ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **5**, May **2016** pp. 1061–1069

ROBUST CONTROL DESIGN FOR IPMC BASED ON A MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION-BASED OPERATOR APPROACH

Aihui Wang[1], Yiping Chang[1], Chunli Zhao[2] and Yan Wang[1]

[1]School of Electric and Information Engineering
[2]College of Information and Business
Zhongyuan University of Technology
No. 41, Zhongyuan Road, Zhengzhou 450007, P. R. China
wah781212@gmail.com

Received October 2015; accepted January 2016

Abstract. *In this paper, a robust nonlinear tracking control design for an ionic polymer metal composite (IPMC) is proposed by using a multi-objective particle swarm optimization (MOPSO)-based operator approach. Addressing the difficulties in obtaining the PI control parameters* (K_p, K_i) *of the former proposed nonlinear robust tracking control system based on operator-based robust right coprime factorization (RRCF) approach, in this paper, how to obtain the* K_p *and* K_i *is investigated by using MATLAB system identification toolbox and MOPSO algorithm. That is, firstly, a new equivalent transfer function model of the robust stable control system of IPMC is identified based on the MATLAB system identification toolbox. For the obtained equivalent transfer model, an MOPSO algorithm is used to obtain the* K_p *and* K_i*. Finally, the effectiveness of the proposed method is also verified by simulation results.*
Keywords: IPMC, Robust tracking control, MATLAB system identification toolbox, MOPSO

1. **Introduction.** Similar to piezoelectric materials, the IPMC, also called artificial muscle, belongs to the category electroactive polymers (EAP), which is one of the most promising EAP actuators, and has been used in the bio-robotics community. An IPMC sample consists of a thin ion-exchange membrane (e.g., Nafion) plated on both surfaces with a noble metal as electrodes. Because IPMCs have the following characteristics: large strain and stress induced electrically, light in weight, small and simple mechanisms, small electric consumption, and low drive voltage, which have been widely used in the developments of miniature robots, biomimetic sensors, actuators, and transducers [1].

The dynamic mode of IPMC is usually broken up into two different types: linear model and nonlinear model. Linear models have no prior knowledge or some knowledge of the system. Nonlinear models have a comprehensive knowledge of the physics system derivation [2]. For linear models, linear quadratic regulator (LQR) and proportional integral derivative (PID) have been used in precise displacement control. However, the IPMC shows mainly nonlinear behaviors in characteristics of large strain and stress. Moreover, the control performance is affected by the parameter variations and various disturbances easily, and it is difficult to obtain a precise mathematical model. So, in order to improve the control performance and achieve robust tracking, some approaches such as adaptive control scheme, neural network control method, and fuzzy control algorithm have been designed in precision position control and achieved some good performance results. However, adaptive control scheme demands real-time seriously and has limitations in the system structure with uncertainties. Neural network control method has to retrain the network when environment changes because of its own self-learning feature. The track convergence rate is slow and the robustness of the control system is not well guaranteed by using fuzzy control approach. Therefore, a practical mathematical model and an effective control strategy are desirable in designing precision displacement control system for the IPMC. As we all know, the operator-based RRCF approach has been a promising control strategy for analysis, stabilization and control of nonlinear system with disturbances and model uncertainties [3,4]. Especially, the operator-based RRCF approach has attracted much attention due to its convenience in researching input-output stability problems of the nonlinear system with uncertainties. Addressing the IPMC with unknown uncertainties consisting of modelling error and external disturbances, a robust stable control system to IPMC has been proposed by using the operator-based RRCF approach, the robust stability of the designed system was guaranteed, and the tracking performance was also guaranteed by the PI tracking controller [5,6].

However, because the equivalent block of the stabilizing system is also nonlinear process, it is difficult to design the optimal control parameters (K_p, K_i) of PI controller. Therefore, for the purpose of real application, how to obtain the K_p and K_i is also a key issue. As we all know, some useful methods in designing K_p and K_i are based on the fundamental assumption that the controlled plant is a linear system with the identified transfer function. Moreover, nowadays there are many swarm intelligence approaches used to optimize the controllers parameters [7-11] for the linear process. In nature, most controllers optimization and design problems are multi-objective, since they normally have several (possibly conflicting) objectives that must be satisfied at the same time. At this point, the definition of a multi-objective optimization problem can be more effective. The multi-objective techniques offer advantages when compared with single objective optimization techniques because they may produce a solution with different tradeoffs among different individual objective. Compared with single evolutional algorithm, the multi-objective particle swarm optimization (MOPSO) has a simpler computational paradigm and has shown faster convergence and better computational efficiency.

Motivated by the aforementioned issues, this paper investigated how to optimize the control parameters (K_p, K_i) using MOPSO algorithm to improve the tracking performance for the former proposed robust control system. Namely, firstly, for the obtained stabilizing system based on operator-based RRCF approach, the MATLAB system identification toolbox is used to identify the equivalent transfer function. Addressing the equivalent linear model, the control parameters (K_p, K_i) are designed by using MOPSO algorithm. Finally, the simulation results are also given to confirm the effectiveness of the proposed method.

The outline of the paper is organized as follows. In Section 2, the preliminaries and problem statement are introduced. Tracking control design and control parameters optimization are investigated in Section 3. The simulation results are given in Section 4, and Section 5 is the conclusion.

2. **Preliminaries and Problem Statement.**

2.1. **Nonlinear dynamic model of IPMC.**

A nonlinear dynamic model with uncertainties of IPMC is modeled by the following equations [6]:

$$\begin{cases} \dot{x} = \dfrac{(x-au)\sqrt{2b\left(\dfrac{xe^{-x}}{1-e^{-x}} - \ln\left(\dfrac{xe^{-x}}{1-e^{-x}}\right) - 1\right)}}{SK_e b(R_a+R_c)\left(1-\dfrac{1-e^{-x}}{xe^{-x}}\right)\dfrac{e^{-x}\left(1-x-e^{-x}\right)}{\left(1-e^{-x}\right)^2}} \\ y = \dfrac{3\alpha_0 K_e\sqrt{2b\left(\dfrac{xe^{-x}}{1-e^{-x}} - \ln\left(\dfrac{xe^{-x}}{1-e^{-x}}\right) - 1\right)}}{aY_e h^2} + \triangle P \end{cases} \tag{1}$$

where, x is the state variable, u is the control input voltage, y is the curvature output, R_a is the electrodes resistance, R_c is the ion diffusion resistance, Y_e is the equivalent Youngos modulus of IPMC, α_0 is the coupling constant, K_e is the effective dielectric constant of the polymer, h is the thickness of IPMC, $a = \frac{F}{RT}$, $b = \frac{F^2C^-}{RTK_e}$, F is Faraday's constant, C^- is the anion concentrations, R is the gas constant, T is the absolute temperature, and $\triangle P$ is uncertainties consisting of measurement error of parameters and model error of the IPMC.

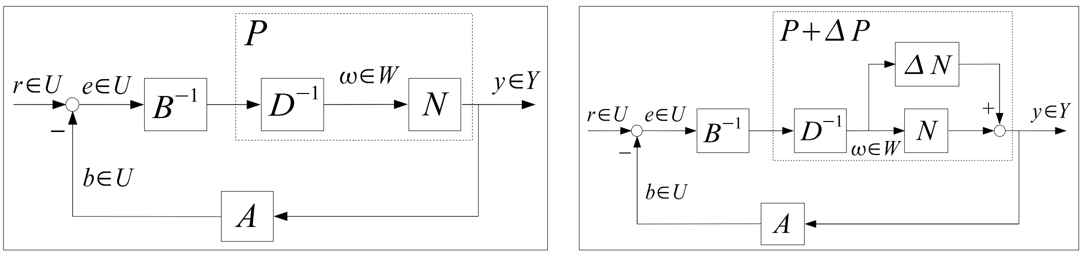
2.2. **Operator-based robust right coprime factorization.** In this section, some basic definitions and notations of operator, right coprime factorization, and RRCF are outlined.

Definition 2.1. ***An operator*** *$Q : \mathbf{X} \to \mathbf{Y}$ is a mapping defined from the input space $\mathbf{X}$ to the output space $\mathbf{Y}$, and can also be expressed in the mathematical form as $y(t) = Q(u)(t)$ where $u(t)$ is the element of $\mathbf{X}$ and $y(t)$ is the element of $\mathbf{Y}$. Let $Q : \mathbf{X} \to \mathbf{Y}$ be an operator mapping from $\mathbf{X}$ to $\mathbf{Y}$, and denoted by $\mathcal{D}(Q)$ and $\mathcal{R}(Q)$, respectively, the domain and range of Q. If the operator $Q : \mathcal{D}(Q) \to \mathbf{Y}$ satisfies addition rule and multiplication rule*

$$Q : ax_1 + bx_2 \to aQ(x_1) + bQ(x_2)$$

for all x_1, $x_2 \in \mathcal{D}(Q)$ and all $a, b \in \mathcal{C}$, then Q is said to be linear; otherwise, it is said to be nonlinear.

The operator based nonlinear feedback control systems are shown in Figure 1, where $\mathbf{U}$ and $\mathbf{Y}$ are used to denote the input space and output space of a given plant operator P, respectively, i.e., $P : \mathbf{U} \to \mathbf{Y}$. P is nominal plant, ΔP is unknown bounded uncertainties, and $P + \Delta P$ is the real plant. In the operator based control system shown in Figures 1(a) and 1(b), the right factorization, right coprime factorization, and RRCF were defined as follows, respectively.



(a) (b)

FIGURE 1. (a) Operator based nonlinear feedback system, (b) operator based nonlinear feedback system with uncertainties

Definition 2.2. *The given plant operator $P : \mathbf{U} \to \mathbf{Y}$ is said to have a* ***right factorization****, if there exists a linear space $\mathbf{W}$ and two stable operators $D : \mathbf{W} \to \mathbf{U}$ and $N : \mathbf{W} \to \mathbf{Y}$ such that $P = ND^{-1}$ where D is invertible. Such a factorization of P is denoted by (N, D) and $\mathbf{W}$ is called a quasi-state space of P. The factorization is said to be coprime, or P is said to have a* ***right coprime factorization*** *in Figure 1(a), if there exist two stable operators $A : \mathbf{Y} \to \mathbf{U}$ and $B : \mathbf{U} \to \mathbf{U}$, satisfying the Bezout identity*

$$AN + BD = T, \quad \text{for some } T \in \mu(\mathbf{W}, \mathbf{U}) \tag{2}$$

Generally speaking, for the corresponding control system with uncertainties, let the Bezout identity of the nominal process and the real process be $AN + BD = M \in \mathcal{U}(\mathbf{W}, \mathbf{U})$, $A(N + \Delta N) + BD = \tilde{M}$, respectively. If

$$\|(A(N + \Delta N) - AN)M^{-1}\| < 1 \tag{3}$$

then the system shown in Figure 1(b) is BIBO stable, and is called ***robust right coprime factorization****.*

It is worth mentioning that the initial state should also be considered, that is, $AN(\omega_0, t_0) + BD(\omega_0, t_0) = M(\omega_0, t_0)$ should be satisfied. In this paper, $t_0 = 0$ and $\omega_0 = 0$ are selected.

2.3. **Problem statement.** For the model described in (1), there exist the unknown uncertainties ΔP in the IPMC dynamic model, and the uncertainties are unknown but bounded. For the model (1), the right factorizations were designed in [5,6],

$$\begin{cases} D(\omega)(t) = \dfrac{S\kappa_e b(R_a+R_c)\dot{\omega}(t)\left(1-\dfrac{1-e^{-\omega(t)}}{\omega(t)e^{-\omega(t)}}\right)\dfrac{e^{-\omega(t)}\left(1-e^{-\omega(t)}-\omega(t)\right)}{\left(1-e^{-\omega(t)}\right)^2}}{a\sqrt{2b\left(\dfrac{\omega(t)e^{-\omega(t)}}{1-e^{-\omega(t)}}-\ln\left(\dfrac{\omega(t)e^{-\omega(t)}}{1-e^{-\omega(t)}}\right)-1\right)}}+\dfrac{\omega(t)}{a} \\ N(\omega)(t) = \dfrac{3\alpha_0\kappa_e\sqrt{2b\left(\dfrac{\omega(t)e^{-\omega(t)}}{1-e^{-\omega(t)}}-\ln\left(\dfrac{\omega(t)e^{-\omega(t)}}{1-e^{-\omega(t)}}\right)-1\right)}}{aY_eH^2} \\ \Delta N(\omega)(t) = \Delta\sqrt{2b\left(\dfrac{\omega(t)e^{-\omega(t)}}{1-e^{-\omega(t)}}-\ln\left(\dfrac{\omega(t)e^{-\omega(t)}}{1-e^{-\omega(t)}}\right)-1\right)} \end{cases} \tag{4}$$

Considering the nonlinear IPMC with bounded uncertainties, a robust stable control system based on Figure 1(b) has been investigated in [5,6], and the operators A and B were designed as,

$$\begin{cases} A(y)(t) = -\frac{aSY_eH^2(R_a+R_c)}{3\alpha_0}\dot{y}(t) \\ B(u_d)(t) = au_d(t) \end{cases} \tag{5}$$

The conditions of robust stability have been given. Moreover, the tracking control scheme was designed based on the obtained stabilizing system and shown in Figure 2. The controller C was designed as,

$$u^*(t) = K_p\tilde{e}(t) + K_i\int\tilde{e}(\tau)d\tau \tag{6}$$

where, K_p and K_i were the parameters of PI controller. The control output y can track the desired reference input r^*, and the detailed proof has been given in [6].

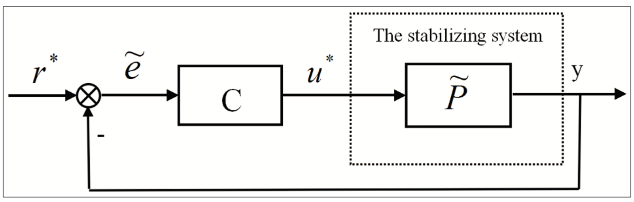


FIGURE 2. The tracking control system

However, because the equivalent block of the stabilizing system is nonlinear process, how to obtain the optimal K_p and K_i is also difficult. As a result, in this paper, a tracking control design for an IPMC is proposed using a new model equivalent approach by using MATLAB system identification toolbox and MOPSO algorithm-based operator approach. Firstly, how to identify an equivalent transfer function model of the obtained stabilizing system of IPMC using MATLAB system identification toolbox is investigated. Secondly, addressing the identified linear model, a proportional integral controller is designed to realize perfect tracking control. Finally, an MOPSO algorithm is used to obtain the optimal control parameters K_p, K_i.

3. Tracking Control Design and Optimization.

3.1. Equivalent model identification using MATLAB system identification toolbox.

In this paper, the stabilizing system in Figure 1(b) is identified as a new equivalent model by using the MATLAB system identification toolbox. For the identified equivalent transfer function, it is easy to obtain the tracking conditions and the control parameters K_p and K_i. As we all know, the MATLAB system identification toolbox is to build the accurate and simplified models of the complex systems by using the time-series data. It provides some useful tools for creating the mathematical models of dynamic systems based on the observed input-output data. The identification process amounts to repeatedly selecting a model structure, computing the best model in the structure, and the final objective is to see if they are satisfied by evaluating the main properties of this model. The program cycle can be itemized as follows.

(1) Design an experiment and collect input-output data from the process to be identified.

(2) Select and define a model structure (a set of candidate system descriptions) within which a model is to be found.

(3) Compute the best model in the model structure according to the input-output data and a given criterion of fit.

(4) If the model is good enough, then stop; otherwise go back to (2) to try another model set.

In this paper, the input data is the reference of the stabilizing system, the output data is the response of the stabilizing system. Suppose that the identified transfer function is the stabilizing system, so in step (2), select transfer function as the model structure. After step (4), choose the model which has the best fit as the new model. As for the obtained new equivalent linear transfer model, a PI tracking system is easily designed in Figure 2. Moreover, in order to obtain more suitable control parameters K_p and K_i for improving further tracking obtainment, an MOPSO algorithm is used to identify the optimal K_p and K_i.

3.2. Parameters optimization using MOPSO algorithm.

3.2.1. *Particle swarm optimization.*

Particle swarm optimization (PSO) is a popular computational and search technique developed by Kennedy and Eberhart that is based on the social behavior of birds flocking to look for food [10]. In PSO, a population of possible solutions (called particles) are first initialized. These particles are then allowed to explore through a solution search space looking for the optimum solution. Each particle maintains the best solution it has found thus far (particle best) as well as the best solution that the group (called particle swarm) has found thus far (global best). The direction of the search is then updated based on the values of particle bests, and the group is global best.

In the canonical PSO, each particle i has position z_i and velocity v_i that is updated at each iteration as follows,
Velocity Equation:

$$\vec{v_i} = \omega \vec{v_i} + C_1 \vec{r_{1i}} \left(\vec{p_i} - \vec{z_i}\right) + C_2 \vec{r_{2i}} \left(\vec{p_g} - \vec{z_i}\right) \tag{7}$$

Position Equation:

$$\vec{z_i} = \vec{z_i} + \vec{v_i} \tag{8}$$

The factors ω and C_1 are the particles inertia and self-confidence factors, respectively. The confidence factor for the entire swarm is expressed by C_2. The quantities r_1 and r_2 are the positive random numbers drawn from a uniform distribution. $\vec{p_i}$ is the best position found so far by particle, and $\vec{p_g}$ is the global best so far found by the swarm.

3.2.2. *Multi-objective particle swarm optimization.* In order to extend the PSO to solve multi-objective problems, the single global best is extended into a fixed-sized archivement of non-dominated solutions accumulated during the search process [11]. The crowding distance density estimator is also incorporated in selecting the global best and in the deletion of the low ranking non-dominated solutions in the sorted archive. The main steps of the MOPSO algorithm can be summarized as follows.

- Position and velocity initialization.
- Initialize each particle personal best.
- Store the non-dominated solution into the global best archivement.
- Repeat the following until the maximum number of iterations is reached or a convergence criterion is met.
  (1) Sort the archivement based on crowding distance in descending order.
  (2) For each particle, randomly select a global best from the top 10% of the archivement, and use it to update the velocity and position.
  (3) Update each particle's personal best: if the particle's new position dominates the current personal best, then replace it with the particle's new position. If they are both non-dominated, there is a 50% of chance the personal best will be replaced.
  (4) Update the global best archivement: insert new non-dominated solutions into archivement and eliminate archived solutions that are dominated by the new solutions. If the archivement is full, Crowing Distance is adopted to maintain the diversity and applied to removing the extra ones.

Designing the appropriate fitness functions is the key to solving the multi-objective optimization problems. In the equivalent stabilizing system, it is expected that the position could achieve the desired output through a stationary process in a relatively short time. In other words, the three indices: setting time t_s, maximum overshoot M_p, steady state error e_{ss} should be minimum. Therefore, setting time, maximum overshoot and steady state error are selected as performance indices. The fitness functions in the MOPSO algorithm are expressed as follows: $f_1 = t_s$, $f_2 = M_p$, and $f_3 = e_{ss}$, where t_s is the setting time of the control system, and is also called as transition time; M_p is the maximum overshoot which reflects the stability of the transition process of the IPMC control system; e_{ss} is the steady state error which reflects the deviation of the actual output and the expected output of the system, and it describes the control accuracy of the control system. In the training, the t_s, M_p, and e_{ss} should be as small as possible by tuning K_p and K_i.

4. **Simulation.**

Some identified physical parameters based on the proposed method in [5] are shown in Table 1. In the simulation, the robust stable simulation result of the proposed IPMC control system based on Figure 1(b) is shown in Figure 3. In equivalent identification, the reference control input is 1 (1/m), and the response is the input-out data of the Matlab system identification toolbox. In Figure 4, the transfer function "tf1, tf2, tf3" are the first, second, third order transfer function and the tf2 which is second order has the best fit for the model of stabilizing system. So, the equivalent transfer model of the stabilizing system in Figure 1(b) can be identified as,

$$G(s) = \frac{0.24}{s^2 + 39.3s + 14.33} \tag{9}$$

The parameters K_p and K_i are optimized by the MOPSO algorithm. The parameters of MOPSO are initialized as follows.

- The population size is set to 40.
- The maximum generations are set to 100.
- The inertia weight ω is set to 0.725.
- The learning factors C_1 and C_2 are set to 2.

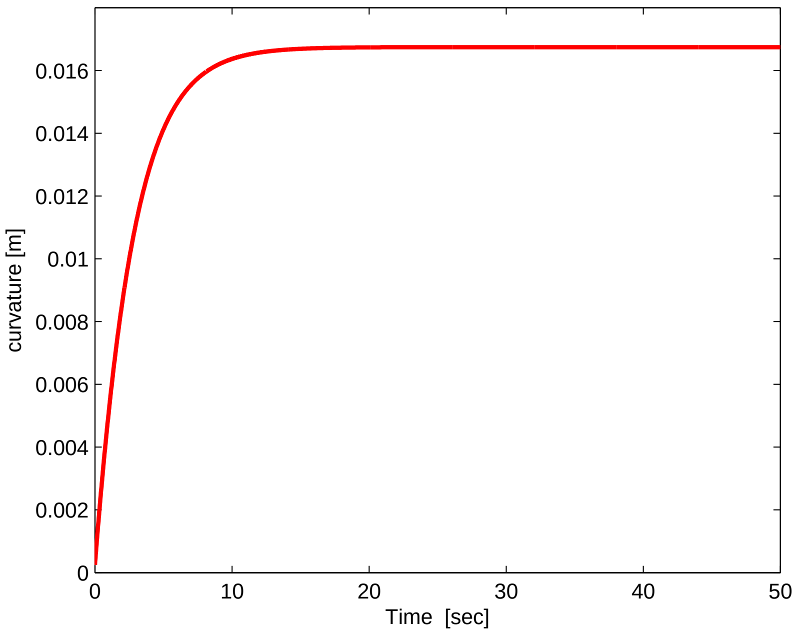


Figure 3. The response of the stabilizing system

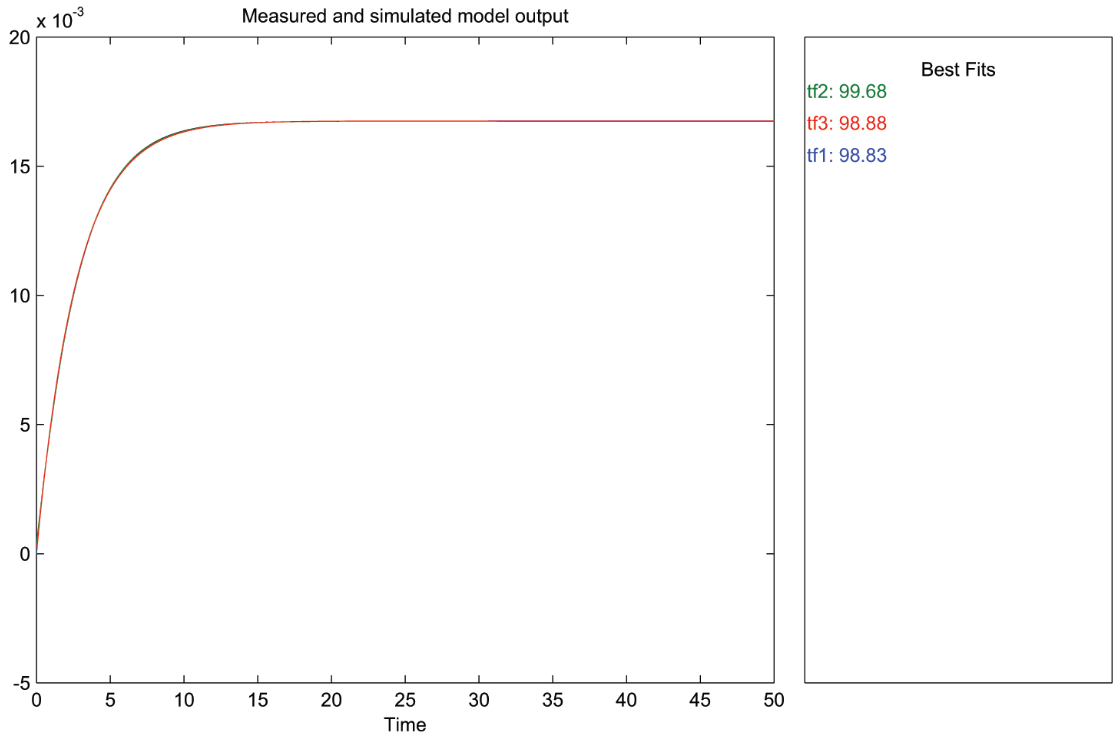


Figure 4. The fits of the transfer models

Table 1. Parameters of the model

L	h	W	T	R_a	R_c
50[mm]	200[μm]	10[mm]	290[K]	18[Ω]	60[Ω]
Y_e	α_0	C^{-1}	F	R	K_e
0.056[Gpa]	0.129[J/C]	981[mol/m^2]	96487[C/mol]	8.3143[J/mol.K]	1.12$\times$10^{-6}[F/m]

TABLE 2. Optimized PI controllers

Solutions	K_p	K_i	t_s	M_p	e_{ss}
The first solution of MOPSO	580	3.99	0.1	0.04	0
The second solution of MOPSO	600	3.95	0.09	0	0
The third solution of MOPSO	610	1.94	0.13	0	0.07
The fourth solution of MOPSO	585	2.31	0.1	0	0.06

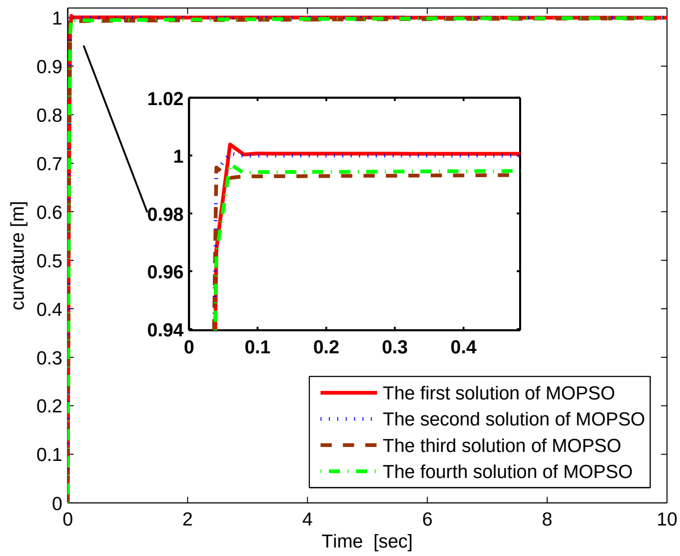


FIGURE 5. The simulation result based on the proposed method

Using the MOPSO algorithm, a large number of optimal solutions are achieved. Due to the limitation in space, only the best four groups identified parameters are listed in Table 2. In order to verify the effectiveness of the PI parameters, the obtained best four group parameters by the MOPSO algorithm have been confirmed in the PI controller, respectively. In the simulation experiment, K_p is in the range $[550, 650]$, K_i is in the range $[0, 5]$ and simulation time is T = 10sec. Step responses of the four group parameters are shown in Figure 5, respectively. In order to observe more clearly, some key parts of Figure 5 are amplified. Based on the part of amplification, it is clear that the controller outputs of the four group parameters have little differences. It also shows that the parameters of the controller have the characteristics of uncertainly and non-uniqueness. From Table 2 and Figure 5, the setting time is less than 0.15sec, and the overshoot and the steady error is so small to meet the demand.

5. **Conclusions.** In this paper, a robust nonlinear tracking control for IPMC was investigated by using MOPSO-based operator approach. The designed robust stable system based RRCF approach was identified as an equivalent transfer function model. The parameters of PI tracking controller were optimized by using MOPSO. Finally, the simulation results were given to show the effectiveness of the proposed method. The future work will develop a new compensation method to design the perfect tracking control system.

Acknowledgment. The authors would like to thank the National Natural Science Foundation of China (61440031), and Henan Higher Key Scientific Research Project (15A1200 03) for their support of this work.

REFERENCES

[1] M. Shahinpoor and K. Kim, Ionic polymer-metal composites. III. Modeling and simulation as biomimetic sensors, actuators, transducers, and artificial muscles, *Smart Materials and Structures*, vol.13, no.6, pp.1362-1388, 2004.

[2] Z. Chen, R. Hedgepeth and X. Tan, A nonlinear control-oriented model for ionic polymer-metal composite actuators, *Smart Materials and Structures*, vol.18, no.9, pp.55-58, 2009.

[3] A. Banos, Stabilization of nonlinear systems based on a generalized bezout identity, *Automatica*, vol.32, pp.591-595, 1996.

[4] M. Deng, *Operator-Based Nonlinear Control System Design and Applications*, Wiley-IEEE Press, 2014.

[5] M. Deng and A. Wang, Robust nonlinear control design to an ionic polymer metal composite with hysteresis using operator based approach, *IET Control Theory & Applications*, vol.6, no.17, pp.2667-2675, 2012.

[6] A. Wang, G. Wei and H. Wang, Operator based robust nolinear control design to an ionic polymer metal composite with uncertainties and input constraints, *Applied Mathematics & Information Sciences*, vol.8, no.5, pp.2471-2477, 2014.

[7] Z. Gaing, A particle swarm optimization approach for optimum design of PID controller in AVR system, *IEEE Trans. Energy Conversion*, vol.19, no.2, pp.384-391, 2004.

[8] J. Grefenstette, Optimization of control parameters for genetic algorithms, *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, vol.16, no.1, pp.122-128, 2000.

[9] J. Lin and C. Lee, Non-linear system control using a recurrent fuzzy neural network based on improved particle swarm optimisation, *Int. J. Systems Science*, vol.41, no.4, pp.381-395, 2010.

[10] M. Martłnez, S. Garcoa-Nieto, J. Sanchis and X. Blasco, Genetic algorithms optimization for normalized normal constraint method under Pareto construction, *Advances in Engineering Software*, vol.40, no.4, pp.260-267, 2009.

[11] X. Pang and L. J. Rybarcyk, Multi-objective particle swarm and genetic algorithm for the optimization of the LANSCE linac operation, *Nuclear Instruments and Methods in Physics Research A*, vol.16, pp.124-129, 2014.