2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Feasible Constrained Nonlinear Predictive Control on Power Plant

X.J. Liu and L.X. Niu

Abstract—**Thermal power plant is required to ensure a fast load change without violating thermal constraints. While model predictive control has been widely used in power plant, incorporating of constraints is a major problem. Two alternative methods of exploiting the nonlinear predictive control are described. One is the input-output feedback linearization technique. The other is the neuro-fuzzy networks(NFNs). Steam-boiler generation control using the two nonlinear predictive methods show satisfactory results and improved performance compared with conventional predictive method. Comparing results considering both the integral absolute value and the relative optimization time needed for completing the simulation have also been addressed in detail.**

I. Introduction

In modern power plant, the coordinated control (CC) scheme is responsible for driving the boiler-turbine-generator set as a single entity, harmonising the slow response of the boiler with the faster response of the turbine-generator, to achieve fast and stable unit response during load tracking manoeuvres and load disturbances. The CC scheme is also crucial to load frequency control(LFC), which is one of the most important issues in power system design and operation, because the objective of the LFC in an interconnected power system is to maintain the frequency of each area and to keep tie-line power near to the scheduled values by adjusting the MW outputs of the LFC generators so as to accommodate fluctuating load demands.

Model predictive control(MPC) has been widely used in power plant in recent years[1][2]. However, due to the nonlinearity and complexity, constraint handling has seldom been incorporated in power plant real-time control application. In using MPC, if the process is linear, the optimum predicted control trajectory is defined through the on-line solution of a quadratic programming problem. For nonlinear system, since the on-line optimization problem is generally nonconvex, the on-line computation demand is high for any reasonably nontrivial systems. Thus, practical implementation of the exact optimization approach is difficult, if not impossible.

One approach for solving this problem is to linearise the nonlinear system via input/output linearization. For certain class of nonlinear systems, such transformation yields a linear dynamic system but with state-dependent(and, in general, nonlinear) constraints which need to be approximated in order to generate a computationally simpler optimization problem. An attractive feature of this approach is that only a quadratic program needs to be solved on-line in real time. However, the design may be overly conservative in some cases: linear approximation is only valid when both state and input do not deviate too much from where they are linearised. This implies that the control actions have to be close to their linearised values in order to preserve stability.

Using a neuro-fuzzy network to learn the plant model from operational process data is another solution[3]. Associate memory network (AMN) is one of the earliest attempts to use neural networks to implement the desired mapping for fuzzy systems. These networks therefore, by embodying both the qualitative and the quantitative approaches, enable heuristic information to be incorporated into and inferred from the network, and allow fuzzy learning rules to be derived.

Considering power plant coordinate system, this article describes how constraint handling can be incorporated in the nonlinear control scheme while ensuring the highest possible rate of load change. Two methods are proposed for nonlinear MPC. In the first one, a solution based on the input-output feedback linearization(IOFL) of the power plant multivariable system is proposed. The main advantage of this approximate IOFL scheme is that it leads to a linear input mapping which helps a further integration of input constraints in the MPC framework. In the second approach, it is shown how the nonlinear neurofuzzy modelling technique, which retains some of the insights obtained from linear systems, can be integrated within an MPC framework. Coordinated control in steam-boiler generation is presented to illustrate the implementation and the performance of the proposed nonlinear constraints MPC. Further, comparing results considering both the integral absolute value and the relative optimization time needed for completing the simulation have also been addressed.

II. Linear MPC based on IOFL

For MPC, input-output feedback linearization(IOFL) is a method to find a static state feedback control law Ψ, such that the resulting closed loop system has a desired linear input-output behavior[4]. In Fig.1, the relation between the power plant output y and the input u is nonlinear while y is linearly related with the newly created external singal v. The input-output feedback linearising control law describes a

Manuscript received September 21, 2007. This work was supported in part by Program for New Century Excellent Talents in University under Grant NCET-06-0207, Natural Science Foundation of Beijing under Grant 4062030

X.J. Liu is with the Department of Automation, North China Electric Power University , Beijing, 102206, P.R. China (corresponding author. Tel: 86-10-80798549; e-mail: liuxj@ncepu.edu.cn).

L.X. Niu is with the Department of Automation, North China Electric Power University , Beijing, 102206, P.R. China (e-mail: niulixia@ncepu.edu.cn).

nonlinear and state-dependent relation between the process input u and the external input v:

$$u = \Psi(x, x^l, v)$$

where x and x^l represent the state vector information from the process and from the desired resulting linear system, respectively. For a square discrete-time affine nonlinear input-output system with p inputs $\mathbf{u}_k = (u_{1,k}, \cdots, u_{p,k})^T$ and p outputs $\mathbf{y}_k = (y_{1,k}, \cdots, y_{p,k})^T$, with relative degree r equal to 1 for all outputs:

$$\mathbf{y_{k+1}} = \mathbf{f(x_k)} + \mathbf{G(x_k)u_k} \tag{1}$$

where $\mathbf{y}_{k+1} = (y_{1,k+1}, \cdots, y_{p,k+1})^T$, the state consists of delayed process inputs and outputs defined as $\mathbf{x}_k = (y_{1,k}, \cdots, y_{1,k-n_l}, \cdots, y_{p,k-n_p}, u_{1,k-1}, \cdots, u_{p,k-m_p})^T$, and with matrix $\mathbf{G}$ invertible for $\forall \mathbf{x} \in X$. Then it is obvious that choosing the control action as:

$$\mathbf{u_k} = \mathbf{p}(\mathbf{x_k}, \mathbf{x}_k^l) + \mathbf{Q(x_k)v_k} \tag{2}$$

with $\mathbf{p}(\mathbf{x_k}, \mathbf{x}_k^l)$ and $\mathbf{Q(x_k)}$ known at time instant k, and given by:

$$\mathbf{p}(\mathbf{x_k}, \mathbf{x}_k^l) = \mathbf{G}^{-1}(\mathbf{x_k})\{-\mathbf{f(x_k)} + \mathbf{CA}\mathbf{x}_k^l\} \tag{3}$$

$$\mathbf{Q(x_k)} = \mathbf{G}^{-1}(\mathbf{x_k})\mathbf{CB} \tag{4}$$

results in the MIMO ($p \times p$) linear system described by

$$\mathbf{y_{k+1}} = \mathbf{CAx_k^l} + \mathbf{CBv_k} \tag{5}$$

$\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are properly chosen state-space matrix.

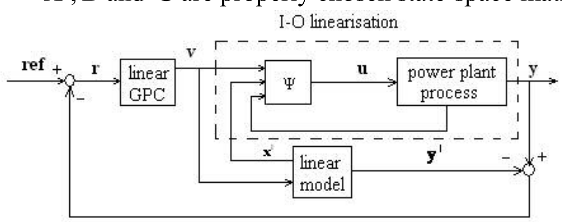


Fig.1. Linear model-based predictive control based on IOFL

From the overall control scheme depicted in Fig.1, the criterion function to be minimized on every sampling instant k is a quadratic criterion on $\tilde{\mathbf{v}} = (\mathbf{v_k}, \cdots, \mathbf{v_{k+H_p-1}})^T$, with H_p being the prediction horizon, given by :

$$J(\tilde{\mathbf{v}}) = (\tilde{\mathbf{y}} - \tilde{\mathbf{r}})^T(\tilde{\mathbf{y}} - \tilde{\mathbf{r}}) + \Delta\tilde{\mathbf{v}}^T \mathbf{W}_v \Delta\tilde{\mathbf{v}} \tag{6}$$

where $\tilde{\mathbf{y}} = (\mathbf{y}_{k+1}^l, \cdots, \mathbf{y}_{k+H_p}^l)^T$, represents the vector with the predicted linear system outputs over H_p, $\tilde{\mathbf{r}} = (\mathbf{r}_{k+1}, \cdots, \mathbf{r}_{k+H_p})^T$ is the vector with the differences between the future reference signals and the modeling errors, $\mathbf{W}_v$ is a square positive definite diagonal weighting matrix of the controller outputs, and $\Delta\tilde{\mathbf{v}} = (v_k - v_{k-1}, \cdots, v_{k+H_p-1} - v_{k+H_p-2})^T$ is the vector which contains the changes in the future control signal. An expansion of the linear system outputs over H_p results in:

$$\tilde{\mathbf{y}} = \mathbf{R}_x \mathbf{x}_k^l + \mathbf{R}_u \tilde{\mathbf{v}}^T \tag{7}$$

with $\mathbf{R}_x$ and $\mathbf{R}_u$ being expansion matrices of the linear state-space description matrices A, B and C. The incorporation of eqn.7 in eqn.6 enables the optimization routine of the linear MPC to find the optimal $\tilde{\mathbf{v}}$ by solving a simple analytical expression. In the presence of level and rate inequality constraints acting on the power plant process inputs , e.g,:

$$\mathbf{u}_{\min} < \tilde{\mathbf{u}} < \mathbf{u}_{\max} \tag{8}$$

$$\Delta\mathbf{u}_{\min} < \Delta\mathbf{u} < \Delta\mathbf{u}_{\max} \tag{9}$$

with $\tilde{\mathbf{u}} = (\mathbf{u}_k, \cdots, \mathbf{u}_{k+H_p-1})^T$

and $\Delta\tilde{\mathbf{u}} = (\mathbf{u}_k - \mathbf{u}_{k-1}, \cdots, \mathbf{u}_{k+H_p-1} - \mathbf{u}_{k+H_p-2})^T$

the minimization of equation (6) subjected to equations (8) and (9) can only be efficiently expressed in terms of the optimization variable $\tilde{\mathbf{v}}$. In this case, fast and reliable QP optimization routines can be used to find the solution of the adopted nonlinear feedback linearising control law Ψ.

The introduction of an extended version of equation (2) over the prediction horizon H_p results in the following nonlinear and state-dependent mapping:

$$\tilde{\mathbf{u}}' = \tilde{\mathbf{G}}^{-1}(\mathbf{x_k}, \tilde{\mathbf{u}}')\{-\mathbf{f}(\mathbf{x_k}, \tilde{\mathbf{u}}') + \mathbf{R}_x \mathbf{x}_k^l\} + \tilde{\mathbf{G}}^{-1}(\mathbf{x_k}, \tilde{\mathbf{u}}')\mathbf{R}_u \tilde{\mathbf{v}} \tag{10}$$

with $\tilde{\mathbf{u}} = (\mathbf{u}_k, \cdots, \mathbf{u}_{k+H_p-2})^T$. Since $\tilde{\mathbf{u}}'$ contains the optimal process inputs over future time steps, the knowledge of this sequence depends on the computation of the optimal control sequence $\tilde{\mathbf{v}}$, which in turn, depends on the knowledge of constraints on future time steps. Therefore, in order to use QP, an adaptive procedure for correcting the constraints linearization error is adopted to guarantee a feasible control sequence for the complete prediction horizon. This technique is fully developed in [4], where the control of a nonlinear chemical reactor was tested.

III. Neuro-fuzzy Network Predictive Control

A. Structure of the Neuro-fuzzy Network

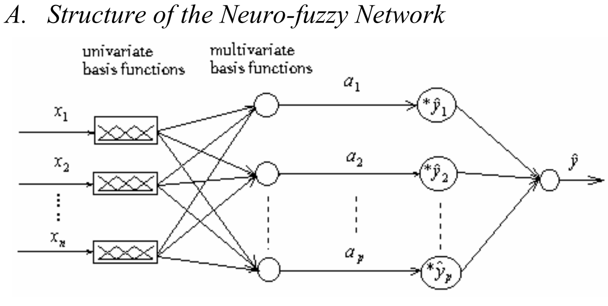


Fig.2. B-spline neuro-fuzzy network

For the MIMO nonlinear dynamic system (1), let the local linear model at the operating point O(t) be given by:

$$\mathbf{x_{k+1}} = \mathbf{Ax_k} + \mathbf{Bu_k}$$
$$\mathbf{y_k} = \mathbf{Cx_k} \tag{11}$$

Note that the matrix $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are a function of the operating point O(t). The nonlinear system (1) is partitioned into several operating regions, such that each region can be approximated by a local linear model. Since NFNs is a class of associative memory networks with knowledge stored locally [3], they can be applied to model this class of nonlinear systems. A schematic diagram of the NFN is shown in Fig. 2, where the membership functions are given by B-spline basis functions. The input of the network is the antecedent variable $[x_1, x_2 \cdots x_n]$, and the output, $\hat{\mathbf{y}}(t)$, is a weighted sum of the output of the local linear models $\hat{\mathbf{y}}_i(t)$. B-spline functions are used as the membership functions in the NFNs for the following reasons [3]:

1) B-spine functions can be readily specified by the order of the basis function and the number of inner knots.

2) They are defined on a bounded support, and the output of the basis function is always positive, i.e., $\mu_k^j(x) = 0,\ x \notin [\lambda_{j-k}, \lambda_j]$ and $\mu_k^j(x) > 0$, $x \in (\lambda_{j-k}, \lambda_j)$.

3) The basis functions form a partition of unity, i.e., $\sum_j \mu_k^j(x) \equiv 1,\ x \in [x_{\min}, x_{\max}]$.

4) The output of the basis functions can be obtained by a recurrence equation.

The NFN shown in Fig. 2 consists of the following fuzzy rules:

IF operating condition i (x_1 is positive small, $\cdots$, and x_n is negative large)

THEN the output is given by the local discrete-time state space model i :

$$\begin{aligned} \mathbf{x}_{\mathbf{k+1}} &= \mathbf{A}_i \mathbf{x}_{\mathbf{k}} + \mathbf{B}_i \mathbf{u}_{\mathbf{k}} \\ \mathbf{y}_{i\mathbf{k}} &= \mathbf{C}_i \mathbf{x}_{\mathbf{k}} \end{aligned} \tag{12}$$

where $\mathbf{A}_i$, $\mathbf{B}_i$ and $\mathbf{C}_i$ are the linear state-space matrices. The multivariate basis function $a_i(x)$, is obtained by the tensor products of the univariate basis functions $\mu_{A_k^i}(x_k)$,

$$a_i(x) = \prod_{k=1}^{n} \mu_{A_k^i}(x_k) \ ; \quad \text{for} \quad i = 1, 2, \cdots, p \tag{13}$$

where n is the dimension of the input vector x, and p, the total number of weights in the NFN, is given by,

$$p = \prod_{i=1}^{n} (R_i + k_i) \tag{14}$$

where k_i and R_i are the order of the basis function and the number of inner knots respectively. The properties of the univariate B-spline basis functions described previously also apply to the multivariate basis functions. The output of the NFN with p fuzzy rules is,

$$\hat{\mathbf{y}}(k) = \frac{\sum_{i=1}^{p} \hat{\mathbf{y}}_i(k) a_i(x)}{\sum_{i=1}^{p} a_i(x)} = \sum_{i=1}^{p} \hat{\mathbf{y}}_i(k) a_i(x) \tag{15}$$

B. Neurofuzzy predictive control

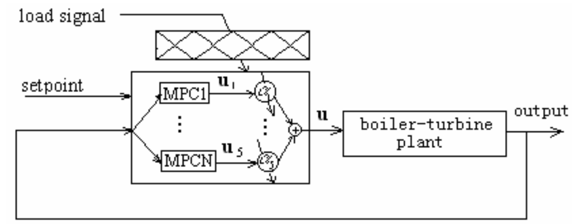


Fig.3. Neuro-fuzzy MPC

The control structure here consists of the family of controllers and the scheduler. At each sample instant the latter decides which controller, or combination of controllers, to apply to the process. Generally, the controllers are tuned about a model obtained from experiments at a particular equilibrium point, since linear model and controllers are quite well understood. They are constructed by interpolating between the members of a family of linear controllers. In this article, these interpolating functions are realized by B-spline neuro-fuzzy networks.

As already described, the neurofuzzy network consists of a set of locally valid submodels together with an appropriate interpolation function. A controller is then designed about each of the local models. The interpolated outputs are then summed and used to supply the control commands to the process. The resultant neuro-fuzzy MPC structure is shown in Fig.3. The interpolation function effectively smoothes the transition between each of the local controllers. In addition, the transparency of the nonlinear control algorithm is improved as the operating space is covered using controllers rather than models.

C. Constraint handling

One of the main application benefits of using a linear predictive controller is its ability to handle process constraints directly within the control law. While the quadratic programming is applied to the neuro-fuzzy MPC, a problem arises, as there is no way of knowing that the summation of all of the controller outputs will not in fact violate a process constraint. Notice that a B-spline neuro-fuzzy network is used, in which the third property signifies that the basis functions form a partition of unity. In such a way, the summation of all of the controller outputs will not in fact violate a process constraint, since they are weighted sum by the normalized B-spline neuro-fuzzy network. Taking consideration of a second order B-spline neuro-fuzzy network. At any time instant, if one variable account for 80% of the total output, another variable will surely account for 20% of the total output. This may be the main attracting factor of the B-spline neuro-fuzzy network while applying to constraint MPC.

IV. Steam-Boiler Generation Control

The essential dynamics of a power plant have been remarkably captured for a 160MW oil fired drum-type boiler-turbine-generator unit in a third order MIMO nonlinear model for overall widerange simulations in [5]. The inputs are the position of valve actuators that control the mass flow rates of fuel, steam to the turbine, and feedwater to the drum. The three outputs are electric power, drum steam pressure, and drum water level deviation. The three state variables are electric power, drum steam pressure, and fluid density. The state equations are:

$$\frac{dP}{dt} = 0.9u_1 - 0.0018u_2P^{9/8} - 0.15u_3$$
$$\frac{dE}{dt} = ((0.73u_2 - 0.16)P^{9/8} - E)/10$$
$$\frac{d\rho_f}{dt} = (141u_3 - 1.1u_2 - 0.19)P)/85 \qquad (16)$$

The drum water level output is calculated using the following algebraic equations:

$$q_e = 0.85u_2 - 0.18)P - 45.59u_1 - 2.51u_3 - 2.09$$
$$\alpha_s = (1/\rho_f\, 0.9u_1 - 0.0015)/1/(0.8P - 25.6) - 0.0015 \qquad (17)$$

A. Modelling Phase

A feedforward neural network is used for modeling the dynamics of the system. The networks are trained based on measured input-output data taken from the process simulation, the identification data set, and validated on a "fresh" validation data set. After completing the learning phase, the following affine neural network model resulted:

$$\hat{y}_{k+1} = f_a(x_k^a) + g_a(x_k^a)u_k \qquad (18)$$

Where the state vector is given by $x_k^a = [y_k, y_{k-1}, y_{k-2}, u_k, u_{k-1}]$. The best structure found for each f_a and g_a consists of one hidden layer feedforward neural network having three neurons with tangent hyperbolic activation function. The resulting IOFL linear model are obtained through a Jacobian linearisation of the neural network around a stable equilibrium point, and given by:

$$\mathbf{x}_{k+1} = \begin{bmatrix} -0.0013 & 0 & 0 \\ -0.029 & -0.1 & 0 \\ -0.0032 & 0 & 0 \end{bmatrix}\mathbf{x}_k + \begin{bmatrix} -0.9 & 0 & -0.15 \\ 0 & 0.073 & 0 \\ 0 & -1.56 & 1.66 \end{bmatrix}\mathbf{v}_k$$
$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.0025 & 0 & 0.005 \end{bmatrix}\mathbf{x}_k \qquad (19)$$

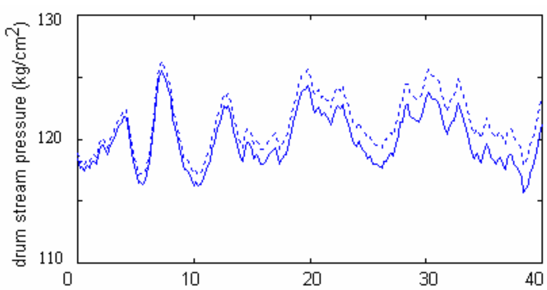


Fig.4 Steam-boiler system(solid line) and IOFL model (dotted line)

The validation results on the fresh data is shown in Fig.4. The IOFL model can only give good approximation over fixed, but not the full operating range, illustrating that the plant is highly nonlinear.

A valid neurofuzzy model of the plant, which is an essential tool for the developing of the neuro-fuzzy MPC, could also be derived. A method to handle the nonlinearity in the plant is to divide the full operating range into a number of local operating regions, where linear models with a good approximation are used. These models are incorporated into the neurofuzzy model for modelling the nonlinear plant. Load is used to select the division between the local regions in the NFN. Based on this approach, the load is divided into five regions, using also the experience of the operators, who regard a load of 160MW as high, 140MW as medium high, 120MW as medium, 100MW as medium low and 80MW as low. This type of partitioning ensures that any change in the real-valued input signal will be reflected by a change in the degree of membership. After training, the neurofuzzy models are:

if LOAD is high:

$$\mathbf{x}_{k+1} = \begin{bmatrix} -0.003 & 0 & 0 \\ -0.0294 & -0.1 & 0 \\ -0.0097 & 0 & 0 \end{bmatrix}\mathbf{x}_k + \begin{bmatrix} -0.9 & 0 & -0.15 \\ 0 & 0.073 & 0 \\ 0 & -1.69 & 1.66 \end{bmatrix}\mathbf{u}_k$$
$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.00053 & 0 & 0.0065 \end{bmatrix}\mathbf{x}_k$$

if LOAD is medium high:

$$\mathbf{x}_{k+1} = \begin{bmatrix} -0.0029 & 0 & 0 \\ -0.0291 & -0.1 & 0 \\ -0.0093 & 0 & 0 \end{bmatrix}\mathbf{x}_k + \begin{bmatrix} -0.9 & 0 & -0.15 \\ 0 & 0.073 & 0 \\ 0 & -1.43 & 1.66 \end{bmatrix}\mathbf{u}_k$$
$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.00050 & 0 & 0.0065 \end{bmatrix}\mathbf{x}_k$$

if LOAD is medium:

$$\mathbf{x}_{k+1} = \begin{bmatrix} -0.0027 & 0 & 0 \\ -0.0288 & -0.1 & 0 \\ -0.0088 & 0 & 0 \end{bmatrix}\mathbf{x}_k + \begin{bmatrix} -0.9 & 0 & -0.15 \\ 0 & 0.073 & 0 \\ 0 & -1.30 & 1.66 \end{bmatrix}\mathbf{u}_k$$
$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.00048 & 0 & 0.0065 \end{bmatrix}\mathbf{x}_k$$

if LOAD is medium low:

$$\mathbf{x}_{k+1} = \begin{bmatrix} -0.0025 & 0 & 0 \\ -0.0285 & -0.1 & 0 \\ -0.0081 & 0 & 0 \end{bmatrix}\mathbf{x}_k + \begin{bmatrix} -0.9 & 0 & -0.15 \\ 0 & 0.073 & 0 \\ 0 & -1.17 & 1.66 \end{bmatrix}\mathbf{u}_k$$
$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.00044 & 0 & 0.0065 \end{bmatrix}\mathbf{x}_k$$

if LOAD is low:

$$\mathbf{x}_{k+1} = \begin{bmatrix} -0.0023 & 0 & 0 \\ -0.0280 & -0.1 & 0 \\ -0.0075 & 0 & 0 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} -0.9 & 0 & -0.15 \\ 0 & 0.073 & 0 \\ 0 & -1.04 & 1.66 \end{bmatrix} \mathbf{u}_k$$

$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.00040 & 0 & 0.0065 \end{bmatrix} \mathbf{x}_k$$

Fig.5 shows comparison of boiler system and neurofuzzy model while load fluctuates within a 40MW range. Good generalization result of the drum steam pressure over the full operating range is obtained from the neurofuzzy model.

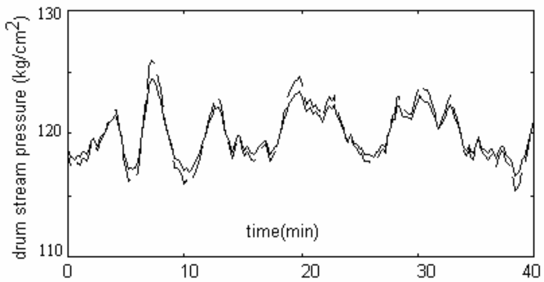


Fig.5 Steam-boiler system(solid line) and neurofuzzy model (dashed line)

B. Control results

In the neuro-fuzzy MPC, the nonlinear controller consists of five local controllers, each one of which is designed about one of the local models, and thus each with a set of tuning parameters. At each sample instant the load signal was fed to the interpolation membership function of the B-spline NFNs, which in turn generates the activation weights for each of the local controllers. Notice that, since the B-spline membership function was chosen to be second order, there are two controllers working at any time instant.

In the IOFL predictive control, the global optimization problem was transformed by quadratic programming(QP) techniques. In order to guarantee a feasible solution without constraints violation over the complete prediction horizon, a convergent algorithm integrates the QP problem on an iterative scheme is incorporated. This approximate method is proposed which simultaneously guarantees a feasible solution without constraints violation over the complete prediction horizon within a finite number of steps.

The linear MPC in neuro-fuzzy controller is obtained by minimizing the following cost function [2],

$$J = E\{\sum_{j=1}^{N}[\hat{y}(t+j) - y_r(t+j)]^2\} + \sum_{j=1}^{M}\lambda_j[\Delta u(t+j-1)]^2 \quad (20)$$

subject to $\quad -0.007 < \dfrac{du_1}{dt} < 0.007\ ,\ -2.0 < \dfrac{du_2}{dt} < 0.02$

$$-0.05 < \frac{du_3}{dt} < 0.05$$

Where $\lambda_j = 0.1 \times I$. The same constraint applies to the I-O linearization controller where in equation (6) $W_v = 0.1 \times I$ to guarantee an equal comparison. The sampling interval in both controllers is chosen to be 5s.

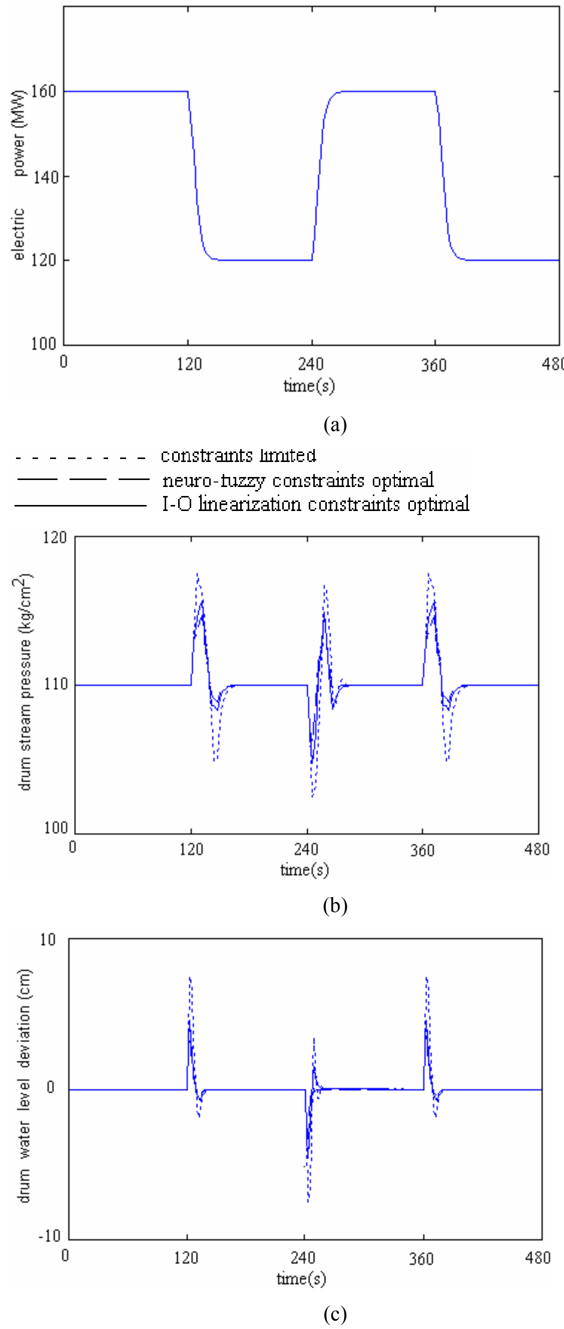
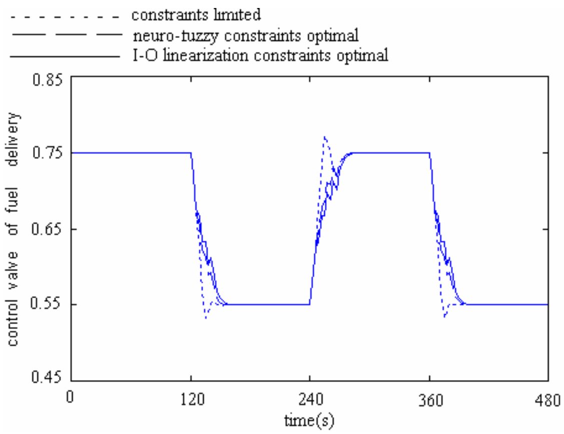
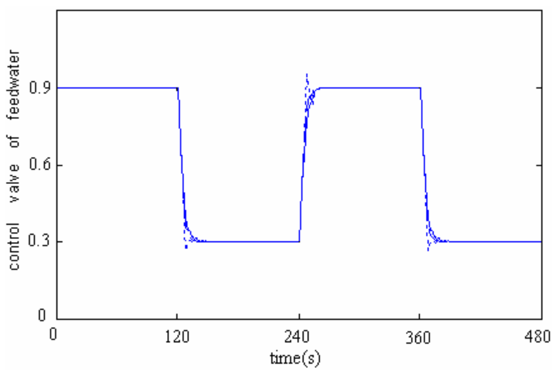


Fig.6 Steam pressure and water level transient processes

Fig.6 shows steam pressure and water level transient processes under planned load changes. The control efforts are shown in Fig.7. The neuro-fuzzy constraints optimal controller exhibits superior action, since it is based on a more exact model of the plant. The control signal seems to be better in anticipating the effect of the rate actuator limit. This is followed by the IOFL constraints optimal controller, due to its ability to cope with constraint. The constraints limited controller offers the worst result, due to the lack of constraint handling ability. The maximum deviation of drum steam pressure for these three methods are 4.64kg/cm^2, 5.64kg/cm^2 and 7.50kg/cm^2 respectively. The maximum deviation of drum water level for these three method are 3.55cm, 4.55cm and 7.50cm respectively.



(a)



(b)

Fig.7 The control efforts

One of the most concerning problem in power plant real-time control is the computing burden. For MPC, choosing of larger predictive horizon can get better performance, nevertheless, resulting in increment of computing burden. For further comparison, the close-loop performance of each controller is analysed with respect to two variables: the integral absolute value and the relative optimization time needed for completing the simulation. The comparison between the two configurations which guarantee a feasible control solution over the complete prediction horizon show a similar trend performance in terms of the closed-loop tracking error, although the constraints optimal neuro-fuzzy MPC is less computational time demanding, as shown in Fig.8. This can be attributable to the accurate off-line neuro-fuzzy model, while IOFL optimization infers to an iterative process, which is time consuming. The trade-off between the total computational demands and the gain in optimality is clearly favourable to the constraints optimal neuro-fuzzy MPC.

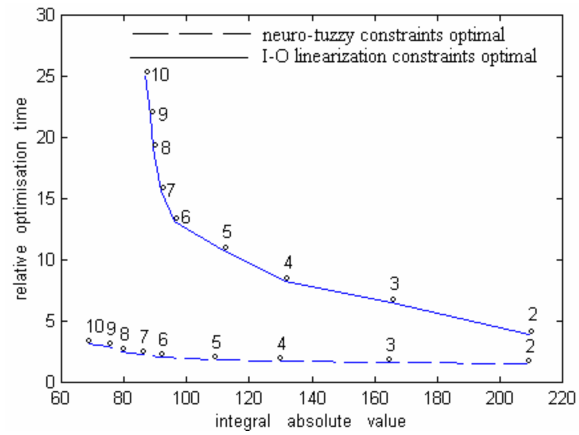


Fig.8 Comparison of controller performance for different prediction horizon. $H_p = 2,\cdots,10$ for I-O linearization; $N = 2,\cdots,10$ for neuro-fuzzy.

V. Conclusion

Constraint handling is the major advantage of using a linear predictive controller in power plant. For nonlinear process, this can usually lead to large differences between the actual and predicted output values when the current output is relatively far away from the operating point at which the linear control model was generated. This paper introduces IOFL and NFNs to formulate feasible solution.

The IOFL-MPC requires a constraints handling approximation procedure to guarantee the feasibility of the iterative QP optimization. In the neuro-fuzzy MPC, a set of local controllers were combined through NFNs to form a local controller network. The proposed nonlinear MPCs were applied in the simulation of the power plant control. Better results are obtained when compared with the constraint limited MPC. The proposed nonlinear controllers therefore offer a feasible and reliable solution for optimization of the control moves.

References

[1] G. Prasad, E. Swidenbank and B.W. Hogg, "A neural net model-based multivariable long-range predictive control strategy applied thermal power plant control," *IEEE Trans. Energy Conversion,* vol. 13, no. 2, pp. 176-182, June 1998.

[2] X.J. Liu, and CW. Chan, "Neuro-fuzzy generalized predictive control of boiler steam temperature," *IEEE Trans. Energy Conversion,* vol. 21, no.4, pp. 900-908, December, 2006.

[3] X.J. Liu, F. Lara-Rosano and C.W. Chan, "Model-reference Adaptive Control Based on Neurofuzzy Networks ," *IEEE Trans. Systems, Man and Cybernetics C,* vol. 34, no. 3, pp. 302-309, August, 2004.

[4] H. A. B. te Braake, et.al., Linear predictive control based on approximate input-output feedback linearization. *IEE Proc.-Control Theory and Application*. Vol.146, no.4, 295-300, 1999.

[5] R.D.Bell and K.J.Astrom, "Dynamic Models for Boiler-Turbine-Alternator Units: Data Logs and Parameter Estimation for a 160MW Unit," Lund Institute of Technology, Sweden, Rep. TFRT-3192, 1987.