Particle swarm optimization based networked control system design with uncertainty

Ti-Hung Chen[1,a], Ming-Feng Yeh[2] and Wen-Yo Lee[1]

[1] *Department of Computer Information and Network Engineering, Lunghwa University of Science and Technology, Taoyuan City 33306, Taiwan*
[2] *Department of Electrical Engineering, Lunghwa University of Science and Technology, Taoyuan City 33306, Taiwan*

Abstract. This study discusses the design of state feedback control for networked control system (NCS) with uncertainty, where the feedback controller is used to solve the latency and the packet dropout caused by NCS. The conventional design approach generally adopts Lyapunov stability theorem to analyse the stability of NCSs and then Lyapunov-Krasovskii stability criterion is solved to obtain the control gain matrix. Rather than using complex Lyapunov-Krasovskii stability criterion and solving difficult linear matrix inequality (LMI) problem in the control design, this study attempts to utilize particle swarm optimization (PSO) algorithm to search an optimal feedback control gain for the NCS. Finally, the computer simulation is illustrated the proposed control strategy and demonstrated the control performance.

1 Introduction

Nowadays, increasing attention has been paid to design of networked control systems (NCSs) in which the components (e.g., sensors, actuators, and controllers) are spatially distributed and connected via a share communication network [1-10]. In NCSs, data is transmitted in atomic unit called data packets. To transmit data packets, the system states and output signals are sampled by the sensors, transmitted to the controllers over the network, and finally the controller signals are transmitted to the actuators by the data packets. The actuators would control the controlled plants according to the receiving data packets.

Two major challenges in NCSs are the effects of “latency” and “packet dropout” on the system performance [11]. The former is caused by propagation, transmission, router, processing, computer, etc., and the latter is resulted from network congestion, multi-path fading, faulty network drivers, etc. To treat the mentioned effects, several methods have been proposed. For example, [2, 8] adopted H_∞ control theory to stabilize NCSs and analysed the system robustness. [3] applied state feedback control method to treat the control problem of NCSs. [4] applied Takagi-Sugeno (T-S) fuzzy control technique to guarantee the stability of NCSs.

While the state feedback control applying to NCSs, the conventional approach generally adopts Lyapunov stability theorem to analyse the stability of NCSs. Then solving Lyapunov-Krasovskii

[a] Corresponding author : cth.chen@gmail.com

stability criterion, resulted from Lyapunov stability theorem, could attain the control gain matrix [2, 8]. Generally speaking, Lyapunov-Krasovskii stability criterion is characterized in terms of a linear matrix inequality (LMI) problem or an eigenvalue problem [12]. However, solving LMI or eigenvalue problem is difficult and generally obtains a feasible solution.

Particle swarm optimization (PSO) algorithm, proposed by Kennedy and Eberhart in 1995 [13], was inspired by the social behavior of bird flocking and fish schooling. It works by maintaining a swarm of particles that move around in the search space influenced by the improvements discovered by the other particles. One important advantage of PSO algorithm is that it does not use the gradient of the problem to be optimized. Therefore, PSO algorithm can be readily employed for a host of optimization problems and has received considerable research interest [14-17]. On the other hand, owing to its simplicity of implementation and ability to quickly converge to a reasonably good solution [18], PSO algorithm has been successfully applied in solving many real-world optimization problems.

With the help of randomly searching ability of PSO algorithm, this study attempts to utilize PSO algorithm to search the optimal control gain matrix for the NCS with uncertainty, instead of solving complex LMI or eigenvalue problem. The computer simulation is illustrated the proposed control strategy and demonstrated the corresponding control performance.

This study is organized as follows: The problem formulation of NCS is given in section 2. Section 3 presents the control scheme of PSO based NCS. In section 4, computer simulation will be given to illustrate the control performance of the proposed control strategy. Section 5 concludes the paper.

2 Problem formulations

Figure 1 illustrates a class of the networked control system which is consisted of the following five parts: a sensor, a controller, a zero-order-hold (ZOH), a communication network channel, and a controlled plant [18]. In the figure, the ZOH plays a role of an actuator. Consider the following controlled plant with uncertainty [8, 19]:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \Delta\mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \boldsymbol{\omega}(t) \tag{1}$$

where $\mathbf{x}(t) = [x(t), \dot{x}(t), ..., x^{(n)}(t)] \in \Re^n$ is the state vector, $\mathbf{u}(t) \in \Re^m$ is the control signal, $\boldsymbol{\omega}(t) \in \Re^n$

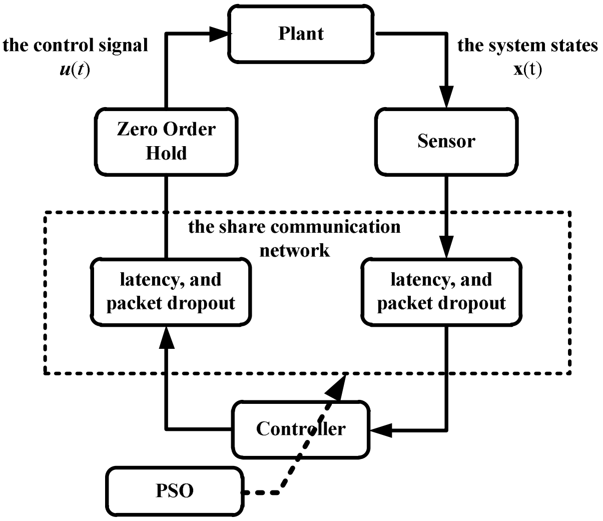
Plant
the control signal
$u(t)$
the system states
x(t)
Zero Order Hold
Sensor
the share communication network
latency, and packet dropout
latency, and packet dropout
Controller
PSO

Figure 1. The framework of PSO-NCS.

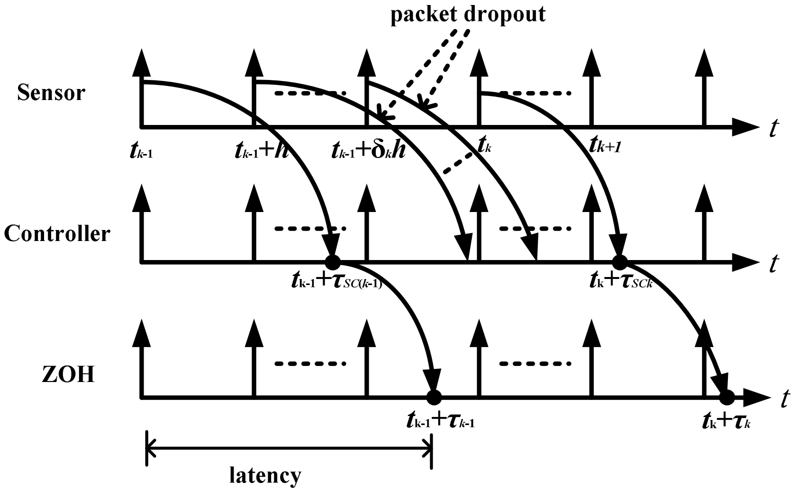

Figure 2. Latency and packet dropout of NCS.

is the unknown external disturbance, $\mathbf{A} \in \Re^{n \times n}$ and $\mathbf{B} \in \Re^{n \times m}$ are the known state matrices, and $\Delta\mathbf{A} \in \Re^{n \times n}$ is the uncertainty state matrix. The initial state vector is given as $\mathbf{x}(t_0) = \mathbf{x}_0$. Without loss of generality, we make the following assumptions [8, 19]:

Assumption 1: $(\mathbf{A}, \mathbf{B})$ is controllable pair.

Assumption 2: The external disturbance $\boldsymbol{\omega}(t)$ is bounded. That is, $\boldsymbol{\omega}(t) \in L_2[0, t_f], \ \forall t_f \in [0, \infty)$.

Assumption 3: $\Delta\mathbf{A}$ is unknown, but bounded.

In the NCSs, the controller is connected with the sensor (or ZOH) via a share communication network channel (consisting of propagation, transmission, router, processing, computer, storage delays, etc.). The use of a share communication network will lead to latency. In addition, network congestion, multi-path fading, and faulty network drivers would cause the packet dropout. Latency and packet dropout might be potential sources to instability and poor performance of NCSs.

Figure 2 shows latency and packet dropout of NCSs. Because of latency and packet dropout, packets are transmitted from the sensor to the controller spend τ_{sck} , and from the controller to the actuator spend τ_{czk} . Define the total latency as $\tau_k = \tau_{sck} + \tau_{czk}$ at every sampling instant t_k and make the following assumptions for the total latency and packet dropout [2, 8, 20]:

Assumption 4: The total latency τ_k is bounded.

Assumption 5: The packet dropout number η_k is bounded as $0 \le \eta_k \le \eta_{\max}$, where $\eta_{\max}$ is upper bound of η_k .

Here, we adopt the state feedback control into inhibit the effect of the latency and the packet dropout such that the state vector $\mathbf{x}(t)$ of NCS would converge to zero, that is $\lim_{t \to \infty} \|\mathbf{x}(t)\| = 0$. In the state feedback control scheme, the controller is designed as follows:

$$\mathbf{u}(t) = \mathbf{K}\mathbf{x}(t) \tag{2}$$

where $\mathbf{K}$ is the control gain matrix and is obtained by solving the Lyapunov-Krasovskii stability criterion. Solving the stability criterion generally requires calculating the complex LMI problem. It is a difficult task sometimes. Rather than using LMI solver, this study attempts to apply PSO algorithm to obtain the control gain matrix $\mathbf{K}$.

3 PSO based NCS design

PSO algorithm uses a simple mechanism that imitates their swarm behaviors to guide the particles to search for globally optimal solutions. In PSO algorithm, each potential solution is regarded as a particle associated with two vectors, i.e., the velocity vector and the position vector. During the evolutionary process, the trajectory of each individual is adjusted by dynamically altering the velocity of each particle, according to its own flying experience (*pBest*) and the flying experience of the other particles (*gBest*) in the search space. With such a search mechanism, the particles have a tendency to fly towards better search area over the course of search process.

In past, the optimal control parameters used in NCSs are obtained by solving the Lyapunov-Krasovskii stability criterion. Such a solving process is complex and generally obtains a feasible solution. In addition, the network communication quality is also unpredictable. Thus, it is impossible to describe latency and packet dropout of NCSs accurately. Sequentially, the gradient of system state cannot obtain. With the help of randomly searching ability of PSO algorithm, this study applies PSO algorithm to search the optimal control gain matrix $\mathbf{K}$, instead of solving the complex Lyapunov-Krasovskii stability criterion. The following procedure can be used for searching the optimal control gain matrix with PSO algorithm.

Step 1: Initialize the swarm by assigning a random position $\mathbf{K}_i^o$ and the corresponding velocity vector $\mathbf{v}_i^o$ to each particle, where $i = 1,2,..., N$, and *N* represents the size of the swarm. Also determine the maximum number of generations J_{max}.

Step 2: Evaluate the fitness value for each particle. In this study, the fitness function is defined as

$$F(\mathbf{K}_i^j) = f(\mathbf{x}(t), \mathbf{u}(t)) = f(\mathbf{x}(t), \mathbf{K}_i^j \mathbf{x}(t)) \tag{3}$$

where $\mathbf{K}_i^j$ denotes the *i*th particle for the *j*th generation.

Step 3: Update the best position for each particle $\mathbf{K}_i^{pBest}$ and the global best position $\mathbf{K}^{gBest}$ by (4) and (5), respectively.

$$\mathbf{K}_i^{pBest} = \arg \max_{h=0,1,...,j} F(\mathbf{K}_i^h) \tag{4}$$

$$\mathbf{K}^{gBest} = \arg \max_{\substack{h=0,1,...,j;\\ i=1,2,...,N}} F(\mathbf{K}_i^h) \tag{5}$$

where $\mathbf{K}_i^{pBest}$ represents the position with the best fitness found so far for the *i*th particle and $\mathbf{K}^{gBest}$ is the best position discovered by the swarm.

Step 4: Update the position and the corresponding velocity for each particle using

$$\mathbf{v}_i^{j+1} = w \cdot \mathbf{v}_i^j + c_1 \cdot \phi_1 \cdot (\mathbf{K}_i^{pBest} - \mathbf{K}_i^j) + c_2 \cdot \phi_2 \cdot (\mathbf{K}^{gBest} - \mathbf{K}_i^j) \tag{6}$$

$$\mathbf{K}_i^{j+1} = \mathbf{K}_i^j + \mathbf{v}_i^{j+1} \tag{7}$$

where w is the inertia factor, c_1 is the “self-confidence” factor, c_2 is the “swarm confidence” factor, and ϕ_1 and $\phi_2 \cdot$ are two uniformly distributed random numbers independently generated within [0, 1].

Step 5: Repeat Steps 2-4 until a stopping criterion is fulfilled (e.g., the maximum number of generations or the goal is reached). And, $\mathbf{K}^{gBest}$ is the optimal control gain matrix.

4 Simulation result

In order to demonstrate the performance of the proposed control strategy, an inverted pendulum control system [8, 21] is used verify it. Let $x_1(t)$ be the angular of the pendulum with respect to the vertical line and $u(t)$ be the applied the control signal. Define $\mathbf{x}(t) = [x_1(t), \dot{x}_1(t)]^T = [x_1(t), x_2(t)]^T$, the dynamic equations of the inverted pendulum system can be described as follows:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \frac{g\sin(x_1) - (m_p l x_2^2 \sin(x_1)\cos(x_1)/M)}{(4l/3) - (m_p l\cos^2(x_1)/M)} + \frac{\cos(x_1)u/M}{(4l/3) - (m_p l\cos^2(x_1)/M)} + \omega(t) \end{aligned} \tag{8}$$

where m_c is the mass of the cart, m_p is the mass of the pole, $M = m_c + m_p$, $g = 9.8\ m/s^2$ is the acceleration due to gravity, and l is the half length of the pole. In this study, the parameters are set as $m_c = 1\ kg$, $m_p = 0.1\ kg$, $l = 0.5\ m$, and the sampling period as 0.001sec. In addition, the unknown external disturbance $\omega(t)$ is a uniformly distributed random number independently generated within [–0.1, 0.1] and the initial state as $\mathbf{x}(0) = [\pi/6,\ 0]^T$. The Jacobian linearized model around the equilibrium $\mathbf{x}(0) = [0,\ 0]^T$ can be obtained directly from the dynamic equations (8), which yields

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \dfrac{g}{(4l/3) - m_p l/M} & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \Delta\mathbf{A} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{a}{(4l/3) - m_p l/M} \end{bmatrix} u + \begin{bmatrix} 0 \\ \omega(t) \end{bmatrix} \tag{9}$$

where $\Delta\mathbf{A}$ is be regarded as the uncertainty terms due to latency, packet dropout and linearization error. In the simulation, each entry in $\Delta\mathbf{A}$ is a uniform random number between –0.1 and 0.1.

In the simulations, three PSO variants, i.e., the standard PSO (Std. PSO), PSO with a linearly varying inertia weight (PSO-LVIW, [22, 23]), and PSO with time-varying acceleration coefficients (PSO-TVAC, [24]), were tested using the same swarm size of 100. Each algorithm was run 50 independent trials with the same maximum number of generations $J_{max} = 100$ for each trial and their worst, mean and best results are used in the comparison. Shi and Eberhart [22, 23] suggested that when w is medium ($0.8 < w < 1.2$), the PSO will have the best chance to find the global minimum, but also takes a moderate number of generations. Besides, as mentioned in Kennedy and Eberhart [13], a recommended choice for the factors c_1 and c_2 is a fixed value of 2 since it on average makes the weights for "social" and "cognition" parts to be 1. This setting also has been adopted by many other researchers. With the help of the above suggestions, the parameters used for the standard PSO is set as $w = 1$ and $c_1 = c_2 = 2$. As suggested in Shi and Eberhart [23], the inertia weight for PSO-LVIW was set to change from 0.9 to 0.4 over the generations. Besides, the boundary values of acceleration coefficients for HPSO-TVAC were set as $c_{1,0} = 2.5$, $c_{1,f} = 0.5$, $c_{2,0} = 0.5$, and $c_{2,f} = 2.5$, which are the best ranges for c_1 and c_2 suggested by Ratnaweera [24]. In the simulation, the PSO based NCS is also compared with the real coded genetic algorithm (RCGA) based NCS, where RCGA makes use of uniform arithmetical crossover with the crossover rate of 0.8 and uniform mutation with mutation rate of 0.1 [25]. Finally, the fitness function is defined as

$$F(\mathbf{K}_i^j) = \int_0^T (x_1^2(t) + 0.0005u^2(t))dt \tag{10}$$

where $T = 5$ in the simulation.

All the programs coded by Matlab version R2014b were executed by a personal computer with Intel® Core™ i5-3470 CPU @ 3.20/3.60 GHz processor, 16.0-GB RAM, and Windows 7 with service pack 1 operating system.

The best, worse and mean fitness of the best-of-run solutions for each algorithm are presented in Table 1, where Std. Dev. represents the corresponding standard deviation. In addition, the optimal control gain matrix associated with the best fitness for each algorithm is given in the last column of Table 1. Since the best fitness values obtained by different algorithm are almost the same, their responses, including the curve of control signal *u*(*t*) and the trajectories of $x_1(t)$ and $x_2(t)$, are close to each other. Figures 3 and 4 therefore only illustrates the trajectories of $x_1(t)$ and $x_2(t)$, respectively, obtained by the standard PSO algorithm, where the solid and dotted lines represents the trajectories corresponding to the best and worst fitness, respectively. The corresponding control signal *u*(*t*) is shown in Figure 5. The simulation results show that the proposed control strategy implemented by each evolution algorithm could perform good stability and inhibit effectively the effect of "latency" and "data packet dropout".

Table 1. Performance comparison on fitness value and the optimal gain.

	Fitness				Optimal gain
	Best	Worst	Mean	Std. Dev.	
Std. PSO	6.0482	6.6321	6.2341	0.1654	[-54.0255, -8.3900]
PSO-LVIW	6.0488	6.8441	6.1844	0.1469	[-52.3876, -8.0424]
PSO-TVAC	6.0471	6.7676	6.2315	0.2023	[-54.4973, -8.4962]
RCGA	6.0469	NaN	(6.0469)*	(0.0000)*	[-53.9474, -8.3808]

*Without regarding a failed run.

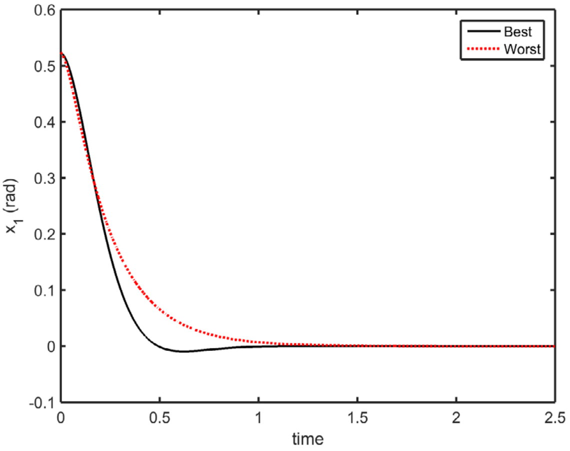

Figure 3. The trajectory of $x_1(t)$.

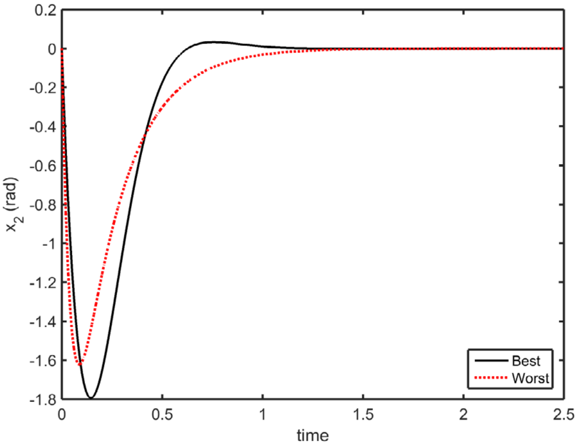

Figure 4. The trajectory of x2(t).

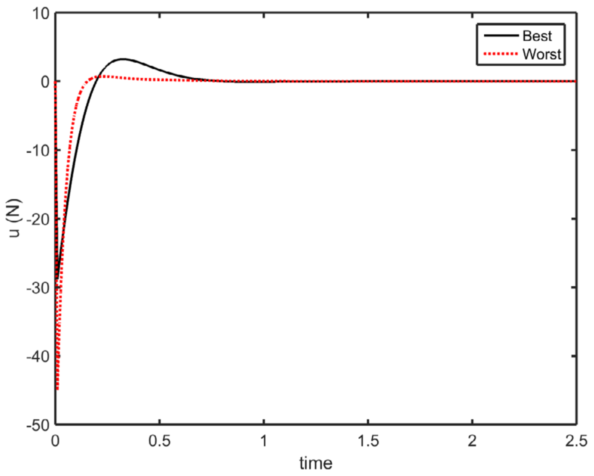

Figure 5. The control signal u(t).

In order to compare the convergence speeds and reliabilities (success performance) of different algorithms, this study selects a threshold value of 6.6 to determine whether a trial (run) is successful or not. The success of an algorithm means that the algorithm can result in a fitness value no worse than the prespecified threshold, with the number of generations less than the corresponding maximum number. The success rate is calculated as the number of successful runs divided by the total number of runs. Table 2 lists the success performance (SP) and average computational time of each algorithm required to reach such a predefined threshold, where FEs represents the number of function evaluations. In addition, the success performance is defined as SP = mean (FEs for successful runs)*(# of total runs)/(# of successful runs) [26]. As can be seen, PSO-LVIW and RCGA yield the best success rate of 98.0%, but the standard PSO achieves the best SP. Although PSO-TVAC has the worst success rate, it also performs an acceptable reliability of 92.0%. Besides, the standard PSO performs significantly faster than PSO-LVIW and RCGA on the average computational time. To sum up, with almost the same best fitness value, the standard PSO with $w = 1$ and $c_1 = c_2 = 2$ outperforms the other three approaches on the efficiency and reliability for the considered NCS with uncertainty.

Table 2. Comparisons on success performance and convergence speed.

	Rate (%)	FEs	SP	Run time(sec.)
Std. PSO	96.00	95.78	**99.7708**	**0.7609**
PSO-LVIW	**98.00**	97.58	99.5714	0.9004
PSO-TVAC	92.00	**91.66**	99.6304	0.8098
RCGA	**98.00**	97.42	99.4082	1.0051

5 Conclusions

Rather than using complex Lyapunov-Krasovskii stability criterion and solving the difficult LMI problem, this study applies the PSO algorithm to search the optimal control gain matrix for NCS with uncertainty, where the feedback controller is used to compensate the latency and packet dropout. The simulation results on an inverted pendulum system show that the PSO based NCS could perform good stability and inhibit effectively the effect of latency and data packet dropout. In addition, the optimal control gain obtained by the standard PSO performs the best success performance and the least computational time on the considered problem.

References

1. G.C. Walsh, H. Ye, and L.G. Bushnell, IEEE Trans. Contr. Syst. Technol., **10** (3), 438-446 (2002)
2. P. Seiler and R. Sengupta, IEEE Trans. Autom. Control, **50** (3), 356-364 (2005)
3. D. Huang and S.K. Nguang, IEEE Tran. Autom. Control, **53** (3), 829-834 (2008)
4. C. Peng and T.C. Yang, IEEE Trans. Fuzzy System, **18** (2), 326-335 (2010)
5. H. Wu, L. Lou, C.C. Chen, S. Hirche, and K. Kühnlenz, IEEE Trans. Ind. Electron., **60** (2), 554-566 (2013)
6. Y. Shi, J. Huang, and B. Yu, IEEE Trans. Ind. Electron., **60** (12), 5864-5874 (2013)
7. R. Yang, G.P. Liu, P. Shi, C. Thomas, and M.V. Basin, IEEE Trans. Ind. Electron., **61** (1), 512-520 (2014)
8. T.H. Chen, Applied Mathematics & Information Sciences, **9** (1L), 133-139 (2015)
9. H. Li, G. Wu, P. Shi, and Y. Gao, IEEE Tran. Cybern., **45** (11), 2378-2389 (2015)
10. J. Qiu, H. Gao, and S.X. Ding, IEEE Trans. Ind. Electron., **63** (2), 1207-1217 (2016)
11. W. Stallings, *Data and Computer Communications Prentice Hall*, **9(e)** (Prentice Hall, N.J., 2010)
12. S. Boyd, L.E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory* (PA: SIAM, Philadelphia, 1994)
13. J. Kennedy and R. Eberhart, IEEE Int. Conf. Neural Netw., 1942-1948, (1995)
14. F. Afshinmanesh, A. Marandi, and M. Shahabadi, IEEE Trans. Antennas Propagat., **56** (7), 1845-1852, (2008)
15. R.R. Pitre, X.R. Li, and R. Delbalzo, IEEE Trans. Aerosp. Electron. Syst., **48** (3), 2551-2565, (2012)
16. N.V. George and G. Panda, IEEE Trans. Instrum. Meas., **61** (12), 3378-3386, (2012)
17. T. Zhou and W. Sun, IEEE Trans. Sustainable Energy, **5** (2), 408-415, (2014)
18. L.X. Zhang, H.J., Gao, and K.N. Okyay, IEEE Trans. Ind. Inform., **9** (1), 403-416 (2013)
19. Q.C. Zhong, D. Rees, and J. Dyn. Syst. Meas. Control, **126**, 905-910 (2004)
20. Z.H. Pang, G.P. Liu, D. Zhou, and M. Chen, IEEE Trans. Ind. Electron., **61** (9), 4867-4877 (2014)
21. K. Tanaka and H.O. Wang, *Fuzzy Control Systems Design and Analysis*: *A Linear Matrix Inequality Approach* (Wiley & Sons, Inc., N.J. 2001)
22. Y. Shi and R.C. Eberhart, IEEE World Congr. Comput. Intell., 69-73 (1998)
23. Y. Shi and R.C. Eberhart, *Proc. IEEE Int. Congr. Evol. Comput.*, Washington, DC, **3**, 101-106 (1999)
24. A. Ratnaweera, S. Halgamuge, and H. Watson, IEEE Trans. Evol. Comput., **8** (3), 240-255 (2004)
25. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs* (Springer-Verlag, New York, 1992)
26. A. K. Qin, V.L. Huang, and P.N. Suganthan, IEEE Trans. Evol. Comput., **13** (2), 398-417 (2009)