Simulation Modelling Practice and Theory 26 (2012) 77–95

Contents lists available at SciVerse ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Offset-free fuzzy model predictive control of a boiler–turbine system based on genetic algorithm

Yiguo Li [a,*], Jiong Shen [a], Kwang Y. Lee [b], Xichui Liu [a]

[a] *Key Laboratory of Energy Thermal Conversion and Control of Ministry of Education, School of Energy and Environment, Southeast University, Nanjing 210096, China*
[b] *Department of Electrical and Computer Engineering, Baylor University, Waco, TX 76798-7356, USA*

A R T I C L E   I N F O

Article history:
Received 11 April 2011
Received in revised form 3 January 2012
Accepted 3 April 2012
Available online 14 May 2012

Keywords:
Boiler–turbine system
Model predictive control
TS fuzzy model
Genetic algorithm (GA)
Terminal cost

A B S T R A C T

This paper presents a model predictive control (MPC) strategy based on genetic algorithm to solve the boiler–turbine control problem. First, a Takagi–Sugeno (TS) fuzzy model based on gap values is established to approximate the behavior of the boiler–turbine system, then a specially designed genetic algorithm (GA) is employed to solve the resulting constrained MPC problem. A terminal cost is added into the standard performance index so that a short prediction horizon can be adopted to effectively decrease the on-line computational burden. Moreover, the GA is accelerated by improving the initial population based on the optimal control sequence obtained at the previous sampling period and a local fuzzy linear quadratic (LQ) controller. Simulation results on a boiler–turbine system illustrate that a satisfactory closed-loop performance with offset-free property can be achieved by using the proposed method.

1. Introduction

A boiler–turbine system is an energy conversion device which transforms the input chemical energy of fuel such as coal, oil, or gas, into the mechanical energy acting on the generator. The purpose of the boiler–turbine system control is to meet the load demand of electric power while maintaining the pressure and water in the drum within tolerance.

The boiler–turbine system presents a challenging control problem owing to its severe nonlinearity over a wide operation range and tight operating constraints on control and control move, which has attracted much attention and has been studied extensively in recent years [1–11].

In [1], a gain-scheduled ℓ^1-optimal approach was presented for boiler–turbine controller design based on a linear parameter varying (LPV) form of the boiler–turbine dynamics. A single linear controller was designed in [2] on the basis of careful choice of the operating range to avoid severe nonlinearity. To overcome the nonlinearity of the boiler–turbine system, many kinds of artificial intelligence techniques have also been applied. In [3], a fuzzy auto-regressive moving average (FARMA) controller was applied to the boiler–turbine system with rules generated using the history of input–output data. In [4], a feedforward fuzzy inference system and a feedback control loop were combined to attain wide-range operation. Particle swarm optimization was employed in [5,6] to realize the optimal mapping between unit load demand and pressure set-point, and genetic algorithm (GA) was used to develop a proportional-integral (PI) controller and a linear quadratic regulator (LQR) controller in [7] for a boiler–turbine system.

* Corresponding author. Address: Key Laboratory of Energy Thermal Conversion and Control of Ministry of Education, School of Energy and Environment, Southeast University, 2 Sipailou, Nanjing 210096, China. Tel.: +86 25 83795951x804; fax: +86 25 83795951x801.
E-mail address: lyg@seu.edu.cn (Y. Li)

1569-190X/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.simpat.2012.04.002

However most of these approaches except [1] cannot effectively deal with the constraints at the controller design stage. As a result, model predictive control (MPC) has been applied to control the boiler–turbine system in recent years. In [8], two types of step-response models for dynamic matrix control (DMC) were investigated in controlling a boiler–turbine system. It shows that the step-response model based on the test data is more suitable than the linearized model for controller design of the drum-type boiler–turbine system. However, because of the severe nonlinearity of the boiler–turbine system, the control performance of the linear model-based controller will degrade for a wide range of operating points. Based on the piecewise affine (PWA) model of a boiler–turbine system, an explicit MPC controller was designed offline in [9] using multi-parameter programming. But its computation burden grows exponentially with prediction horizon and the dimensions and the number of dynamics of the PWA model. Two nonlinear predictive control approaches were studied in [10], one based on neuro-fuzzy networks and another based on input–output feedback linearization technique, and showed better performance than the conventional predictive method.

In this paper, we propose a nonlinear predictive control strategy to solve the boiler–turbine control problem based on GA. First, a Takagi–Sugeno (TS) fuzzy model is established to approximate the behavior of the boiler–turbine system. Compared with the ordinary method which distributes the local linear models evenly in the whole operating range for simplification [10–12], we propose a systematic approach based on *gap values* in determining the local linear models for the TS fuzzy model of a boiler–turbine system. Then, a specially designed GA is employed to solve the resulting constrained nonlinear predictive control problem. A term for terminal cost is added into the standard performance index to further enhance the control performance, and GA is accelerated by selecting the initial population based on the optimal control sequence obtained at the previous sampling period and a local controller.

The remainder of this paper is organized as follows: Section 2 introduces the boiler–turbine dynamics. Section 3 establishes the TS fuzzy model of the boiler–turbine system based on the gap values. Offset-free MPC design using GA optimization is presented in Section 4. A linear H_∞ controller is introduced briefly in Section 5. Simulation results are given in Section 6. Finally, some conclusions are drawn in Section 7.

2. Boiler–turbine dynamics

A boiler–turbine system is an energy conversion device that consists of steam boiler and turbine. A schematic picture of a drum–boiler–turbine system is shown in Fig. 1. The aim of the steam boiler part is to transfer the input chemical energy of fuel into the thermal energy that is directly fed to the turbine part.

The boiler–turbine model used in this study is a third-order nonlinear dynamics developed by Bell and Åström in 1987 [13]. The model is based on a 160 MW oil-fired plant in Malmo, Sweden. The boiler dynamic model is provided by both physical and empirical methods based on data obtained from a series of experiments and identifications which capture all the relevant characteristics of the process.

Assume that the nonlinear dynamics of the system are in the form:

$$\begin{cases} \dot{\boldsymbol{x}} = F(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} = G(\boldsymbol{x}, \boldsymbol{u}) \end{cases} \tag{1}$$

where F and G are the state and output nonlinear equations. The dynamics of the boiler–turbine system is given by [1–5,7–11]:

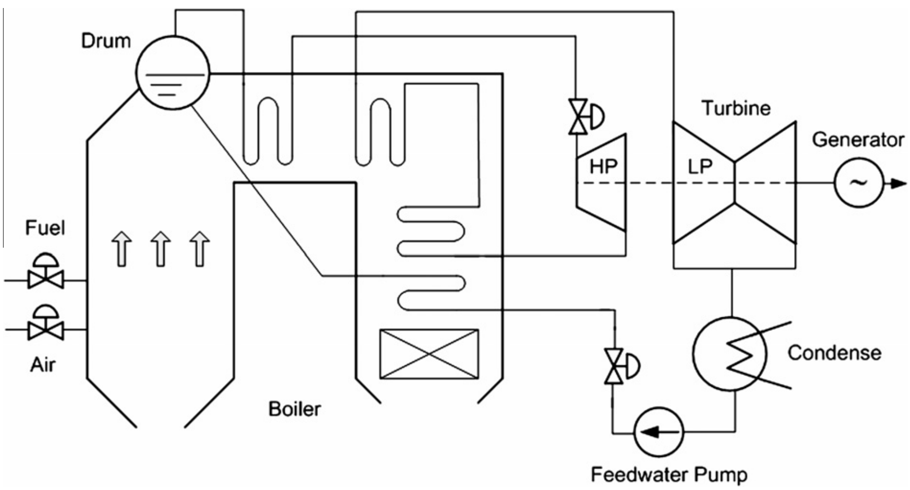

Fig. 1. Schematic diagram of a drum–boiler–turbine system.

$$\begin{cases} \dot{x}_1 = -0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3 \\ \dot{x}_2 = (0.073u_2 - 0.016)x_1^{9/8} - 0.1x_2 \\ \dot{x}_3 = (141u_3 - (1.1u_2 - 0.19)x_1)/85 \\ y_1 = x_1 \\ y_2 = x_2 \\ y_3 = 0.05(0.13073x_3 + 100a_{cs} + q_e/9 - 67.975) \end{cases} \tag{2}$$

where

$$q_e = (0.854u_2 - 0.147)x_1 + 45.59u_1 - 2.514u_3 - 2.096 \tag{3}$$

$$a_{cs} = \frac{(1 - 0.001538x_3)(0.8x_1 - 25.6)}{x_3(1.0394 - 0.0012304x_1)} \tag{4}$$

The inputs u_1, u_2 and u_3 are the valve positions for fuel flow, steam control and feedwater flow, respectively. The three state variables, x_1, x_2 and x_3, are drum steam pressure (kg/cm^2), power output (MW) and fluid density in the drum (kg/m^3), respectively. The output y_3 is the drum water level (m), q_e is the evaporation rate (kg/s) and a_{cs} is the steam quality factor.

The control inputs are subject to magnitude and rate saturations as follows. The control valve positions are normalized to the interval $[0,1]$. The other constraints model the dynamics of the control valve actuators to limit the rate of change of the valve positions.

$$\begin{cases} 0 \leqslant u_1, u_2, u_3 \leqslant 1 \\ -0.007 \leqslant \dot{u}_1 \leqslant 0.007 \\ -2 \leqslant \dot{u}_2 \leqslant 0.02 \\ -0.05 \leqslant \dot{u}_3 \leqslant 0.05 \end{cases} \tag{5}$$

Some typical operating points of the boiler–turbine model (2) are shown in Table 1 [1,2].

3. TS fuzzy modeling for the boiler–turbine system based on gap values

We consider the following discrete affine TS fuzzy model to approximate the boiler–turbine system. Due to the load dependent characteristic of the power plant, power output x_2 is chosen as the unique antecedent variable.

R_i: *if* $x_2(k)$*is* Z_i,

$$then \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{A}_i\boldsymbol{x}(k) + \boldsymbol{B}_i\boldsymbol{u}(k) + \boldsymbol{a}_i \\ \boldsymbol{y}(k) = \boldsymbol{C}_i\boldsymbol{x}(k) + \boldsymbol{D}_i\boldsymbol{u}(k) + \boldsymbol{b}_i \end{cases}, \quad i = 1, \ldots, r \tag{6}$$

where R_i denotes the ith rule of the fuzzy model, and Z_i is the ith fuzzy set of x_2. The linear model parameters are evaluated as

$$\begin{cases} \boldsymbol{A}_i = \exp(\widetilde{\boldsymbol{A}}_i \cdot T_s); \boldsymbol{B}_i = \left(\int_0^{T_s} \exp(\widetilde{\boldsymbol{A}}_i \cdot T_s)dt\right) \cdot \widetilde{\boldsymbol{B}}_i \\ \boldsymbol{C}_i = \frac{\partial G}{\partial \boldsymbol{x}}\Big|_{(\boldsymbol{x}^i, \boldsymbol{u}^i)}; \boldsymbol{D}_i = \frac{\partial G}{\partial \boldsymbol{u}}\Big|_{(\boldsymbol{x}^i, \boldsymbol{u}^i)} \\ \boldsymbol{a}_i = \left(\int_0^{T_s} \exp(\widetilde{\boldsymbol{A}}_i \cdot T_s)dt\right) \cdot \tilde{\boldsymbol{a}}_i \\ \boldsymbol{b}_i = G(\boldsymbol{x}^i, \boldsymbol{u}^i) - (\boldsymbol{C}_i\boldsymbol{x}^i + \boldsymbol{D}_i\boldsymbol{u}^i) \end{cases} \tag{7}$$

$$\widetilde{\boldsymbol{A}}_i = \frac{\partial F}{\partial \boldsymbol{x}}\Big|_{(\boldsymbol{x}^i, \boldsymbol{u}^i)}, \quad \widetilde{\boldsymbol{B}}_i = \frac{\partial F}{\partial \boldsymbol{u}}\Big|_{(\boldsymbol{x}^i, \boldsymbol{u}^i)}, \quad \tilde{\boldsymbol{a}}_i = F(\boldsymbol{x}^i, \boldsymbol{u}^i) - (\widetilde{\boldsymbol{A}}_i\boldsymbol{x}^i + \widetilde{\boldsymbol{B}}_i\boldsymbol{u}^i),$$

where T_s is the sampling time; $(\boldsymbol{x}^i,\boldsymbol{u}^i)$ is a linearization point, and when it is an equilibrium point, $F(\boldsymbol{x}^i,\boldsymbol{u}^i) = 0$.

Next step is to determine the linearization points $(\boldsymbol{x}^i,\boldsymbol{u}^i)$, and thus, to determine the number of the fuzzy rules and the state-space matrices of local linear models. Most of the time, this is an ad hoc procedure, relying on the designer's experience

Table 1
Typical operating points of the boiler–turbine system.

	#1	#2	#3	#4	#5	#6	#7
x_1	75.6	86.4	97.2	108	118.8	129.6	135.4
x_2	15.27	36.65	50.52	66.65	85.06	105.8	127
x_3	299.6	342.4	385.2	428	470.8	513.6	556.4
u_1	0.156	0.209	0.271	0.34	0.418	0.505	0.6
u_2	0.483	0.552	0.621	0.69	0.759	0.828	0.8971
u_3	0.183	0.256	0.34	0.433	0.543	0.663	0.793

and knowledge about the process. In this study, a more systematic approach using *gap values* is offered. Therefore, the gap concept between two linear systems is first introduced to facilitate our discussion in the subsequent sections.

Let $P_1 = N_1M_1^{-1}$ and $P_2 = N_2M_2^{-1}$ be normalized right coprime factorizations of the transfer functions of two linear systems. Then the gap between the two systems can be defined by [2,14,15]:

$$\delta_g(P_1, P_2) = \max\{\vec{\delta}(P_1, P_2), \vec{\delta}(P_2, P_1)\} \tag{8}$$

where $\vec{\delta}(P_1, P_2)$ is the directed gap and can be computed by

$$\vec{\delta}(P_1, P_2) = \inf_{Q_1 \in H_\infty} \left\| \begin{bmatrix} M_1 \\ N_1 \end{bmatrix} - \begin{bmatrix} M_2 \\ N_2 \end{bmatrix} Q_1 \right\|_\infty \tag{9}$$

Here, H_∞ represents the Hardy space with its norm $\|\Delta(s)\|_\infty = \max_\omega \bar{\sigma}(\Delta(j\omega)), \bar{\sigma}(\cdot)$ denotes the maximum singular value of $\Delta(j\omega)$ and Q_1 is any function that belongs to H_∞ [2].

For any two linear systems, the gap is bounded as

$$0 \leqslant \delta_g(P_1, P_2) \leqslant 1 \tag{10}$$

The gap can be regarded as the distance between two linear systems, and it is a generalization of the conventional distance expressed by the ∞-norm [2]. An important feature of the gap metric is that it is applicable not only to stable systems, but also to integrating with unstable systems.

The proposed method in determining linearization points comprises the following five steps.

Step 1: Distribute N_m linearization points $Op_i = (x_2^i, x_1^i) = (x_2^i, f(x_2^i)), i = 1, 2, \ldots, N_m$, evenly in terms of power output x_2 along the power–pressure curve $f(.)$ in the whole operating range, where we use Op_i to denote the linearization points for briefness since the linearization equilibrium point $(\boldsymbol{x}^i, \boldsymbol{u}^i)$ can be solely determined by Op_i according to (2) with the drum water level chosen to be $y_3^i = 0; N_m$ is a large number representing the initial number of linearization points. Next, linearize the boiler–turbine system around these points to obtain N_m linear models $G_i (i = 1, 2, \ldots, N_m)$.

Step 2: Prescribe a distance level ε, and calculate the gap values between adjacent linear models to obtain $N_m - 1$ gap values $\delta_i (i = 1, 2, \ldots, N_m - 1)$.

Step 3: Find the minimal gap value δ_k among δ_i, i.e.,

$$\delta_k = \delta_g(G_k, G_{k+1}) = \min_{i=1}^{N_m-1} \delta_i \tag{11}$$

where G_k and G_{k+1} are the two adjacent linear models corresponding to δ_k.

Step 4: If $\delta_k \geqslant \varepsilon$, stop the algorithm. Otherwise, merge the two linearization points corresponding to δ_k into one new linearization point $Op_{new} = \left(0.5\left(x_2^k + x_2^{k+1}\right), f\left(0.5\left(x_2^k + x_2^{k+1}\right)\right)\right)$, then linearize the boiler–turbine system around Op_{new} to obtain a new linear model G_{new}.

Step 5: Calculate the gap values between G_{new} and its two adjacent linear models G_{k-1} and G_{k+2}, and let $N_m = N_m - 1$. Rearrange the gap values and go to Step 3.

By using the previous procedures, we can finally obtain r linearization points $\widetilde{O}p_i (i = 1, 2, \ldots, r)$, which partitions the operating space as fuzzy sets with the overlapping triangle membership functions (see Fig. 3). The local state-space matrices $\{\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{C}_i, \boldsymbol{D}_i, \boldsymbol{a}_i, \boldsymbol{b}_i\}$ in the consequent part of the fuzzy model (6) are also evaluated at these points according to (7).

By using a singleton fuzzifier, the product inference, and the center-average defuzzifier, the fuzzy model (6) can be expressed as follows:

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{a} \\ \boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) + \boldsymbol{b} \end{cases} \tag{12}$$

where

$$\boldsymbol{A} = \sum_{i=1}^{r} \mu_i(x_2(k))\boldsymbol{A}_i, \quad \boldsymbol{B} = \sum_{i=1}^{r} \mu_i(x_2(k))\boldsymbol{B}_i, \quad \boldsymbol{C} = \sum_{i=1}^{r} \mu_i(x_2(k))\boldsymbol{C}_i, \quad \boldsymbol{D} = \sum_{i=1}^{r} \mu_i(x_2(k))\boldsymbol{D}_i,$$
$$\boldsymbol{a} = \sum_{i=1}^{r} \mu_i(x_2(k))\boldsymbol{a}_i, \quad \boldsymbol{b} = \sum_{i=1}^{r} \mu_i(x_2(k))\boldsymbol{b}_i,$$

$\mu_i(x_2(k)) = Z_i(x_2(k))/\sum_{i=1}^{r} Z_i(x_2(k))$, and $Z_i(.)$ is the membership function of fuzzy set Z_i.

By using the proposed method, more fuzzy rules will be distributed in the highly nonlinear operating ranges of the boiler–turbine system. As a result, it may achieve better global approximation accuracy with the same number of fuzzy rules compared with the ordinary method which distributes the local linear models evenly in the whole operating range.

4. Offset-free model predictive control based on genetic algorithm

4.1. Model predictive control problem formulation for the boiler–turbine system

Because of the use of the TS fuzzy predictive model, a nonlinear optimization problem needs to be solved to achieve the MPC. However, the conventional nonlinear programming method, e.g., sequential quadratic programming (SQP) provides local optimum values only and, in addition, these values depend on the selection of the starting point. The GA has shown better performance in solving this kind of optimization problem [16–21]. In this paper, modifications are made on the standard GA-based model predictive control via taking advantage of the well-developed theory on the stability of MPC [22–24].

The three key ingredients of the stabilizing MPC are summarized in [22], which include a terminal set, a terminal cost and a local controller. In this paper, we employ a *fictitious* fuzzy linear quadratic (LQ) controller as the local controller and a corresponding terminal cost is added to the performance index of the standard GA-based MPC.

The performance index and the constraints of the MPC for the fuzzy dynamic system (12) are defined by

$$J(k) = \sum_{j=0}^{N-1}\Big[(\hat{\boldsymbol{x}}(k+j+1|k)-\bar{\boldsymbol{x}})^T\boldsymbol{Q}(\hat{\boldsymbol{x}}(k+j+1|k)-\bar{\boldsymbol{x}})+(\boldsymbol{u}(k+j|k)-\bar{\boldsymbol{u}})^T\boldsymbol{R}(\boldsymbol{u}(k+j|k)-\bar{\boldsymbol{u}})\Big]+\Psi(\hat{\boldsymbol{x}}(k+N+1|k)-\bar{\boldsymbol{x}}) \tag{13}$$

s.t.

$$\boldsymbol{0} \leqslant \boldsymbol{u}(k+j|k) \leqslant \boldsymbol{1}, \quad j=0,\ldots,N-1 \tag{14}$$

$$\Delta\boldsymbol{u}_{\min} \leqslant \Delta\boldsymbol{u}(k+j|k) \leqslant \Delta\boldsymbol{u}_{\max}, \quad j=0,\ldots,N-1; \tag{15}$$

where N is the prediction horizon; $\boldsymbol{Q}$ and $\boldsymbol{R}$ are weighting matrices, and $\boldsymbol{Q}=\boldsymbol{Q}^T \geqslant 0, \boldsymbol{R}=\boldsymbol{R}^T > 0; \hat{\boldsymbol{x}}(k+j+1|k)$ is the predicted state at instant $k+j+1$ based on the current state $\boldsymbol{x}(k)$ and control sequence; $\boldsymbol{u}(k+j|k)$ is the control action at instant $k+j$ calculated from solving the optimization problem at instant $k; \bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$ are the equilibrium values of the state and control input vectors that correspond to the current set-points (see (22)).

The last term in (13) represents the terminal cost. This terminal cost represents the stabilizing cost that would be required when the system is to be controlled beyond the finite-time horizon N toward the infinite-time horizon. We assume that this stabilizing controller can be designed by a local LQ controller near the equilibrium point. The local controller can also be designed by other linear control techniques. In that case, a Lyapunov equation instead of (18) will need to be solved [24].

The optimal cost in driving the system to equilibrium from the time instant $k+N+1$ to the infinite time is defined by

$$\Psi(\hat{\boldsymbol{x}}(k+N+1|k)-\bar{\boldsymbol{x}}) = (\hat{\boldsymbol{x}}(k+N+1|k)-\bar{\boldsymbol{x}})^T\boldsymbol{P}(\hat{\boldsymbol{x}}(k+N+1|k)-\bar{\boldsymbol{x}}) \tag{16}$$

$$\boldsymbol{P} = \sum_{i=1}^{r}\mu_i(\hat{x}_2(k+N+1|k))\boldsymbol{P}_i \tag{17}$$

where $\boldsymbol{P}_i$ is the symmetric positive semi-definite solution of the algebraic *Riccati* equation

$$\boldsymbol{P}_i = \boldsymbol{A}_i^T\boldsymbol{P}_i\boldsymbol{A}_i - \boldsymbol{A}_i^T\boldsymbol{P}_i\boldsymbol{B}_i\left(\boldsymbol{B}_i^T\boldsymbol{P}_i\boldsymbol{B}_i+\boldsymbol{R}'\right)^{-1}\boldsymbol{B}_i^T\boldsymbol{P}_i\boldsymbol{A}_i + \boldsymbol{Q}' \tag{18}$$

where $\boldsymbol{Q}'$ and $\boldsymbol{R}'$ are the weighting matrices for the local LQ controller for the linearized model (6), and $\boldsymbol{Q}'=\boldsymbol{Q}'^T \geqslant 0$, $\boldsymbol{R}'=\boldsymbol{R}'^T > 0$.

The local controller $\boldsymbol{K}$ corresponding to the terminal cost is designed as

$$\boldsymbol{K} = \sum_{i=1}^{r}\mu_i(\hat{x}_2(k+N+1|k))\boldsymbol{K}_i \tag{19}$$

where

$$\boldsymbol{K}_i = -\left(\boldsymbol{B}_i^T\boldsymbol{P}_i\boldsymbol{B}_i+\boldsymbol{R}'\right)^{-1}\boldsymbol{B}_i^T\boldsymbol{P}_i\boldsymbol{A}_i. \tag{20}$$

Thus the local controller is represented by a fuzzy LQ controller based on membership functions of the terminal state. In formulating the optimization problem (13)–(15), which is viewed as a *global* optimization problem, we assume that there is a finite-time horizon length N, such that the prediction of the state vector, $\hat{\boldsymbol{x}}(k+N+1|k) \in \Omega$, where Ω is the terminal set, and the constraints can be assumed to be inactive for $j \geqslant N$, because it is very difficult to calculate the terminal set for a given nonlinear system, and also to reduce the computational load. As a result, the terminal set constraint is omitted in (13)–(15). Note that the assumption on the terminal set may be met if the prediction horizon is long enough or, in the case of short horizon, a large weighting matrix $\boldsymbol{R}'$ is used in designing the local controller.

The local LQ controller is only used to determine a terminal penalty matrix $\boldsymbol{P}_i$ offline and to help the GA improve the initial population, which will be illustrated in Part 4.3 of this section.

This method does not require the globally optimal input profile to be found numerically at every step. Stability only requires feasible solutions to the optimization problem. The computational (and performance) advantage of this scheme lies in the fact that shorter horizons can be used, without jeopardizing performance and stability. This is especially beneficial for on-line application of the GA-based predictive control approach.

4.2. Offset-free output tracking

To achieve offset-free output tracking, a simple disturbance estimator is run at each step, assuming that the output disturbance is given by the difference between the measured output and the expected output at time $k-1$:

$$\hat{\boldsymbol{d}}(k) = \boldsymbol{y}(k-1) - \hat{\boldsymbol{y}}(k-1|k-2) \tag{21}$$

This is then used to estimate the equilibrium values of the control input $\bar{\boldsymbol{u}}$, assuming that the disturbance will remain constant at this estimated value: $\bar{\boldsymbol{d}} = \hat{\boldsymbol{d}}(k)$. Specifically, the equilibrium values $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$ are found by setting $\boldsymbol{x}(k+1) = \boldsymbol{x}(k) = \bar{\boldsymbol{x}}, \boldsymbol{u}(k) = \bar{\boldsymbol{u}}$ in (12):

$$\begin{cases} \bar{\boldsymbol{x}} = \overline{\boldsymbol{A}}\boldsymbol{x} + \overline{\boldsymbol{B}}\bar{\boldsymbol{u}} + \bar{\boldsymbol{a}} \\ \boldsymbol{y}^r = [x_1^r \quad x_2^r \quad 0]^T = \overline{\boldsymbol{C}}\bar{\boldsymbol{x}} + \overline{\boldsymbol{D}}\bar{u} + \bar{\boldsymbol{b}} + \hat{\boldsymbol{d}}(k) \end{cases} \tag{22}$$

where x_1^r and x_2^r denote the set-points for the drum pressure and the power output, respectively, and the set-point for drum water lever is always set to be 0. The matrices $(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \overline{\boldsymbol{C}}, \overline{\boldsymbol{D}}, \bar{\boldsymbol{a}}, \bar{\boldsymbol{b}})$ in (22) are determined by the membership function of x_2^r as in (12).

Note that this solution results in offset-free control in the presence of an unknown but constant disturbance, even if the steady-state gains in the model are not accurate.

4.3. Genetic optimization of control inputs

A dynamic nonlinear optimization problem has been formulated based on the TS fuzzy model, where conventional optimization techniques cannot be easily applied. Therefore, in this work, the online optimization problem is solved using a GA.

The algorithm starts with an initial population of chromosomes, which represent possible solutions of the optimization problem, i.e., control inputs. For each chromosome the objective function is computed. New generations are produced by the genetic operators such as selection, crossover and mutation. The algorithm stops after the maximum allowed time has passed. To deal with constraints in this optimization problem, the penalty function method is commonly used. However, this approach lowers the efficiency of a GA, because the genetic material is wasted due to the unfeasible solutions in the standard genetic operation. In this paper, specially designed genetic operators are employed to make the newly generated chromosomes satisfy the constraints automatically.

A chromosome which is a candidate solution of the optimization problem is represented by s^l, whose elements consist of present and future control inputs and has the following structure [17]:

$$s^l = \{ \boldsymbol{u}^l(k) \quad \boldsymbol{u}^l(k+1) \quad \cdots \quad \boldsymbol{u}^l(k+N-1) \}, \quad l = 1, 2, \ldots, L \tag{23}$$

where k indicates the current time, and L is the number of chromosomes. The algorithm can be described as follows:

Step 1 (initial population generation): Set the number of iterations $i = 1$. Predictive control uses the receding horizon principle, which implies that an optimization has to be performed at each time step. Therefore, the past solutions give important information, which can be used to improve the initial population of the current solutions.
Assume the optimal input sequence obtained at k-1 is $\boldsymbol{U}^*(k-1) = \{ \boldsymbol{u}^*(k-1), \quad \boldsymbol{u}^*(k) \quad , \ldots, \quad \boldsymbol{u}^*(k+N-2) \}$. At the current time k, consider a "shifted" input sequence $\widetilde{\boldsymbol{U}}(k)$ as shown below, where the last gene takes $\tilde{\boldsymbol{u}}(k+N-1)$, to be one of the initial chromosomes. This chromosome might be a very good guess for the solution of the next optimization problem. The optimal input sequence found at k-1 and an initial chromosome at time k obtained by shifting the input sequence forward by one are given below:

$$\boldsymbol{U}^*(k-1) = \left\{ \underbrace{\boldsymbol{u}^*(k-1)}_{applied}, \quad \boldsymbol{u}^*(k), \quad \ldots, \quad \underbrace{\boldsymbol{u}^*(k+N-3)}, \quad \underbrace{\boldsymbol{u}^*(k+N-2)} \right\}$$

$$\swarrow \qquad \swarrow \qquad \cdots \qquad \swarrow$$

$$s^1 = \tilde{\boldsymbol{U}}(k) = \left\{ \boldsymbol{u}^*(k), \quad \boldsymbol{u}^*(k+1), \quad \ldots, \quad \boldsymbol{u}^*(k+N-2), \quad \underbrace{\tilde{\boldsymbol{u}}(k+N-1)}_{new\ tail} \right\}$$

where the newly added tail is defined by

$$\tilde{u}_i(k+N-1) = \begin{cases} low_i, & \text{if } \tilde{u}'_i < low_i \\ \tilde{u}'_i, & \text{if } low_i \leqslant \tilde{u}'_i \leqslant upper_i \ (i=1,2,3) \\ upper_i, & \text{if } \tilde{u}'_i > upper_i \end{cases} \tag{24}$$

$$\tilde{\boldsymbol{u}}' = \boldsymbol{K}(\hat{\boldsymbol{x}}(k+N-1) - \bar{\boldsymbol{x}}) + \bar{\boldsymbol{u}}; \tag{25}$$

$$low_i = \max\{0, u_i^*(k+N-2) + \Delta u_{i\min}\}; \tag{26}$$

$$upper_i = \min\{1, u_i^*(k+N-2) + \Delta u_{i\max}\}. \tag{27}$$

where $\boldsymbol{K}$ is the optimal gain (19) computed for the local controller.

To satisfy both the control and control move constraints, we use a simple procedure to generate the remaining $L-1$ chromosomes, $s^2, \ldots, s^L$, of the initial population:

$$u_i^l(k+j) = \min(1, \max(0, u_r)) \tag{28}$$

$(1 \leqslant i \leqslant 3,\ 0 \leqslant j \leqslant N-1, 2 \leqslant l \leqslant L)$. where

$$u_r = \begin{cases} rand[u_i(k-1) + \Delta u_{i\min}, u_i(k-1) + \Delta u_{i\max}] \\ (j=0) \\ rand\left[u_i^l(k+j-1) + \Delta u_{i\min}, u_i^l(k+j-1) + \Delta u_{i\max}\right] \\ (1 \leqslant j \leqslant N-1) \end{cases} \tag{29}$$

In the above equations, u_r is a random number. A new random number u_r is generated each time (28) is used.

Step 2 (fitness function evaluation): Evaluate the objective function of (13) for all of the chosen chromosomes. Then calculate their fitness value according to

$$fitness(l) = 1/(1+J_l), \quad l = 1, 2, \ldots, L \tag{30}$$

where J_l is the value of the objective function for the lth chromosome.

Then, calculate the normalized fitness value of each chromosome, which implies the *selection probability*, calculated by

$$p_l = fitness(l) \Bigg/ \sum_{l=1}^{L} fitness(l) \tag{31}$$

Step 3 (selection operation): Preserve m_2 best individuals, and reintroduce them into the population for the next generation. Therefore, the partly optimized chromosomes will not get lost in spite of the disruption of building blocks during crossover.

Generate the rest $L - m_2$ chromosomes according to their selection probabilities. The chromosomes with high fitness values will have more chances to be selected.

Step 4 (crossover operation): For each chromosome, generate a random number r_1 between 0 and 1. If r_1 is lower than the probability of crossover p_c, this particular chromosome will undergo the process of crossover. Mate the selected chromosomes, and for each selected pair one of the following two crossover operations is implemented with equal probability.

(1) *The one-point crossover operation*

Generate a random integer z between 1 and $N-1$, which indicate the position of the crossover point. Two new chromosomes are produced by interchanging all of the members of the parents following the crossover point, which can be expressed graphically as follows:

$$s^l = \{\boldsymbol{u}^l(k), \quad \ldots, \quad \boldsymbol{u}^l(k+z-1), | \quad \boldsymbol{u}^l(k+z), \quad \ldots, \quad \boldsymbol{u}^l(k+N-1)\}$$

$$s^{l+1} = \{\boldsymbol{u}^{l+1}(k), \quad \ldots, \quad \boldsymbol{u}^{l+1}(k+z-1), | \quad \boldsymbol{u}^{l+1}(k+z), \quad \ldots, \quad \boldsymbol{u}^{l+1}(k+N-1)\}$$

$$\Downarrow \text{ the one-point crossover}$$

$$s^l_{new} = \{\boldsymbol{u}^l(k), \quad \ldots, \quad \boldsymbol{u}^l(k+z-1), | \quad \boldsymbol{u}^{l+1}(k+z), \quad \ldots, \quad \boldsymbol{u}^{l+1}(k+N-1)\}$$

$$s^{l+1}_{new} = \{\boldsymbol{u}^{l+1}(k), \quad \ldots, \quad \boldsymbol{u}^{l+1}(k+z-1), | \quad \boldsymbol{u}^l(k+z), \quad \ldots, \quad \boldsymbol{u}^l(k+N-1)\}$$

The previous operation might produce infeasible offspring if the input values at the crossover point do not satisfy the control move constraints. This situation is avoided by the following correction mechanism for each of the new chromosomes s^l_{new} and s^{l+1}_{new}.

For s^{l+1}_{new}, suppose $d_i = u_i^l(k+z) - u_i^{l+1}(k+z-1)$, then

$$u_i^l(k+z+j) = \begin{cases} u_i^l(k+z+j) - (d_i - \Delta u_{i\max}), & \text{if } d_i > \Delta u_{i\max} \\ u_i^l(k+z+j) - (d_i - \Delta u_{i\min}), & \text{if } d_i < \Delta u_{i\min} \end{cases} \tag{32}$$

$(1 \leqslant i \leqslant 3,\ 0 \leqslant j \leqslant N - z - 1)$
Similar equations can be obtained for the chromosome s_{new}^l.

(2) *The uniform crossover operation*

For the uniform crossover operation, two new chromosomes based on s^l and s^{l+1} are produced by

$$\begin{cases} s_{new}^l = r_2 \cdot s^l + (1 - r_2) \cdot s^{l+1} \\ s_{new}^{l+1} = (1 - r_2) \cdot s^l + r_2 \cdot s^{l+1} \end{cases} \tag{33}$$

where r_2 is a random number between 0 and 1.

Step 5 (mutation operation): For every member of each chromosome, generate a random number r_3 between 0 and 1. If r_3 is lower than the probability of mutation p_m, this particular member of the chromosomes will undergo the process of mutation, otherwise it will remain unchanged. For the selected members, lower and upper bounds $[b_l(j), b_u(j)]$ are defined as:

$$b_u(j) = \begin{cases} \min(\Delta u_{i\max} + u_i(k-1), \Delta u_{i\max} + u_i(k+1), 1), & j = 0 \\ \min(\Delta u_{i\max} + u_i(k+j-1), \Delta u_{i\max} + u_i(k+j+1), 1), & 0 < j < N-1 \\ \min(\Delta u_{i\max} + u_i(k+j-1), 1), & j = N-1 \end{cases} \tag{34}$$

$$b_l(j) = \begin{cases} \max(\Delta u_{i\min} + u_i(k-1), \Delta u_{i\min} + u_i(k+1), 0), & j = 0 \\ \max(\Delta u_{i\min} + u_i(k+j-1), \Delta u_{i\min} + u_i(k+j+1), 0), & 0 < j < N-1 \\ \max(\Delta u_{i\min} + u_i(k+j-1), 0), & j = N-1 \end{cases} \tag{35}$$

The above bounds define the region of values of which will produce a feasible solution. The mutation operation is then achieved by the generation of a random number within $[b_l(j), b_u(j)]$.

$$u_i^{new}(k+j) = r_m(j), \ \text{if } r_3 < p_m \tag{36}$$

where $r_m(j)$ is a random number within $[\,b_l(j), b_u(j)]$.

Step 6 (repeat or stop): If the maximum allowed time has not expired, set and return the algorithm to Step 2. Otherwise, stop the algorithm and select the chromosome that produced the highest value of the fitness function throughout the entire procedure.

The previous modified GA makes it possible to calculate the suboptimal control in real time.

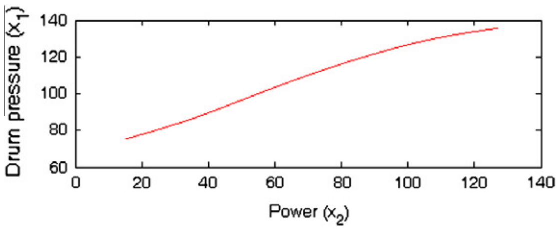

Fig. 2. Power–pressure curve in sliding operation mode.

Table 2
Process of determination of linearization points using gap values.

Time	Power output (x_2^i) at the linearization points									Gap values between the adjacent linearized models							
1	15.27	29.2	43.2	57	71	85	99	113	127	0.051	0.0565	0.056	0.0487	0.0586	0.086	0.147	0.6358
2	15.27	29.2	43.2	64		85	99	113	127	0.051	0.0565	0.081	0.0812		0.086	0.147	0.6358
3	22.2		43.2	64		85	99	113	127	0.083		0.081	0.0812		0.086	0.147	0.6358
4	22.2		53.6			85	99	113	127	0.1257		0.1105			0.086	0.147	0.6358
5	22.2		53.6			92		113	127	0.1257		0.1469		0.1934			0.6358

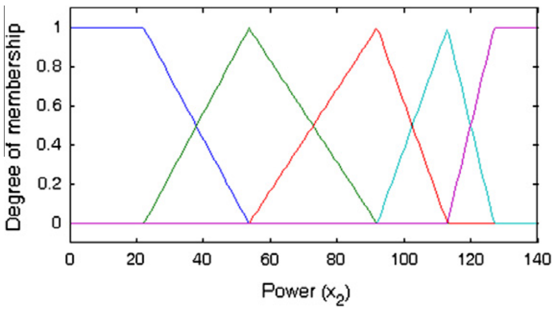

Fig. 3. Membership function of the power level x_2.

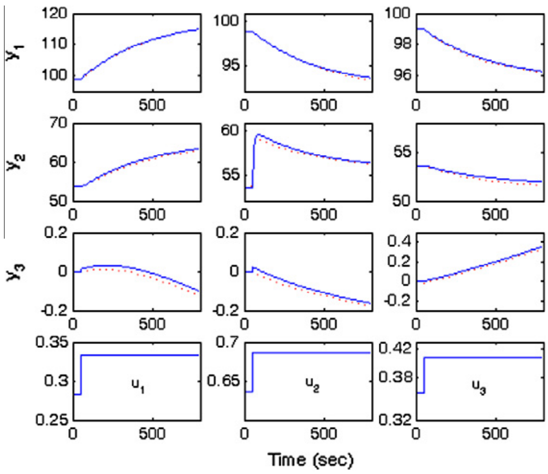

Fig. 4. Comparison of step responses of the TS fuzzy model (dotted lines) and the original nonlinear model of the boiler–turbine system (solid lines) at low load.

5. H$_\infty$ controller

In [2], a linear controller was designed for this system based on a loop-shaping H$_\infty$ approach using a linearized model at the nominal operating point. A precompensator and postcompensator were designed with a constant decoupler, aligning the singular values of the model at 0.001 rad/s.

To compensate for the effects of the constraints, this approach employed the anti-windup bumpless transfer (AWBT) technique after the controller design. But since the implementation of the AWBT technique is not easy for a state space controller, the H$_\infty$ controller was simplified by using a PID reduction method. Finally, it was reduced to four SISO PI controllers as follows:

$$K(s) = \begin{bmatrix} 0.0485 + 0.0012/s & 0 & 1.2091 + 0.0486/s \\ 0 & 0.0197 + 0.0045/s & 0 \\ 0 & 0 & 7.2548 + 0.2914/s \end{bmatrix} \tag{37}$$

This controller was then compensated using the AWBT technique.

6. Simulation results

6.1. TS fuzzy model of the boiler–turbine system

From the typical operating points given in Table 1, we fitted a power–pressure curve $x_{10} = f(x_{20})$ of the boiler–turbine system in sliding operation mode as shown in Fig. 2. The linearized model of the nonlinear model (2)–(4) about an equilibrium point ($\boldsymbol{x}_0$, $\boldsymbol{u}_0$) was obtained using Taylor series expansion with

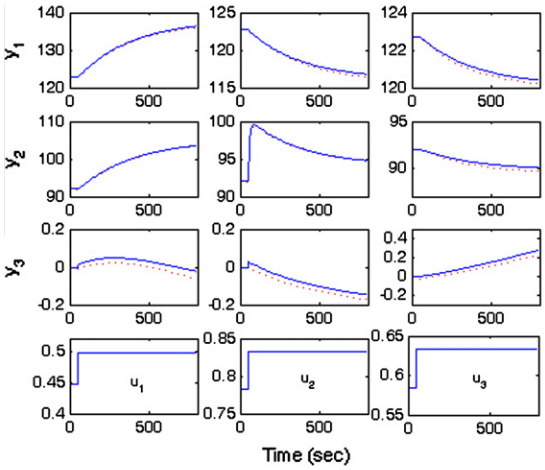

Fig. 5. Comparison of step responses of the TS fuzzy model (dotted lines) and the original nonlinear model of the boiler–turbine system (solid lines) at medium load.

Table 3
Mean square errors of the TS fuzzy model.

Step inputs	x_{20} = 53.6 MW (N_t = 80)			x_{20} = 92 MW (N_t = 80)			x_{20} = 120 MW (N_t = 80)		
	MSE (y_1)	MSE (y_2)	MSE (y_3)	MSE (y_1)	MSE (y_2)	MSE (y_3)	MSE (y_1)	MSE (y_2)	MSE (y_3)
u_1	0.0201	0.0651	0.0004	0.0500	0.0128	0.0009	0.1644	0.4891	0.0002
u_2	0.0930	0.0089	0.0002	0.0700	0.0045	0.0007	0.1729	0.3450	0.00002
u_3	0.2079	0.2706	0.0001	0.0281	0.0869	0.0014	0.0562	0.0198	0.0024

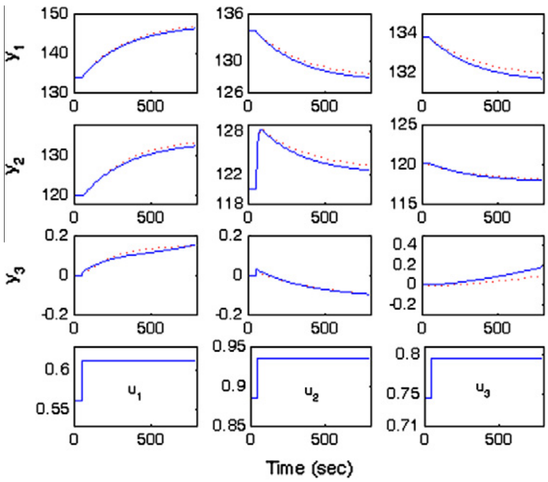

Fig. 6. Comparison of step responses of the TS fuzzy model (dotted lines) and the original nonlinear model of the boiler–turbine system (solid lines) at high load.

$$\boldsymbol{A}_0 = \begin{bmatrix} -\frac{0.0162}{8}u_{20}x_{10}^{1/8} & 0 & 0 \\ \left(\frac{6.57}{80}u_{20} - \frac{1.44}{80}\right)x_{10}^{1/8} & -0.1 & 0 \\ \left(\frac{0.19}{85} - \frac{1.1}{85}u_{20}\right) & 0 & 0 \end{bmatrix} \tag{38}$$

$$\boldsymbol{B}_0 = \begin{bmatrix} 0.9 & -0.0018x_{10}^{9/8} & -0.15 \\ 0 & 0.073x_{10}^{9/8} & 0 \\ 0 & -\frac{1.1}{85}x_{10} & \frac{141}{85} \end{bmatrix} \tag{39}$$

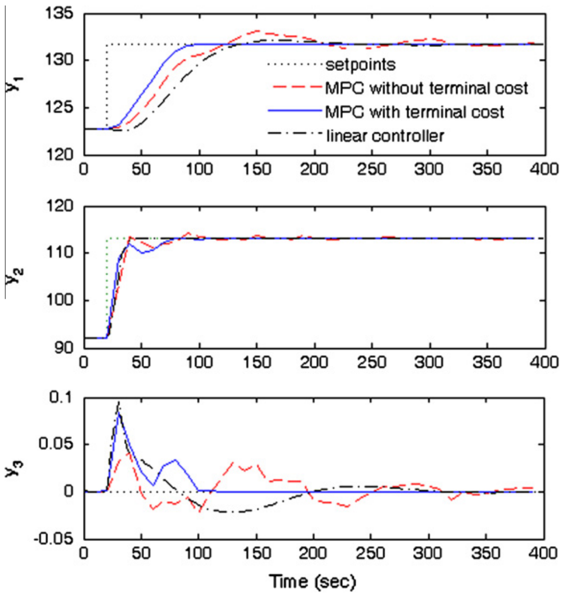

Fig. 7. Comparison of the closed-loop responses obtained with the linear controller, and with the model predictive controller with and without a terminal cost.

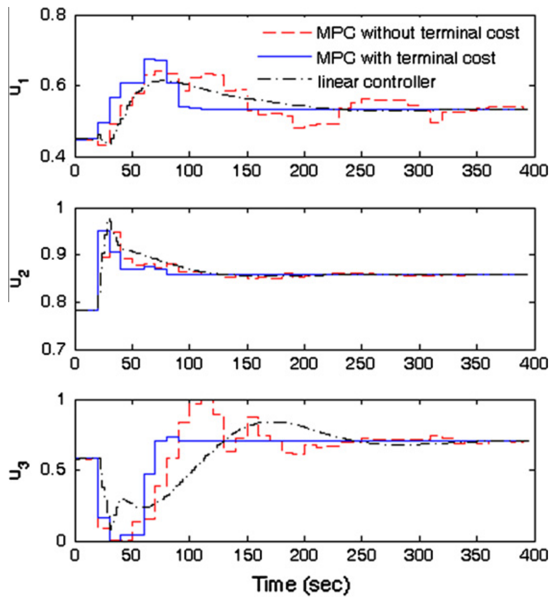

Fig. 8. Control inputs of the system obtained with the linear controller, and with the model predictive controller with and without a terminal cost.

$$\boldsymbol{C}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_{31} & 0 & c_{33} \end{bmatrix} \tag{40}$$

where

$$c_{31} = \frac{4(1 - 0.001583x_{30})}{x_{30}(1.0394 - 0.0012304x_{10})^2} + 0.0047u_{20} - 0.00082$$

$$c_{33} = 0.0065 - \frac{4x_{10} - 128}{x_{30}^2(1.0394 - 0.0012304x_{10})}$$

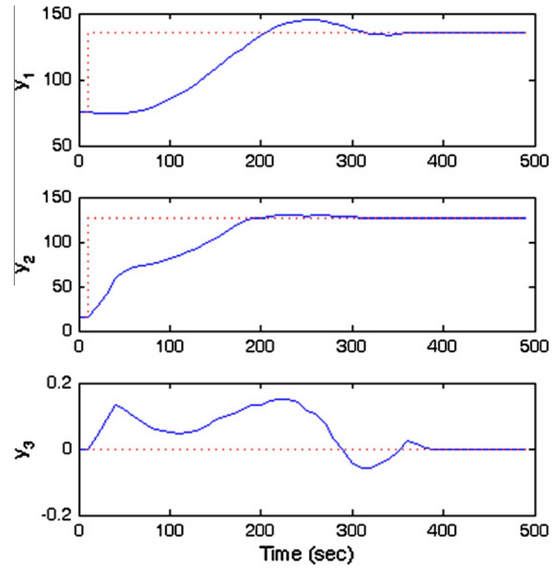

Fig. 9. Output response of the system obtained with the model predictive controller with a terminal cost when the operating point was changed from #1 to #7.

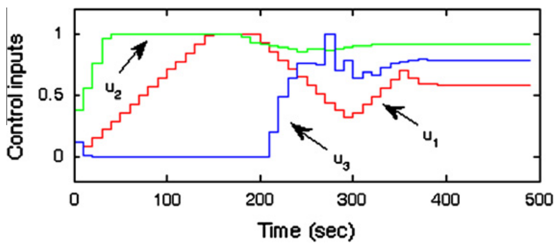

Fig. 10. Control inputs of the system obtained with the model predictive controller with a terminal cost when the operating point was changed from #1 to #7.

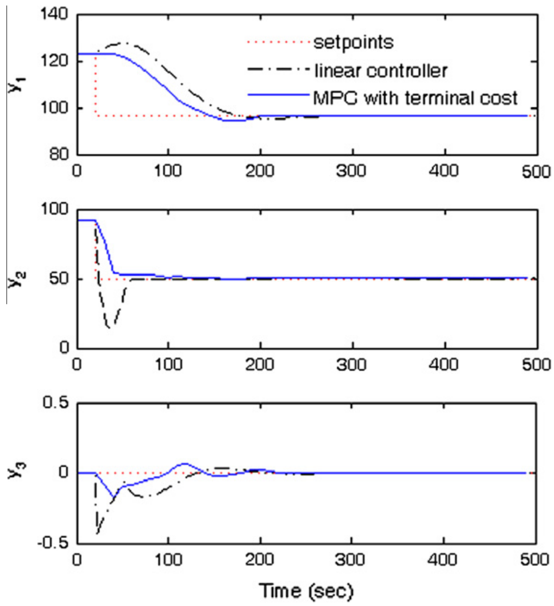

Fig. 11. Output responses of the system during load rejection.

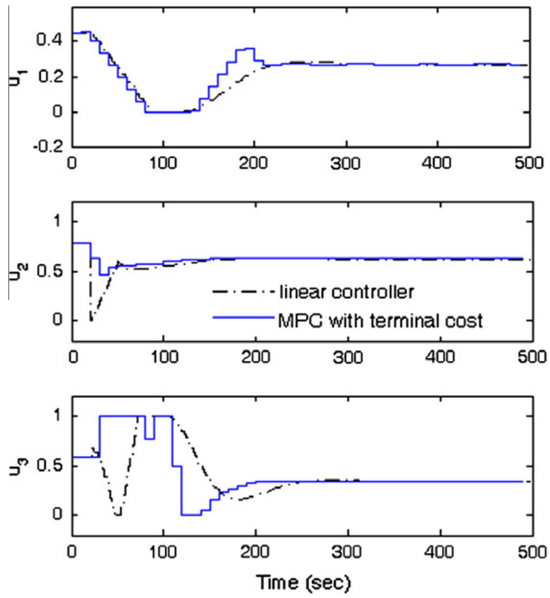

Fig. 12. Control inputs of the system during load rejection.

$$\boldsymbol{D}_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.2533 & 0.00474x_{10} & -0.014 \end{bmatrix} \tag{41}$$

In terms of Table 1, we initially distributed nine linearization points evenly along the power–pressure curve within the power range, $x_2 \in [15.27, 127]$. Then, by applying the gap based method depicted in Section 3 step by step with $\varepsilon = 0.1$, we ended up with five linearization points after going through four mergers. This process is displayed in Table 2, where the final five linearization points are [22.2, 53.6, 92, 113, 127].

Based on the five linearization points and after choosing the sampling time as $T_s = 10$ s, we obtained the discrete TS fuzzy model (6) with five rules for the boiler–turbine system with matrices $\{\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{C}_i, \boldsymbol{D}_i, \boldsymbol{a}_i, \boldsymbol{b}_i, i = 1, 2, \ldots, 5\}$, which are given in Appendix A. The membership functions of the power level x_2 are shown in Fig. 3.

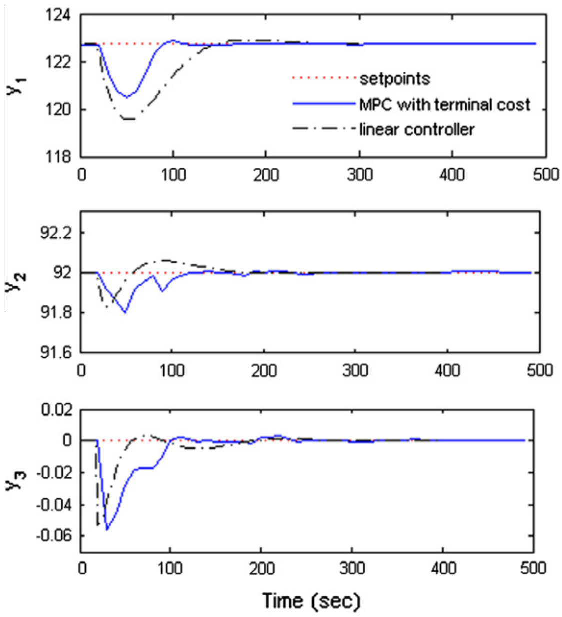

Fig. 13. Output responses of the system under an input step disturbance.

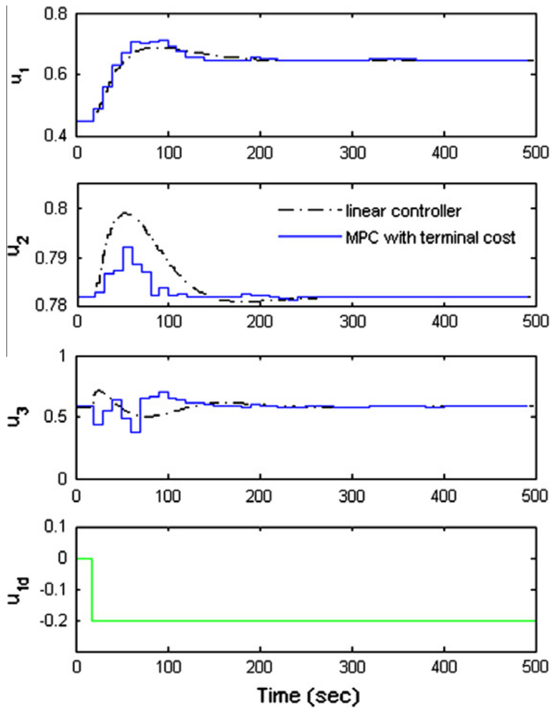

Fig. 14. Control inputs of the system under an input step disturbance.

Figs. 4–6 show a comparison of the step responses obtained with the TS fuzzy model (dotted lines) and the original nonlinear model (solid lines), for three different initial operating points $\boldsymbol{x}_{01} = (98.9647, 53.6, 446.8871)^T$, $\boldsymbol{u}_{01} = (0.2836, 0.6369, 0.3584)^T$, $\boldsymbol{y}_{01} = (98.9647, 53.6, 0)^T$; $\boldsymbol{x}_{02} = (122.74, 92, 385.18)^T$, $\boldsymbol{u}_{02} = (0.4474, 0.7820, 0.5834)^T$, $\boldsymbol{y}_{02} = (122.74, 92, 0)^T$; and $\boldsymbol{x}_{03} = (133.7807, 120, 317.8048)^T$, $\boldsymbol{u}_{03} = (0.5609, 0.8855, 0.7439)^T$, $\boldsymbol{y}_{03} = (133.7807, 120, 0)^T$. In the results shown in these figures, three inputs with a step size of 0.05 were applied independently. The mean square errors (MSEs) of the outputs are shown in Table 3, where N_t is the number of test data points.

It can be observed that the outputs y_1, y_2 and y_3 of the TS fuzzy model describe the original nonlinear dynamics effectively.

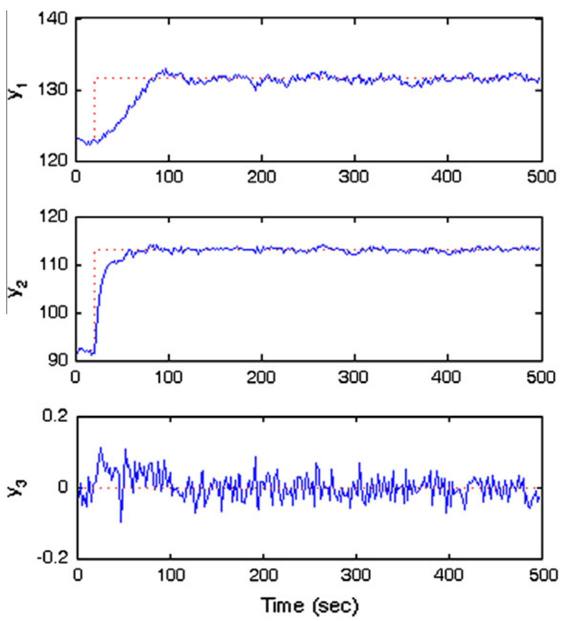

Fig. 15. Output responses of the system obtained with the proposed MPC controller under an output noise disturbance.

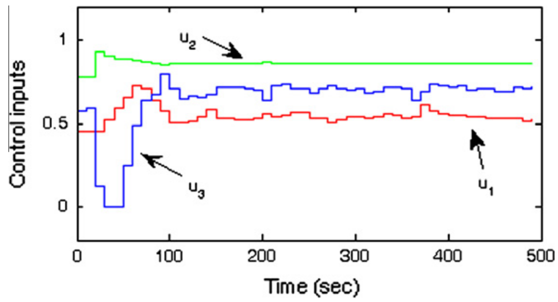

Fig. 16. Control inputs of the system obtained with the proposed MPC controller under an output noise disturbance.

6.2. Predictive control of the boiler–turbine system

In this section, we describe the application of the proposed model predictive controller to the boiler–turbine system.

In addition to using the same parameters N, $\boldsymbol{Q}$, $\boldsymbol{R}$, etc. as in regular MPC, the weighting matrices $\boldsymbol{Q}'$ and $\boldsymbol{R}'$ need to be chosen properly in this method, because they affect the weighting matrix $\boldsymbol{P}$ in the terminal cost (16), and therefore affect the control performance. During the simulations, we first fixed $\boldsymbol{Q}'$ by setting $\boldsymbol{Q}' = \boldsymbol{Q}$, and then simply changed $\boldsymbol{R}'$. It was found that increasing $\boldsymbol{R}'$ would also increase the penalty on the terminal states, but that satisfactory control performance could still be achieved, even though $\boldsymbol{R}'$ varied over a wide range. However, too large a value of $\boldsymbol{R}'$ had a bad influence on the control performance, whereas too small a value of $\boldsymbol{R}'$ reduced the effects of the terminal cost.

By trial and error, the weighting matrices and control parameters used in all of the simulations described here were chosen as $\boldsymbol{Q} = \boldsymbol{Q}' = \mathrm{diag}(120, 20, 5)$, $\boldsymbol{R} = \mathrm{diag}(100, 80, 80)$, $\boldsymbol{R}' = \mathrm{diag}(10000, 8000, 8000)$, $p_c = 0.8$, $p_m = 0.1$, $L = 20$ and $m_2 = 2$.

Using the system matrices $\{\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{C}_i, \boldsymbol{D}_i, \boldsymbol{a}_i, \boldsymbol{b}_i, i = 1, 2, \ldots, 5\}$, the positive semi-definite matrix solution $\boldsymbol{P}_i$ and the local controller $\boldsymbol{K}_i$ were first calculated by means of (18) and (20).

The linear H_∞ controller introduced in Section 5 was also used for comparison. The sampling time of the linear controller was set to 2 s.

We considered the following cases.

In the first case, we assumed that the system was initially in the same steady state as in Fig. 5, and at $t = 20$ s, the setpoints for the pressure and power output were increased to 131.66 and 113, respectively, while the drum water level was kept at zero. Fig. 7 shows a comparison of the closed-loop responses of the boiler–turbine system obtained with the linear controller (dash–dotted line), the model predictive controller with a terminal cost (solid line) and the model predictive controller without a terminal cost (dashed line), where a short horizon $N = 2$ was adopted. Fig. 8 shows the corresponding control actions. These results show that good control performance can be achieved by considering a terminal cost, even when a short horizon is adopted. The main advantage of adopting a short horizon is that the online computational burden is effectively decreased,

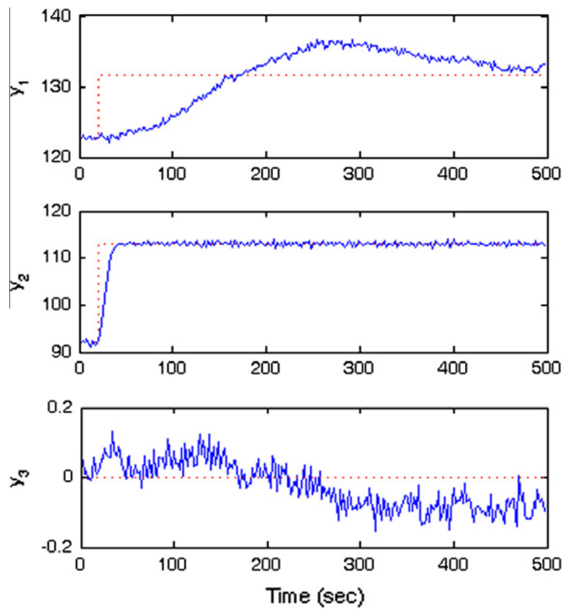

Fig. 17. Output responses of the system obtained with the linear controller under an output noise disturbance.

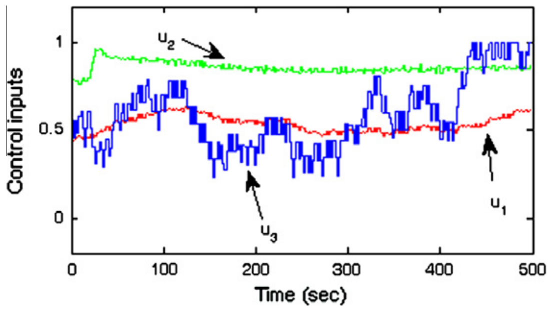

Fig. 18. Control inputs of the system obtained with the linear controller under an output noise disturbance.

since both the number of decision variables and the number of prediction steps are reduced. In addition, although the linear controller can also successfully control the system at this operating point near its nominal model, the model predictive controller with a terminal cost achieves better performance.

To further test the performance of the proposed model predictive controller with a terminal cost, another two simulations were done on the boiler–turbine system under step-like load changes. In the first simulation, the system was driven from operating point #1 (see Table 1) to the distant operating point #7 at $t = 10$ s while the drum water level was maintained at zero. The proposed MPC controller showed good tracking behavior, as shown in Figs. 9 and 10. However, the linear controller resulted in instability of the system; the results have been omitted here for brevity. The second simulation was done to imitate the condition of load rejection. With the same initial operating point as in Fig. 5, we assumed that the setpoint of the power output was suddenly decreased from 92 to 50 MW. The responses and the control inputs of the proposed MPC controller and the linear controller are shown in Figs. 11 and 12. It is apparent that the MPC controller has better control performance and less overshoot.

Next, an unmeasured input disturbance and also output noise disturbances were considered in order to evaluate the disturbance rejection ability of the control system. We first assumed that at $t = 18$ s, there was an additive unmeasured disturbance $u_{1d} = -0.2$ acting on the valve position u_1 controlling the fuel flow. The outputs of the boiler–turbine system obtained

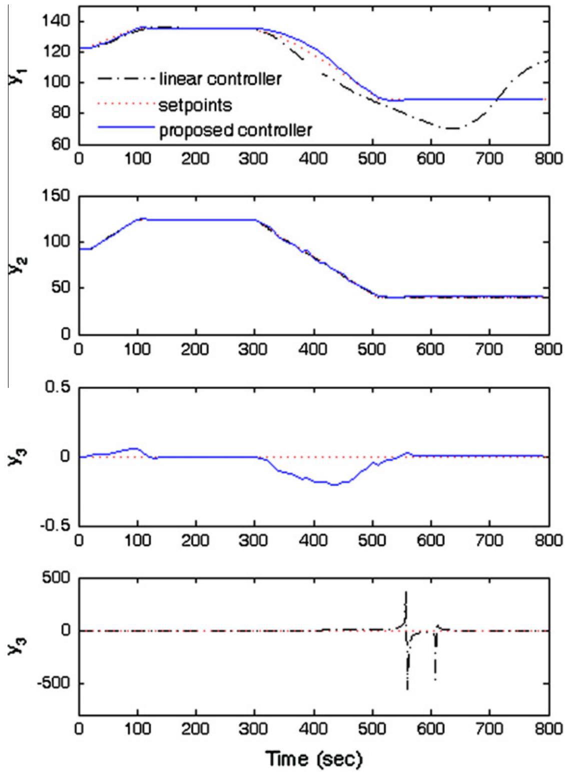

Fig. 19. Output responses of the system under wide-range operation.

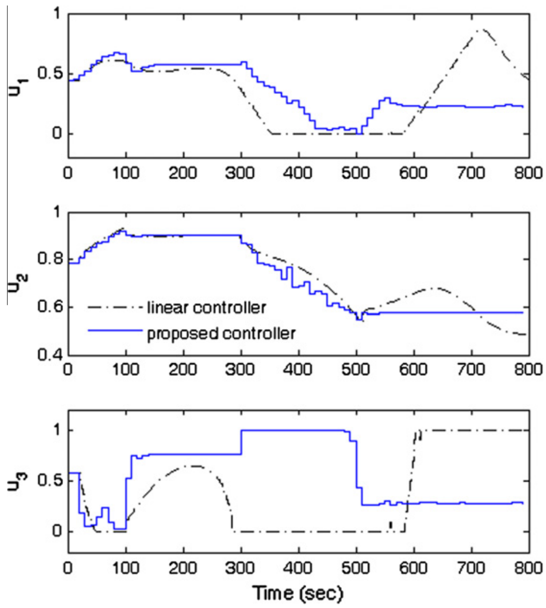

Fig. 20. Control inputs of the system under wide-range operation.

with both the MPC controller with a terminal cost and the linear controller are shown in Fig. 13, and the inputs are shown in Fig. 14. It is evident from Fig. 13 that the proposed MPC controller handles the sudden decrease in fuel flow more effectively than does the linear controller. Second, we assumed that the outputs y_1, y_2, and y_3 were affected by white Gaussian noise with the power equal to 0.12, 0.1, and 0.001 Watt, respectively. The responses and the control inputs of the two controllers are depicted in Figs. 15–18. By comparing these figures with the results for the noise-free cases shown in Figs. 7 and 8, one can conclude that the proposed MPC method is more robust to output noise than the linear controller is.

The final case was designed to demonstrate the tracking ability of the proposed method during wide-range operation. As shown in Fig. 19, after $t = 20$ s the setpoint for the load demand (the dotted line in the plot of y_2) changed linearly between 40 and 124 MW at a rate of 0.4 MW/s (15% maximum continuous rating/min), and the drum pressure setpoint (the dotted line in the plot of y_1) changed in proportion between 89.46 and 134.9 kg/cm^2. Figs. 19 and 20 show the simulation results. It can be observed that good tracking of the unit load demand and drum pressure was obtained, with a smooth transition between the operating windows, while the water level deviation (y_3) was regulated. The control results obtained using the linear controller are also given (dash–dotted lines). It can be observed that the linear controller can guarantee satisfactory control performance only at operating points near its nominal model, and wide-range operation may result in instability of the boiler–turbine system.

7. Conclusion

In this paper, we have proposed a model predictive control strategy to solve the boiler–turbine control problem, in which a TS fuzzy model was first established for local models based on gap values to approximate the behavior of the boiler–turbine system, then a specially designed GA was employed to solve the resulting nonlinear constrained predictive control problem. Meanwhile, a modification was made on the standard performance index by adding a terminal cost, which can further enhance the control performance. Moreover, the GA was accelerated by improving the initial population based on the optimal control sequence obtained at the previous sampling period and a local fuzzy LQ controller. In addition, the measure to achieve offset-free property was also discussed. Simulation results on the boiler–turbine system illustrate that the proposed method can successfully handle the control and control move constraints, and that a satisfactory closed-loop performance can be achieved.

Acknowledgments

The authors would like to thank anonymous reviewers and the editor for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (Nos. 51076027 and 51036002).

Appendix A. Local linear models for TS fuzzy model

$$\boldsymbol{A}_1 = \begin{bmatrix} 0.9851 & 0 & 0 \\ 0.1986 & 0.3679 & 0 \\ -0.0347 & 0 & 1 \end{bmatrix}; \quad \boldsymbol{B}_1 = \begin{bmatrix} 8.9328 & -2.4337 & -1.4888 \\ 1.0439 & 62.5699 & -0.1740 \\ -0.1567 & -10.1683 & 16.6141 \end{bmatrix}; \quad \boldsymbol{a}_1 = [\,1.2109 \quad -29.4221 \quad 4.4968\,]^T;$$

$$\boldsymbol{A}_2 = \begin{bmatrix} 0.9773 & 0 & 0 \\ 0.3977 & 0.3679 & 0 \\ -0.0593 & 0 & 1 \end{bmatrix}; \quad \boldsymbol{B}_2 = \begin{bmatrix} 8.8973 & -3.1289 & -1.4829 \\ 1.9998 & 80.4326 & -0.3333 \\ -0.2679 & -12.7178 & 16.6327 \end{bmatrix}; \quad \boldsymbol{a}_2 = [\,2.2411 \quad -55.3966 \quad 8.0905\,]^T;$$

$$\boldsymbol{A}_3 = \begin{bmatrix} 0.9714 & 0 & 0 \\ 0.5246 & 0.3679 & 0 \\ -0.0779 & 0 & 1 \end{bmatrix}; \quad \boldsymbol{B}_3 = \begin{bmatrix} 8.8708 & -3.9711 & -1.4785 \\ 2.7655 & 102.0555 & -0.4609 \\ -0.3521 & -15.7214 & 16.6467 \end{bmatrix}; \quad \boldsymbol{a}_3 = [\,3.4951 \quad -86.9743 \quad 12.2813\,]^T;$$

$$\boldsymbol{A}_4 = \begin{bmatrix} 0.9685 & 0 & 0 \\ 0.5988 & 0.3679 & 0 \\ -0.0876 & 0 & 1 \end{bmatrix}; \quad \boldsymbol{B}_4 = \begin{bmatrix} 8.8575 & -4.2939 & -1.4763 \\ 3.1586 & 110.3244 & -0.5264 \\ -0.3963 & -16.8519 & 16.6540 \end{bmatrix}; \quad \boldsymbol{a}_4 = [\,4.1434 \quad -103.3970 \quad 14.4336\,]^T;$$

$$\boldsymbol{A}_5 = \begin{bmatrix} 0.9666 & 0 & 0 \\ 0.6539 & 0.3679 & 0 \\ -0.0944 & 0 & 1 \end{bmatrix}; \quad \boldsymbol{B}_5 = \begin{bmatrix} 8.8487 & -4.4253 & -1.4748 \\ 3.4507 & 113.6710 & -0.5751 \\ -0.4271 & -17.3084 & 16.6592 \end{bmatrix}; \quad \boldsymbol{a}_5 = [\,4.5551 \quad -113.8303 \quad 15.8101\,]^T;$$

$$\boldsymbol{b}_1 = [0 \;\; 0 \;\; -3.2278]^T; \boldsymbol{b}_2 = [0 \;\; 0 \;\; -3.1609]^T; \boldsymbol{b}_3 = [0 \;\; 0 \;\; -3.0124]^T; \boldsymbol{b}_4 = [0 \;\; 0 \;\; -2.9110]^T; \boldsymbol{b}_5 = [0 \;\; 0 \;\; -2.7787]^T;$$

$$\boldsymbol{C}_1(3,1) = 0.0036; \boldsymbol{C}_1(3,3) = 0.0057; \quad \boldsymbol{C}_2(3,1) = 0.0055; \boldsymbol{C}_2(3,3) = 0.0050; \quad \boldsymbol{C}_3(3,1) = 0.0082; \boldsymbol{C}_3(3,3) = 0.0037;$$

$$\boldsymbol{C}_4(3,1) = 0.0106; \boldsymbol{C}_4(3,3) = 0.0025; \quad \boldsymbol{C}_5(3,1) = 0.0140; \boldsymbol{C}_5(3,3) = 0.0006; \quad \boldsymbol{D}_1(3,2) = 0.3740; \boldsymbol{D}_2(3,2) = 0.4693;$$

$$\boldsymbol{D}_3(3,2) = 0.5816; \boldsymbol{D}_4(3,2) = 0.6243; \quad \boldsymbol{D}_5(3,2) = 0.6418.$$

References

[1] R. Garduno-Ramirez, K.Y. Lee, Wide-range operation of a power P. Chen, J. Shamma, Gain-scheduled ℓ^1-optimal control for boiler–turbine dynamics with actuator saturation, Journal of Process Control 14 (3) (2004) 263–277.
[2] W. Tan, H.J. Marquez, T. Chen, J. Liu, Analysis and control of a nonlinear boiler–turbine unit, Journal of Process Control 15 (8) (2005) 883–891.
[3] U. Moon, K.Y. Lee, A boiler–turbine system control using a fuzzy auto-regressive moving average (FARMA) model, IEEE Transactions on Energy Conversion 18 (1) (2003) 142–148.
[4] R. Garduno-Ramirez, K.Y. Lee, Wide-range operation of a power unit via feedforward fuzzy control, IEEE Transactions on Energy Conversion 15 (4) (2000) 421–426.
[5] J.S. Heo, K.Y. Lee, R. Garduno-Ramirez, Multiobjective control of power plants using particle swarm optimization techniques, IEEE Transactions on Energy Conversion 21 (2) (2006) 552–561.
[6] J.S. Heo, K.Y. Lee, A multi-agent system-based reference governor for multiobjective power plant operation, IEEE Transactions on Energy Conversion 23 (4) (2008) 1082–1092.
[7] R. Dimeo, K.Y. Lee, Boiler–turbine control system design using a genetic algorithm, IEEE Transactions on Energy Conversion 10 (4) (1995) 752–759.
[8] U. Moon, K.Y. Lee, Step-response model development for dynamic matrix control of a drum-type boiler–turbine system, IEEE Transactions on Energy Conversion 24 (2) (2009) 423–430.
[9] M. Keshavarz, M. Barkhordari Yazdi, M.R. Jahed-Motlagh, Piecewise affine modeling and control of a boiler–turbine unit, Applied Thermal Engineering 30 (8–9) (2010) 781–791.
[10] X. Liu, P. Guan, C.W. Chan, Nonlinear multivariable power plant coordinate control by constrained predictive scheme, IEEE Transactions on Control System Technology 18 (5) (2010) 1116–1125.
[11] H. Habbi, M. Zelmat, B. Bouamama, A dynamic fuzzy model for a drum–boiler–turbine system, Automatica 39 (7) (2003) 1213–1219.
[12] Z. Ji, Y. Zhou, Y. Shen, Stabilization of a class of fuzzy control systems via piecewise fuzzy Lyapunov function approach, in: Proc. of the 2007 American Control Conference, 2007, pp. 4065–4070.
[13] R.D. Bell, K.J. Åström, Dynamic models for boiler–turbine alternator units: data logs and parameter estimation for a 160MW unit, Tech. Rep. Report LUTFD2/(TFRT-3192)/1-137, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1987.
[14] T.T. Georgiou, On the computation of the gap metric, Systems Control Letters 11 (4) (1988) 253–257.
[15] T.T. Georgiou, M. Smith, Optimal robustness in the gap metric, IEEE Transactions on Automatic Control 35 (6) (1990) 673–686.
[16] S.C. Shin, S.B. Park, GA-based predictive control for nonlinear processes, Electronics Letters 34 (20) (1998) 1980–1981.
[17] H. Sarimveis, G. Bafas, Fuzzy model predictive control of non-linear process using genetic algorithms, Fuzzy Sets and Systems 139 (1) (2003) 59–80.
[18] W. Naeem, R. Sutton, J. Chudley, An online genetic algorithm based model predictive control autopilot design with experimental verification, International Journal of Control 78 (14) (2005) 1076–1090.

[19] U. Yuzgec, Y. Becerikli, M. Turker, Nonlinear predictive control of a drying process using genetic algorithms, ISA Transactions 45 (4) (2006) 589–602.
[20] C. Onnen, R. Babuška, U. Kaymak, J.M. Sousa, H.B. Verbruggen, R. Isermann, Genetic algorithm for optimization in predictive control, Control Engineering Practice 5 (10) (1997) 1363–1372.
[21] M.G. Na, B.R. Upadhyaya, Model predictive control of an SP-100 space reactor using support vector regression and genetic optimization, IEEE Transactions on Nuclear Science 53 (4) (2006) 2318–2327.
[22] D.Q. Mayne, J.B. Rawlings, Constrained model predictive control: stability and optimality, Automatica 36 (6) (2000) 789–813.
[23] J.M. Maciejowski, Predictive Control with Constraints, Prentice-Hall, New York, 2002.
[24] H. Chen, F. Allgöwer, A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability, Automatica 34 (10) (1998) 1205–1217.