Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 928343, 7 pages
http://dx.doi.org/10.1155/2015/928343

Research Article

State Feedback H_∞ Control of Power Units Based on an Improved Particle Swarm Optimization

Zhongqiang Wu and Xuefei Xiao

Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China

Correspondence should be addressed to Zhongqiang Wu; mewzq@163.com

Received 28 June 2015; Revised 31 August 2015; Accepted 13 September 2015

Academic Editor: Marco Mussetta

A new state feedback H_∞ control scheme is presented used in the boiler-turbine power units based on an improved particle swarm optimizing algorithm. Firstly, the nonlinear system is transformed into a linear time-varying system; then the H_∞ control problem is transformed into the solution of a Riccati equation. The control effect of H_∞ controller depends on the selection of matrix P, so an improved particle swarm optimizing (PSO) algorithm by introducing differential evolution algorithm is used to solve the Riccati equation. The main purpose is that mutation and crossover are introduced for a new population, and the population diversity is improved. It is beneficial to eliminate stagnation caused by premature convergence, and the algorithm convergence rate is improved. Finally, the real-time optimizing of the controller parameters is realized. Theoretical analysis and simulation results show that a state feedback H_∞ controller can be obtained, which can ensure asymptotic stability of the system, and the double objectives of stabilizing system and suppressing the disturbance are got. The system can work well over a large range working point.

1. Introduction

Boiler and turbine are main equipment of coal-fired power generation units, so the coordination control plays an important role in improving the performance of the generation system. The nonlinear control method which has been used in the coordinated control system can solve the optimizing operation problem over large range working point effectively. It becomes the hot topic in recent years. Backstepping method [1, 2] has better rapidity and guarantees the stability of system. Adaptive decoupling control is used [3], and the good tracking and adaptive ability of system is shown. The terminal sliding mode controller can improve the precision of the system [4].

In this paper, firstly, the nonlinear coordinate system for boiler and steam turbine is transformed into a linear time-varying system with external disturbance. Then the H_∞ control problem can be transformed into the solution of Riccati equation [5, 6]. Since the selection of the Riccati equation solution has a great influence on the control effect, the improved particle swarm optimizing (PSO) algorithm is applied to seek the solution of Riccati equation. Particle swarm optimizing (PSO) algorithm [7–9] can adaptively adjust parameters according to speed vector, which is different from other evolution algorithms. Each particle executes a kind of "consciousness" variation in the evolutionary process, namely, particles only toward some good directions flight according to the experience of the populations, so that the PSO algorithm has more opportunities to fly to more optimal solution area faster. The other advantage of PSO algorithm is ease of implementation, because there are very few parameters needed to be adjusted. PSO algorithm is able to expand the scope of the feasible solution and ensure the existence of the solution. In order to eliminate the stagnation and avoid premature convergence to stop, an improved particle swarm optimizing algorithm is adopted by introducing differential evolution algorithm [10–13], which guarantees the population converging to the global optimal solution continuously. Finally, the real-time optimizing of the controller parameters is realized. Theoretical analysis and simulation results show that the H_∞ controller obtained can ensure asymptotic stability of system and the double objectives of stabilizing system and suppressing the disturbance are reached. The system can work well over a large range working point.

2. The Linearization of Boiler-Steam Turbine Model

The model of boiler-turbine is the foundation of coordinate control. The Astrom model [14] is a very typical three-order MIMO nonlinear coordination system model of machine furnace, which has been quoted widely. The model can be described as the following dynamic equations:

$$\begin{aligned} \frac{dp}{dt} &= -0.0018u_2p^{9/8} + 0.9u_1 - 0.15u_3, \\ \frac{dP_0}{dt} &= \left(0.073u_2 - 0.016\right)p^{9/8} - 0.1P_0, \\ \frac{d\rho_f}{dt} &= \frac{\left(141u_3 - \left(1.1u_2 - 0.19\right)p\right)}{85}, \end{aligned} \tag{1}$$

where p is the drum pressure, P_0 is the output power, ρ_f is the fluid density inside the system, u_1 is the opening degree of the fuel flow regulator, u_2 is the opening degree of the steam flow regulator, and u_3 is the opening degree of the water flow regulator.

Considering the actual situation of the regulator and dynamic process itself, the opening position of the regulator and its changing rate are limited [14].

The opening degrees of the three regulators should satisfy the conditions below:

$$\begin{aligned} &\left|\frac{du_1}{dt}\right| \le 0.007/\mathrm{s}, \quad 0 \le u_1 \le 1; \\ &-2/\mathrm{s} \le \frac{du_2}{dt} \le 0.02/\mathrm{s}, \quad 0 \le u_2 \le 1; \\ &\left|\frac{du_3}{dt}\right| \le 0.05/\mathrm{s}, \quad 0 \le u_3 \le 1. \end{aligned} \tag{2}$$

The output of the system is the drum pressure p (kg/cm^2), the output power P_0 (MW), and the drum pressure X_W (m). The drum pressure and the output power are two state variables and the drum water level is given by the following equation:

$$X_W = 0.05\left[0.13073\rho_f + 100\alpha_{\mathrm{cs}} + \frac{q_e}{9} - 67.975\right], \tag{3}$$

where α_{cs} is the coefficient of steam quality and q_e (kg/s) is evaporation rate:

$$\begin{aligned} \alpha_{\mathrm{cs}} &= \frac{\left(1 - 0.001538\rho_f\right)\left(0.8p - 25.6\right)}{\rho_f\left(1.0394 - 0.0012304p\right)}, \\ q_e &= \left(0.854u_2 - 0.147\right)p + 45.59u_1 - 2.514u_3 \\ &\quad - 2.096. \end{aligned} \tag{4}$$

Set

$$\begin{aligned} x &= \left[x_1 \;\; x_2 \;\; x_3\right]^T = \left[p \;\; P_0 \;\; \rho_f\right]^T, \\ u &= \left[u_1 \;\; u_2 \;\; u_3\right]^T, \\ y &= \left[y_1 \;\; y_2 \;\; y_3\right]^T = \left[p \;\; P_0 \;\; X_w\right]^T. \end{aligned} \tag{5}$$

At the selected working point (x^0, u^0), the nonlinear function is expanded into a Taylor series and the first-order derivative item holds; the linear time-varying model can be got:

$$\begin{aligned} \dot{\overline{x}} &= A\overline{x} + B\overline{u}, \\ \overline{y} &= C\overline{x} + D\overline{u}, \end{aligned} \tag{6}$$

where $\overline{x} = x - x^0$, $\overline{y} = y - y^0$, $\overline{u} = u - u^0$, and the expression of A, B, C, and D could be obtained by partial derivative operation:

$$\begin{aligned} A &= \begin{bmatrix} -0.002025u_2x_1^{1/8} & 0 & 0 \\ \frac{9}{8}\left(0.073u_2 - 0.016\right)x_1^{1/8} & -0.1 & 0 \\ -\frac{1.1u_2 - 0.19}{85} & 0 & 0 \end{bmatrix}, \\ B &= \begin{bmatrix} 0.9 & -0.0018x_1^{9/8} & 0.15 \\ 0 & 0.073x_1^{9/8} & 0 \\ 0 & -0.01294x_1 & 0 \end{bmatrix}, \\ C &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_{31} & 0 & c_{33} \end{bmatrix}, \\ D &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.25328 & 0.0047444x_1 & -0.013967 \end{bmatrix}, \end{aligned} \tag{7}$$

where $x = x^0$, $u = u^0$, and

$$\begin{aligned} c_{31} &= 0.05\left[\frac{80.002\left(1 - 0.001538x_3\right)}{x_3^2\left(1.0394 - 0.0012304x_1\right)^2} \right. \\ &\quad \left. + \frac{0.854u_2 - 0.147}{9}\right], \\ c_{33} &= 0.05\left[0.13073 - \frac{100\left(0.8x_1 - 25.6\right)}{x_3^2\left(1.0394 - 0.0012304x_1\right)}\right]. \end{aligned} \tag{8}$$

The coefficient matrix of model above will change with the states of working points selected, so that the whole system is transformed into a linear system with time-varying parameters. The opening degrees of the three regulators are restrained by the following conditions:

$$\begin{aligned} &\left|\frac{d\overline{u}_1}{dt}\right| \le 0.007/\mathrm{s}, \quad -0.34 \le \overline{u}_1 \le 0.66; \\ &-2/\mathrm{s} \le \frac{d\overline{u}_2}{dt} \le 0.02/\mathrm{s}, \quad -0.69 \le u_2 \le 0.31; \\ &\left|\frac{d\overline{u}_3}{dt}\right| \le 0.05/\mathrm{s}, \quad -0.435 \le u_3 \le 0.565. \end{aligned} \tag{9}$$

3. State Feedback H_∞ Control

The actual system with external disturbance is considered, so the state feedback H_∞ control scheme is used to realize the stable control of machine furnace coordinate system. The linear time-varying model of (6) is expressed as follows:

$$\dot{\overline{x}} = A\overline{x} + B\overline{u} + B_1 w,$$
$$\overline{y} = C\overline{x} + D\overline{u}, \tag{10}$$

where $w \in R^{3\times 1}$ is the external disturbance and $B_1 \in R^{3\times 3}$ is the corresponding coefficient matrix.

The H_∞ control problem above is equivalent to design a feedback control law

$$\overline{u} = K\overline{x} + r \tag{11}$$

which can ensure the correspondingly closed-loop system stable ($(A + BK)$ stable), and

$$\left\| T_{yw}(s) \right\|_\infty < \gamma, \tag{12}$$

where $T_{yw}(s)$ represents the closed-loop transfer function matrix form w to y, $\gamma > 0$ is a given constant, $K \in R^{3\times 3}$ is the control matrix, and $r \in R^{3\times 1}$ is the reference input.

Assumption 1 ((A, B) is internally stable). For the system (10) meeting Assumption 1, the necessary and sufficient conditions of the H_∞ control question above are that for a constant $\varepsilon > 0$ and a positively definite matrix Q the Riccati equation

$$\begin{aligned}&\left(A - BH_F D^T C\right)^T P + P\left(A - BH_F D^T C\right)\\&\quad - \frac{1}{\varepsilon} PB\Phi_F^T \Phi_F B^T P - PBH_F B^T P + \gamma^{-2} PB_1 B_1^T P\\&\quad + C^T\left(I - DH_F D^T\right)C + Q = 0\end{aligned} \tag{13}$$

has positive-definite symmetric solution $P > 0$. If the system (13) has solution, then closed-loop system is internally stable. The state feedback matrix of (12) is given by the following equation:

$$K = -\left(\frac{1}{2\varepsilon}\Phi_F^T \Phi_F + H_F\right) B^T P - H_F D^T C, \tag{14}$$

where $H_F = \Sigma^T(\Sigma\Sigma^T)^{-1}(U^T U)^{-1}(\Sigma\Sigma^T)^{-1}\Sigma$, U and Σ are any matrix to satisfy under equations [5] $D = U\Sigma$, $U \in R^{3\times 1}$, $\Sigma \in R^{1\times 3}$, and $\operatorname{rank} U = \operatorname{rank} \Sigma = 1$, where matrix $\Phi_F \in R^{2\times 3}$ satisfied $\Phi_F \Phi_F^T = I$ and $\Phi_F \Sigma^T = 0$.

Combining (10) and (11), the standard state feedback H_∞ control can be got:

$$\begin{bmatrix}\dot{\overline{x}}\\ \overline{y}\end{bmatrix} = G\begin{bmatrix}\overline{x}\\ w\\ \overline{u}\end{bmatrix} = \begin{bmatrix}A & B_1 & B\\ C & 0 & D\end{bmatrix}\begin{bmatrix}\overline{x}\\ w\\ \overline{u}\end{bmatrix},$$
$$\overline{u} = K\overline{x} + r. \tag{15}$$

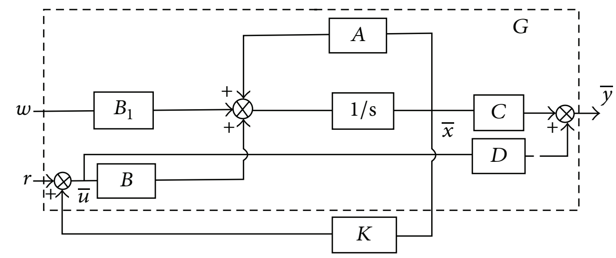

Figure 1: The block diagram of the standard state feedback H_∞ control.

The block diagram of the standard state feedback H_∞ control is shown in Figure 1.

The state feedback H_∞ control problem above can be transformed into the solution of Riccati equation (13). The control effect of H_∞ controller depends on the selection of matrix P; in this paper an improved particle swarm optimizing (PSO) algorithm is used for solving the Riccati equation. This scheme can expand the scope of the feasible solution and ensure the existence of the solution, and the real-time optimizing of the controller parameters is realized.

4. Particle Swarm Algorithm Is Used to Solve the Matrix Equation

In this paper, an improved particle swarm optimization algorithm is applied to seek the solution of Riccati equation, rewriting Riccati equation (13) into the following form:

$$\begin{aligned}&\left(A - BH_F D^T C\right)^T P + P\left(A - BH_F D^T C\right)\\&\quad - \frac{1}{\varepsilon} PB\Phi_F^T \Phi_F B^T P - PBH_F B^T P + \gamma^{-2} PB_1 B_1^T P\\&\quad + C^T\left(I - DH_F D^T\right)C + Q = E,\end{aligned} \tag{16}$$

where γ and ε are the given constants and matrixes H_F, Φ_F, and Q are the given coefficient matrix.

The fitness value of the objective function is

$$\min f(E) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left|E_{ij}\right|, \tag{17}$$

where E_{ij} is the triangle array element of matrix E; the smaller $f(E)$, the closer E to zero matrix. The feasible solution can assure the internally asymptotical stability of the system effectively.

4.1. Basic Particle Swarm Algorithm. The particle swarm algorithm regards each individual as a particle with no weight and volume, which flights in the n-dimensional search space at a certain speed. The dynamic adjustment of the flight speed is based on the flight experience of the individual and the populations. In this paper nine unknown parameters of matrix P needed to be optimized; for the symmetry of P, the dimensions of the search space are six.

The particle number of populations is s; for the ith particle, $X_i(t) = \{x_{i1}(t), x_{i2}(t), \ldots, x_{i6}(t)\}$ and $V_i(t) = \{v_{i1}(t), v_{i2}(t), \ldots, v_{i6}(t)\}$ are the current position and velocity, respectively. The best position of the ith particle is recorded as $Z_i(t) = \{z_{i1}(t), z_{i2}(t), \ldots, z_{i6}(t)\}$, which is also the position with the best fitness value of the ith particle recorded and called *pbest*. For minimizing problems, the smaller the fitness value of objective function, the better the corresponding fitness value.

In this paper set the objective function of minimizing problem to be $f(X)$; the best position of ith particle depends on the following scheme:

$$Z_i(t+1) = \begin{cases} Z_i(t), & f(X_i(t+1)) \geq f(Z_i(t)) \\ X_i(t+1), & f(X_i(t+1)) < f(Z_i(t)). \end{cases} \tag{18}$$

Set the best position of all particles recorded as $Z_g(t)$, which is called the global best position *gbest*; then

$$\begin{aligned} Z_g(t) &\in \{Z_0(t), Z_1(t), \ldots, Z_s(t)\}, \\ f(Z_g(t)) &= \min\{f(Z_0(t)), f(Z_1(t)), \ldots, f(Z_S(t))\}. \end{aligned} \tag{19}$$

The standard evolving equation of particle swarm algorithm can be described as follows:

$$\begin{aligned} V_i(t+1) = \zeta V_i(t) &+ c_1 \mathrm{rand}_1(t)[Z_i(t) - X_i(t)] \\ &+ c_2 \mathrm{rand}_2(t)[Z_g(t) - X_i(t)], \end{aligned} \tag{20}$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{21}$$

where c_1 and c_2 are acceleration constants and $\mathrm{rand}_1(t)$ and $\mathrm{rand}_2(t)$ are the random number within $[0,1]$. ζ is the inertia weight which is similar to the temperature of simulated annealing, the bigger ζ has good global convergence ability, and the smaller ζ has strong local convergence ability. Therefore, with the increase of the number of iterations, the inertia weight should decrease constantly. Thereby, the particle swarm algorithm has strong global convergence ability in early and strong local convergence ability in later period. In this paper self-adjusting of the inertia weight ζ meets

$$\zeta = 0.9 - \frac{t}{\mathrm{MaxNumber}} \times 0.5. \tag{22}$$

MaxNumber is the biggest iterating number.

For (20), the right hand can be divided into three parts: the first part represents the original velocity, the second part is the impact of the best position of the ith particle (*pbest*) on the position of the current particle, and the third part is the impact of the global best position (*gbest*) on the position of the current particle, and the second part and the third part represent the modification of the original velocity.

Traditional PSO algorithm has a potential property which would cause the algorithm entrapping into a local optimum. In addition, the parameters of traditional PSO algorithm have stronger dependence. In order to eliminate the stagnation and avoid premature convergence, the differential evolution (DE) algorithm is introduced, and an improved PSO algorithm guarantees the populations converging to the global optimal solution continuously.

4.2. Differential Evolution Algorithm. DE algorithm has characteristics of remembering individual optimal solution and can share the information in populations [15–18]. DE algorithm initializes the population randomly in the feasible solution space of the problem, by means of mutation and crossover to produce a new population for the current population. Then, an one-to-one selection is made among the two populations consequently to generate a new final population.

In this paper a six-dimensional $X_i(t) = \{x_{i1}(t), x_{i2}(t), \ldots, x_{i6}(t)\}$ is used to denote a particle individual, so in each generation of the populations, the particle number is s. Execute mutation operation to each target individual $X_i(t)$ to obtain corresponding variation individual $M_i(t+1) = \{m_{i1}(t+1), m_{i2}(t+1), \ldots, m_{i6}(t+1)\}$ according to the following equation:

$$M_i(t+1) = X_{r1}(t) + F(X_{r2}(t) - X_{r3}(t)), \tag{23}$$

where F is set in (0, 2], called mutation constant which controls the ratio of the father generation difference vector $(X_{r2} - X_{r3})$ to avoid search stagnation. r_1, r_2, and r_3 are randomly selected from the set $\{1, 2, \ldots, s\}$, which are different from each other and are different from index i. Crossover operation is executed to target individual and its variation individual for acquiring trial individual $U_i(t+1) = \{u_{i1}(t+1), u_{i2}(t+1), \ldots, u_{i6}(t+1)\}$ according to the following equation:

$$u_{ij}(t+1) = \begin{cases} m_{ij}(t+1), & (\mathrm{rand}(j) \leq C_R) \text{ or } j = \mathrm{rand}(n(i)) \\ x_{ij}(t), & \text{otherwise}, \end{cases} \tag{24}$$

where $\mathrm{rand}(j) \in [0,1]$ is uniform distributing random number and $j \in \{1, 2, \ldots, 6\}$ is the jth position coordinate. $\mathrm{rand}(n(i)) \in \{1, 2, \ldots, 6\}$ is a variable index selected randomly, which ensures at least one variable quantity contributed by mutation ones. C_R is a crossover constant within $[0,1]$, which controls the dispersion of the population.

Compare the target individual $X_i(t)$ with the trial individual $U_i(t+1)$ to decide whether the trial individual is a member of the next generation $X_i(t+1)$ according to the following equation:

$$X_i(t+1) = \begin{cases} U_i(t+1), & f(U_i(t+1)) \geq f(X_i(t)) \\ X_i(t), & \text{otherwise}. \end{cases} \tag{25}$$

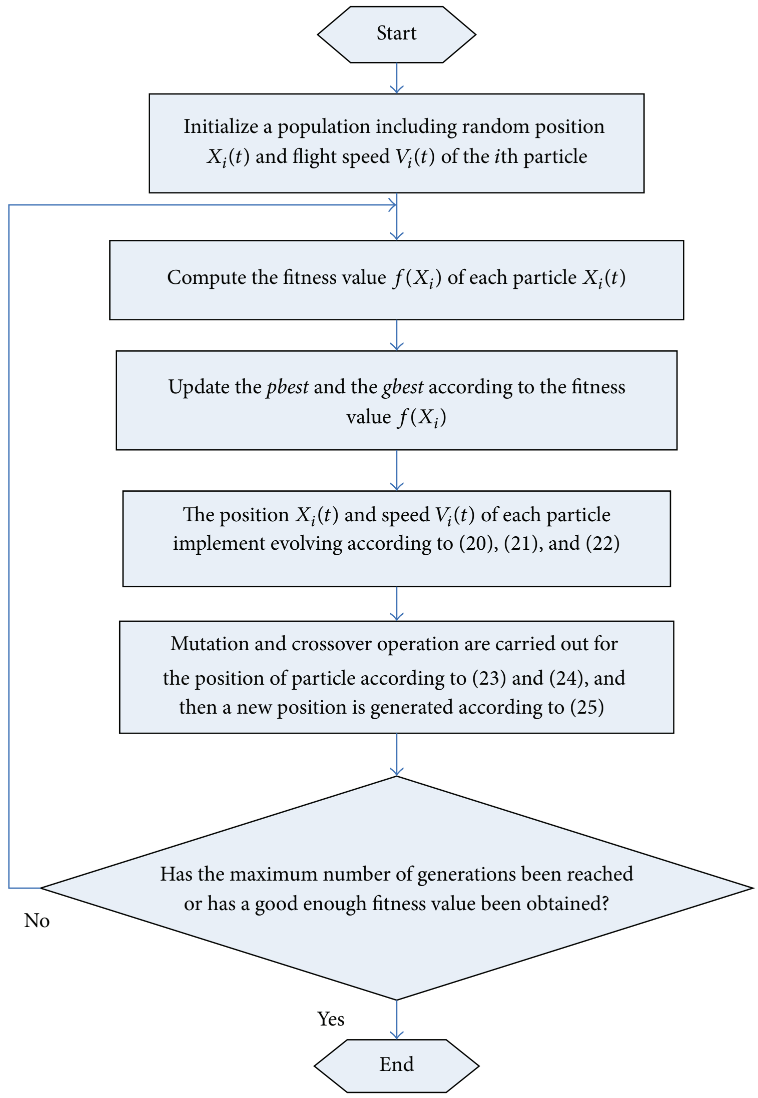

Figure 2: The flow chart of improved particle swarm optimizing algorithm.

4.3. Improved Particle Swarm Optimizing Algorithm. The improved particle swarm optimizing algorithm by introducing differential evolution algorithm can eliminate stagnation and avoid premature convergence, and the procedure is as follows.

Step 1. Initialize a population which contains s particles randomly, and set mutation and crossover constants, and determine the maximum number of generations and the maximum flight velocity.

Step 2. The dimension of the particle is six which is judged by the matrix P. For the ith particle, it has random position $X_i(t)$ and flight speed $V_i(t)$ in the 6-dimensional search space. $V_i(t) = \mathrm{Rand}(t) * V_{\max}$, where $\mathrm{Rand}(t)$ is within $[-1, 1]$. Suppose the initial population is the position with the best fitness value *pbest*.

Step 3. Compute the fitness value $f(X_i)$ of each particle $X_i(t)$, and compare the fitness value of each current particle with its *pbest*. If the current fitness value is smaller than the *pbest* fitness value, regard the current position as current *pbest* position.

Step 4. Compare the fitness value of each current particle with the *gbest*. If the current fitness value is smaller than the *gbest*, regard current position as current *gbest* position.

Step 5. The position and speed of each particle implement evolving according to (20), (21), and (22).

Step 6. Mutation and crossover operation are carried out for the position of particle according to (23) and (24), and then a new position is generated according to (25).

Step 7. If the maximum number of generations has not been reached, or a good enough fitness value has not been obtained, return to Step 3.

The flow chart of improved particle swarm optimizing algorithm is shown in Figure 2.

5. Simulation Research

Set $x^0 = [108.00, 66.65, 428.0]^T$ and $u^0 = [0.340, 0.690, 0.435]^T$ is the initial working point. $c_1 = 1.25$, $c_2 = 1.5$, $s = 50$, $\gamma = 1$, $\varepsilon = 0.01$, and MaxNumber $= 200$ are previously selected. In this paper

$$\begin{gathered} Q = I, \\ \Phi_F = \begin{bmatrix} 0.44 & -0.242 & -0.86 \\ 0.784 & -0.374 & 0.495 \end{bmatrix}, \\ B_1 = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ H_F = \begin{bmatrix} 0.6003 & 1.2165 & -0.0331 \\ 1.2165 & 2.4569 & -0.067 \\ -0.0331 & -0.067 & 0.0018 \end{bmatrix}. \end{gathered} \tag{26}$$

The improved particle swarm algorithm is applied in (10) for simulation test. MATLAB software is adopted. $w = [0.8 \sin 2t, 0.5 \cos t, \sin 2t]^T$ is external disturbance, and the reference input signal of system is taken as

$$\begin{aligned} r_1 &= 108 + 10.8 \times 1(t - 200) + 10.8 \times 1(t - 1000), \\ r_2 &= 66.65 + 18.41 \times 1(t - 600) + 20.74 \\ &\quad \times 1(t - 1400), \\ r_3 &= 0. \end{aligned} \tag{27}$$

At $t = 200$ s, the drum pressure p is stable in 118.8 (kg/cm^2).

At $t = 600$ s, the output power P_0 is stable in 85.06 (MW).

At $t = 1000$ s, the drum pressure p is stable in 129.0 (kg/cm^2).

At $t = 1400$ s, the output power P_0 is stable in 105.8 (MW).

In all process the drum water level is stable in 0 (m), where $1(t - t_0)$ is the unit step function.

The response of the system outputs with two particle swarm optimizing algorithms is shown in Figures 3–5.

It can be seen from Figures 3–5 that the change of drum pressure at 200 s and 1000 s causes the tiny change of output power, but it soon becomes smooth. The change of output power at 200 s and 1000 s combining the change of drum pressure at 200 s and 1000 s causes the tiny change of drum water level, but it soon becomes smooth. The small peak in the changing states is caused by the change of the reference input at different times. It is to show the effect of the decoupling. At the same time there are three disturbances $0.8 \sin 2t$, $0.5 \cos t$, and $\sin 2t$ added in this system. The simulation results are acceptable.

The simulation results show that the method can guarantee the closed-loop system asymptotical stability in a large range working point. The system has strong robustness for external disturbance.

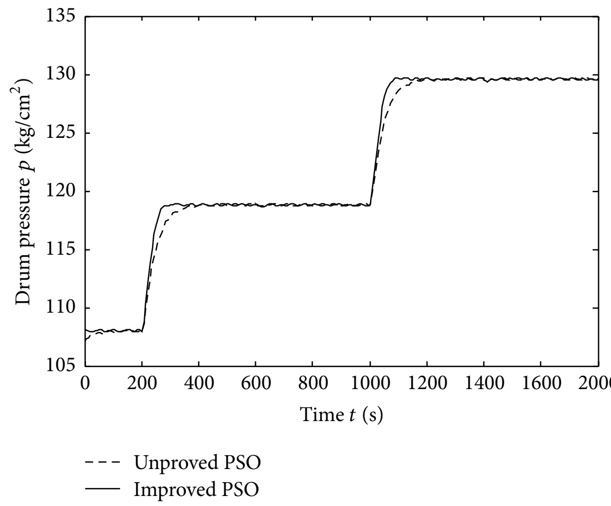

Figure 3: Output response curve of drum pressure.

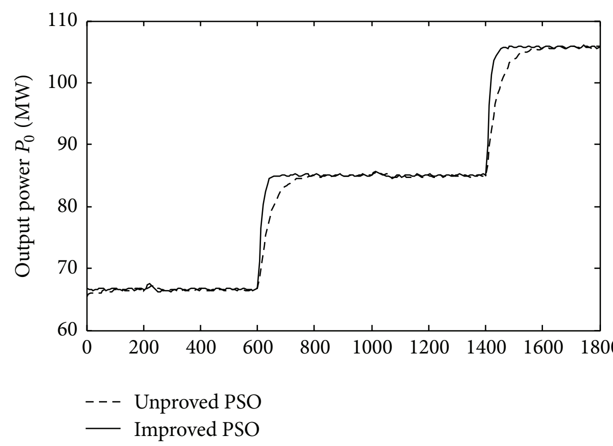

Figure 4: Output power response curve.

6. Conclusions

In this paper, a new state feedback H_∞ control scheme based on an improved particle swarm optimizing algorithm is presented and applied to the boiler-turbine unit with external disturbance. The following results are got:

(1) In this paper, the machine furnace nonlinear coordinate system is translated into a linear time-varying system; then the H_∞ control problem is transformed into the solution of Riccati equation.

(2) With the improved particle swarm algorithm to solve the Riccati equation, the parameters of the controller real-time optimizing are achieved. The method can enhance the precision of optimizing and ensure the stability in large range working point.

(3) The control system has stronger robustness for external disturbance, and the double purposes of stabilizing system and suppressing the disturbance have been realized.

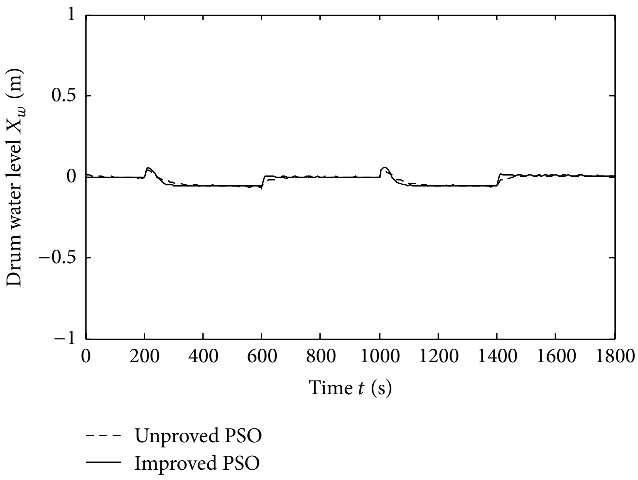

Figure 5: Output response curve of drum water level.

(4) The output responses with nice steady state performance and decoupling effect are got.

Conflict of Interests

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

References

[1] H. Pu and W. Le, "Backstepping PID methods for coordinated control of boiler-turbine units," *Proceedings of the CSEE*, vol. 30, no. 2, pp. 17–22, 2010.

[2] Y. Wang, R. Tian, J. Bai, and Y. Wang, "Nonlinear control for boiler-turbine coordinate systems of power unit based on backstepping," *Journal of North China Electric Power University*, vol. 33, no. 5, pp. 47–55, 2006.

[3] Z. Wang, G. Wang, and H. Chen, "Adaptive decoupling control for thermal power unit based on inverse model," *Proceedings of the CSEE*, vol. 31, no. 29, pp. 118–123, 2011.

[4] D.-H. Zou and B.-H. Wang, "Integrated adaptive terminal sliding mode controller for turbo-generator," *Power System Protection and Control*, vol. 38, no. 17, pp. 33–36, 2010.

[5] T. Surinkaew and I. Ngamroo, "Robust power oscillation damper design for DFIG-based wind turbine based on specified structure mixed H_2/H_∞ control," *Renewable Energy*, vol. 66, pp. 15–24, 2014.

[6] L. Jin and Z. C. Li, "Solving Riccati matrix algebraic equation with an improved genetic algorithm," *The Modern Machinery*, no. 5, pp. 36–39, 2011.

[7] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[8] Z. Wu, W. Jia, L. Zhao, and C. Wu, "Short-term wind speed forecast based on B-spline neural network optimized by PSO," *Mathematical Problems in Engineering*, vol. 2015, Article ID 278635, 7 pages, 2015.

[9] Y. Marinakis, M. Marinaki, and G. Dounias, "A hybrid particle swarm optimization algorithm for the vehicle routing problem," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 463–472, 2010.

[10] A. A. Abou El Ela, M. A. Abido, and S. R. Spea, "Optimal power flow using differential evolution algorithm," *Electric Power Systems Research*, vol. 80, no. 7, pp. 878–885, 2010.

[11] N. N. Son and H. P. Anh, "Adaptive displacement online control of shape memory alloys actuator based on neural networks and hybrid differential evolution algorithm," *Neurocomputing*, vol. 166, pp. 464–474, 2015.

[12] H. Beirami, A. Z. Shabestari, and M. M. Zerafat, "Optimal PID plus fuzzy controller design for a PEM fuel cell air feed system using the self-adaptive differential evolution algorithm," *International Journal of Hydrogen Energy*, vol. 40, no. 30, pp. 9422–9434, 2015.

[13] J. Smrekar, M. Assadi, M. Fast et al., "Design of mixed H_2/H_∞ optimal control systems using multi objective different evolution algorithm," *Energy*, vol. 2, no. 34, pp. 144–152, 2009.

[14] M. Sayed, S. M. Gharghory, and H. A. Kamal, "Gain tuning PI controllers for boiler turbine unit using a new hybrid jump PSO," *Journal of Electrical Systems and Information Technology*, vol. 2, no. 1, pp. 99–110, 2015.

[15] M. Mussetta, P. Pirinoli, S. Selleri, and R. E. Zich, "Meta-PSO for multi-objective EM problems," in *Multi-Objective Swarm Intelligent Systems*, vol. 261 of *Studies in Computational Intelligence*, pp. 125–150, Springer, Berlin, Germany, 2010.

[16] S. Selleri, M. Mussetta, P. Pirinoli, R. E. Zich, and L. Matekovits, "Differentiated meta-PSO methods for array optimization," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 1, pp. 67–75, 2008.

[17] M. Mussetta, P. Pirinoli, S. Selleri, R. E. Zich, and L. Matekovits, "Improved Particle Swarm Optimization algorithms for electromagnetic optimization," *Journal of Intelligent and Fuzzy Systems*, vol. 19, no. 1, pp. 75–84, 2008.

[18] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.