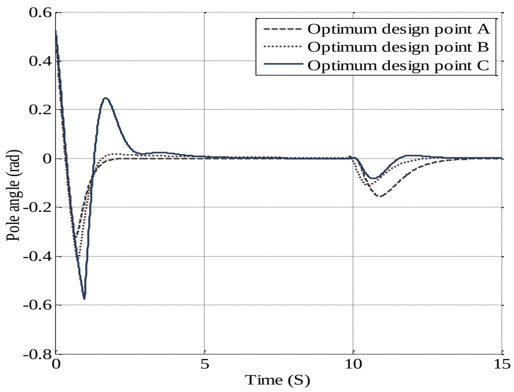

Fig. 7. Simulations of the pole angle with a unit impulse disturbance correspond to the optimum design points A, B, and C shown in the Pareto front.

5. Conclusion

In this paper, the decoupled sliding mode technique is proposed for stabilising an inverted pendulum system. Choice of a suitable combination of the controller parameters is critical for obtaining a satisfactory behaviour. Thus, different multi-objective optimization methods are employed in order to optimize the decoupled sliding mode controller parameters regarding to error of position and angle simultaneously. Comparison among Pareto optimal fronts obtained from these methods demonstrated that the Ingenious-MOPSO out performs others.

References

[1] J. Kennedy, R.C. Eberhart, "Particle swarm optimization," in Proceedings of the IEEE International Conference on Neural Networks, vol. IV, Perth, Australia, pp. 1942-1948, 1995.

[2] P.J. Angeline, "Using selection to improve particle swarm optimization," in Proceedings of the IEEE Congress on Evolutionary Computation, Anchorage, AK, pp. 84-89, May 1998.

[3] R. C. Eberhart, Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in Proceedings of the IEEE Congress on Evolutionary Computation, Anchorage, AK, pp. 611-616, May 1998.

[4] H. Yoshida, K. Kawata, Y. Fukuyama, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," IEEE Transactions on Power Systems, vol. 15, pp. 1232-1239, 2000.

[5] X. Hu, R. C. Eberhart, "Multi-objective optimization using dynamic neighborhood particle swarm optimization," In Proceedings of the IEEE World Congress on Computational Intelligence (CEC'02), pp. 1677-1681, 2002.

[6] J. E. Fieldsend, S. Singh, "A multi-objective algorithm based upon particle swarm optimization and efficient data structure and turbulence," In Workshop on Computational Intelligence, pp. 34-44, 2002.

[7] S. Mostaghim, 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.

[8] K. E. Parsopoulos, D. K. Tasoulis, M. N. Vrahatis, "Multi-objective optimization using parallel vector evaluated particle swarm optimization," Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, vol. 2, pp. 823-828, 2004.

[9] P. K. Tripathi, S. Bandyopadhyay, S. K. Pal, "Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients," Information Sciences, vol. 177, pp. 5033-5049, 2007.

[10] S. J. Tsai, T. Y. Sun, C. C. Liu, S. T. Hsieh, W. C. Wu, S. Y Chiu, "An improved multi-objective particle swarm optimizer for multi-objective problems," Expert Systems with Applications, vol. 37, pp. 5872-5886, 2010.

[11] Y. Wang, Y. Yang, "Particle swarm optimization with preference order ranking for multi-objective optimization," Information Sciences, vol. 179, pp. 1944-1959, 2009.

[12] S. Mostaghim, J. Teich, "The role of ε-dominance in multi objective particle swarm optimization methods," Congress on Evolutionary Computation, vol. 3, pp. 1764-1771, 2003.

[13] A. G. Hernandez-Diaz, L. V. Santana-Quintero, C. A. Coello Coello, J. Molina, R. Caballero, "Improving the efficiency of ε-dominance based grids, Information Sciences," vol. 181, no. 15, pp. 3101-3129, 2011.

[14] K. Atashkari, N. Nariman-Zadeh, M. Golcu, A. Khalkhali, 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, pp. 1029-1041, 2007.

[15] M. J. Mahmoodabadi, A. Bagheri, S. Arabani Mostaghim, M. Bisheban, “Simulation of stability using Java application for Pareto design of controllers based on a new multi-objective particle swarm optimization,” Mathematical and Computer Modelling, vol. 54, no. 5-6, pp. 1584-1607, 2011.

[16] M. J. Mahmoodabadi, M. Taherkhorsandi, A. Bagheri, “Optimal robust sliding mode tracking control of a biped robot based on ingenious multi-objective PSO,” Neurocomputing, vol. 124, pp. 194–209.

[17] P. Fleming, R. Purshouse, “Evolutionary algorithms in control systems engineering: a survey,” Control Engineering Practice, vol. 10, no. 11, pp. 1223-1241, 2002.

[18] C. Fonseca, P. Fleming, “Multi-objective optimal controller design with genetic algorithms,” International Conference on Control, vol. 1, March, pp. 745-749, 1994.

[19] G. Sanchez, M. Villasana, M. Strefezza, “Multi-objective pole placement with evolutionary algorithms,” Lecture Notes in Computer Science, vol. 4403, pp. 417, 2007.

[20] W. Qiao, G. Venayagamoorthy, R. Harley, “Design of optimal PI controllers for doubly fed induction generators driven by wind turbines using particle swarm optimization,” International Joint Conference on Neural Networks, pp. 1982-1987, 2006.

[21] Z. L. Gaing, “A particle swarm optimization approach for optimum design of PID controller in AVR system,” IEEE Transaction on Energy Conversion, vol. 19, no. 2, pp. 384-391, 2004.

[22] K. C. Ng, Y. Li, D. Munay-Smith, K.C. Shaman, “Genetic algorithm applied to fuzzy sliding mode controller design,” Genetic Algorithms in Engineering Systems, Innovations and Applications 12-14, Conference Publication No. 414, IEE, pp. 220-225, 1995.

[23] C. C Wong, S.Y. Chang, “Parameter selection in the sliding mode control design using genetic algorithms,” Tamkang Journal of Science and Engineering vol. 1 no. 2, pp. 115-122, 1998.

[24] C. C. Kung, T. H. Chen, L. H. Kung, “Modified adaptive fuzzy sliding mode controller for uncertain nonlinear systems,” IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences, vol. 88, no. 5, pp. 1328-1334, 2005.

[25] N. Yagiz, Y. Hacioglu, “Robust control of a spatial robot using fuzzy sliding modes,” Mathematical and Computer Modeling vol. 49, no. 1-2, pp. 114-127, 2009.

[26] C. Zhi-mei, M. Wen-jun, Z. Jing-gang, Z. Jian-chao, “Scheme of sliding mode control based on modified particle swarm optimization,” Systems Engineering-Theory & Practice, vol. 29, no. 5, pp. 137-141, 2009.

[27] E. Alfaro-Cid, E. W. McGookina, D. J. Murray-Smith, T. I. Fossen, “Genetic algorithms optimization of decoupled Sliding Mode controllers: simulated and real results,” Control Engineering Practice, vol. 13, pp. 739-748, 2005.

[28] M. J. Mahmoodabadi, A. Bagheri, N. Nariman-zadeh, A. Jamali, R. Abedzadeh Maafi, “Pareto design of decoupled sliding-mode controllers for nonlinear systems based on a multiobjective genetic algorithm,” Journal of Applied Mathematics pp. 22, 2012.

[29] M. J. Mahmoodabadi, S. Arabani Mostaghim, A. Bagheri, N. Nariman-zadeh, “Pareto optimal design of the decoupled sliding mode controller for an inverted pendulum system and its stability simulation via Java programming,” Mathematical and Computer Modelling, vol. 57, pp. 1070–1082, 2013.

[30] A. P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley & Sons, 2005.

[31] R. C. Eberhart, J. Kennedy, “A new optimizer using particle swarm theory,” Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39-43, 1995.

[32] A. Ratnaweera, S. K. Halgamuge, “Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficient,” IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 240-255, 2004.

[33] C. A. Coello Coello, D. A. Van Veldhuizen, G. B. Lamont, In: Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic, Dordrecht, 2002.

[34] A. Jamali, A. Hajiloo, 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.

[35] W. S. Lin, C. S. Chen, “Robust adaptive sliding mode control using fuzzy modeling for a class of uncertain MIMO nonlinear systems,” Control

Theory and Applications, IEE, vol. 149, no. 3, pp. 193-201, 2002.

[36] J. Jing, Q.H. Wuan, “Intelligent sliding mode control algorithm for position tracking servo system,” International Journal of Information Technology, vol. 12, no. 7, pp. 57-62, 2006.

[37] Utkin, Sliding Modes and Their Application in Variable Structure Systems, Central Books Ltd, 1978

[38] M. Dotoli, P. Lino, B. Turchiano, “A decoupled fuzzy sliding mode approach to swing-Up and stabilize an inverted pendulum,” The CSD03, The 2nd IFAC Conference on Control Systems Design, Bratislava, Slovak Republic, pp. 113-120, September 2003.

[39] R. Toscana, “A simple robust PI/PID controller design via numerical optimization approach,” Journal of Process Control, vol. 15, pp. 81–88, 2005.

[40] W.A. Wolovich, Automatic Control Systems, Harcourt Brace College Publication Orlando, Saunders College Publishing, USA, 1994.