Article

Transactions of the Institute of Measurement and Control
2015, Vol. 37(8) 952–960

DOI: 10.1177/0142331214550681
tim.sagepub.com
SAGE

Centralized predictive controller design based on an uncertain neural network for multivariable non-linear systems

Ahmed Mnasser[1] and Faouzi Bouani[2]

Abstract

In this paper, we propose a feedforward neural networks-based robust predictive controller for a class of multi-input–multi-output non-linear systems. Using the structured uncertainties of the output layer's weights of the neural networks model, the non-linear model of the real system is determined at each operating point. The control law is formulated as a minimax problem, which is solved online. The non-convex optimization is developed by minimizing the worst case of the objective cost function, taking into account the uncertainties of the non-linear model and the input control signal constraints. The efficiency of the proposed neural predictive controller is illustrated, in simulation, with a multivariable system example.

Keywords

Artificial neural networks, MIMO systems, minimax optimization, predictive control

Introduction

Most industrial processes are multivariable systems with a high interrelationship between their input–output variables. Nowadays, industrial systems require controllers with a high degree of complexity, and the implementation of conventional techniques of control for such a class of systems are not capable of achieving the desired control (Behera and Kar, 2010). Model predictive control (MPC) is one of the most advanced strategies that can be used to control non-linear systems. Due to MPC's ability to deal with constraints in the control problem imposed on process inputs and outputs, it has been considered a successful tool for the control of industrial processes. MPC is based on the use of a process model in order to predict future outputs over a certain horizon (Lawrynzuk and Tatjewski, 2010). The success of this technique is related to the precision of the system model to be controlled. Several studies have investigated providing the solutions of control problems associated with the MPC technique for single-input–single-output non-linear systems (Morari, 2009). The stabilization formulations of this class of method have been studied in the literature (Lu et al., 2010). In Holkar and Waghmare (2010), the authors provided details of most control design methods based on MPC concepts that have been implemented in industrial processes. Subsequently, the MPC technique can be easily extended to deal with multivariable systems.

Model-based controllers that are tested with linear models have been successfully used in industrial processes. However, as most industrial processes are characterized by non-linearity and complexity, this technique is not often adopted for cases where the real time control needs model changes and different operating points (Ho et al., 2012). In fact, it is necessary to use a non-linear model to describe the unknown non-linear dynamics of a real system. Several types of non-linear models have been used in the literature, such as the polynomial model (Hernandez and Arkun, 1993), Volterra model (M'Salhi et al., 2001) and neural networks model (Narendra and Parthasarathy, 1990).

Most of the research has used neural networks for modelling non-linear systems, because of their inherent ability to learn and to approximate any non-linear function, since the establishment of the universal approximator properties (Hornik et al., 1989). Feedforward neural networks (FNNs) are used effectively for identification of multi-input–multi-output (MIMO) non-linear systems. It has been proved that the FNN can approximate any continuous non-linear function (Wang et al., 2012; Zhang et al., 2012). The successful application of a predictive controller based on a neural network model for non-linear systems has appeared in an accurate non-linear model and an efficient control algorithm (Fei et al., 2006; Mnasser et al., 2013; Pan and Wang, 2012). In

[1]Analysis, Conception and Control of Systems Laboratory, Faculty of Sciences of Tunis, Tunis El Manar University, Tunisia
[2]Analysis, Conception and Control of Systems Laboratory, National Engineering School of Tunis, Tunis El Manar University, Tunisia

Corresponding author:
Ahmed Mnasser, Analysis, Conception and Control of Systems Laboratory, Faculty of Sciences of Tunis, Tunis El Manar University, Tunisia.
Email: mnasser_a@yahoo.fr

MPC, the process model is used to compute the output prediction sequence over the prediction horizon. In presence of a non-linear model, the predictor is then a non-linear function relative to the control sequences. Consequently, the optimization problem is non-convex.

Kotta et al. (2006) have presented the realizability of the FNNs based on an input–output model to control non-linear systems. An adaptive learning algorithm is used to identify the unknown parameters in the neural network model (Cheng et al., 2007). For decades, robust optimal control problems have appeared in various research studies. Some formulations of robust predictive controllers consider additive uncertainty (Bemporad et al., 2003; Raimondo et al., 2009) or structured uncertainty (Bouzouita et al., 2007; Yan and Wang, 2014). Kheriji et al. (2011) proposed a robust predictive controller for a class of constrained MIMO systems based on a linear model with parametric uncertainty. Yan and Wang (2014) presented a neural networks approach based on predictive control design for non-linear systems. In fact, an unknown non-linear dynamical system is decomposed by means of Jacobian linearization. Researchers have used recurrent neural networks not only for modelling systems via supervised learning, but also for optimizing the quadratic programming problem control.

The aim of this work consists of the development of a MIMO robust predictive controller based on a non-linear model having uncertain parameters. FNNs are used to approximate the behaviour of the plant. Taking into account that the neural model will not always assume an exact representation of the non-linear system being controlled, parametric uncertainties of connection weights of the output layer of the neural model are introduced. The predictive controller design procedure is implemented to highly non-linear MIMO systems based on only one neural network model. The proposed model consists of the connection weights of the output layer of the neural network, which are bounded uncertainties, whereas the connection weights from inputs to hidden neurons are fixed. The neural network structure can be used to model a wide class of non-linear systems with a reduced number of unknown parameters. The values of the control signals are determined at each time instant by the minimization of the worst-case quadratic cost in taking into account bounded parameters of connection weights of the output layer of the neural model and the control signals constraints. Therefore, the minimax optimization problem, which is non-convex relative to the unknown parameters, will be solved online (Magni et al., 2009). As a result, the proposed controller based on a single neural networks model assumes good closed-loop performances at all system operating points.

The outline of the paper is organized as follows. A problem formulation and preliminaries on model predictive control strategy and the structure of the FNNs are given in the next section. In the third section, a robust predictive controller design based on neural networks is described. In the fourth section, an example to illustrate the performance of the proposed approach is illustrated. Finally, the last section is dedicated to conclusions.

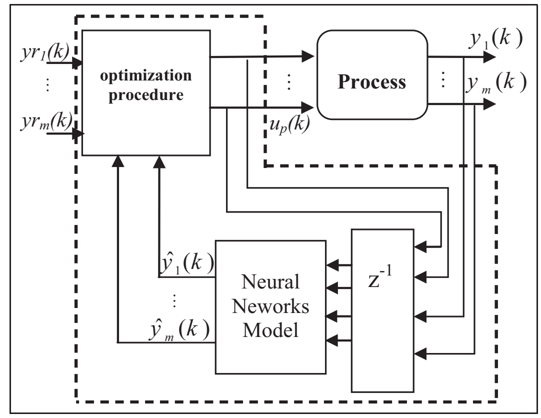

Figure 1. Multi-input–multi-output (MIMO) controller based on a feedforward neural network (FNN) model.

Problem formulation and control design

An introduction to the properties of the model predictive control and the FNNs is given in this section. Consider a MIMO non-linear system that has p inputs denoted $[u_1(k), \ldots, u_p(k)]$ and m outputs denoted $[y_1(k), \ldots, y_m(k)]$ in which $p \leq m$.

Control law

MPC is an optimal control strategy that uses an explicit process model to predict the behaviour of the plant. The control design based on FNNs model for multivariable system is depicted in Figure 1. As shown in Figure 1, the non-linear model is used to compute the predictor sequences. The control law is calculated by the minimization of the cost function $J(\Delta u_n)$, for $n = 1, \ldots, p$, which is given by the following problem:

$$\min_{\Delta u_n} J(\Delta u_n) \tag{1}$$

The cost function is defined as a quadratic objective function, which is given by:

$$\begin{aligned} J(\Delta u_n) = & \sum_{i=1}^{m} \sum_{h=1}^{N_2} (yr_i(k+h) - \hat{y}_i(k+h))^2 \\ & + \lambda \sum_{n=1}^{p} \sum_{h=1}^{N_u} (\Delta u_n(k+h-1))^2 \end{aligned} \tag{2}$$

where m and p represent, respectively, the number of outputs and inputs of the system. N_2 represents the prediction horizon. $yr_i(k)$ and $\hat{y}_i(k+h)$ are, respectively, the reference trajectory and the h-step ahead predictor corresponding to the ith output of the system. The $\Delta u_n(k+h-1)$ denotes the control increment corresponding to pth input, which is given by:

$$\Delta u_n(k+h-1) = u_n(k+h-1) - u_n(k+h-2) \tag{3}$$

$\Delta U_n = [\Delta u_n(k), \ldots, \Delta u_n(k + N_u - 1)]^T \in IR^{pN_u}$ is the vector of the optimization variables, λ is the control input weighting factor. The control signal is manipulated only within the control horizon (N_u) and remains constant afterwards, i.e.

$$\Delta u_n(k + h) = 0, \text{ for } N_u \leq h \leq N_2. \tag{4}$$

Neural networks prediction

In order to predict the future system outputs, a non-linear dynamic process model is employed. In fact, neural networks are able to describe the system behaviour over the whole operating range. Using the available inputs and outputs data, a neural network can be trained to approximate the unknown non-linear function. Then, the model output is given by:

$$\hat{y}_i(k) = NN[X(k)] \tag{5}$$

where the $NN[.]$ is the neural network and $X(k) = [U(k), Y(k)]^T$ is the input vector of the NN model in which $U(k)$ and $Y(k)$ are presented, respectively, by elements:

$$U(k) = \left[u_1(k-1), \ldots, u_1(k-d_u), \ldots, u_p(k-1), \ldots, u_p(k-d_u)\right]$$

$$Y(k) = \left[y_1(k-1), \ldots, y_1(k-d_y), \ldots, y_m(k-1), \ldots, y_m(k-d_y)\right]$$

d_y and d_u represent, respectively, the known upper orders of inputs and the outputs of the system. In the literature, FNNs are considered a powerful tool for modelling non-linear systems. They are structured in successive layers of neurons, from inputs to the output layer by the intermediate hidden layers. Hornik et al. (1989) has proved that a single hidden layer in the FNNs is sufficient for approximating any unknown non-linear function. The number of neurons in the hidden layer of the NN model is selected until an acceptable testing error value is obtained.

The model output $\hat{y}_i(k)$ is given by the following relation:

$$\hat{y}_i(k) = w_{i0} + \sum_{j=1}^{L} w_{ij} s_j(k), \text{ for } i = 1, \ldots, m \tag{6}$$

where L and m are, respectively, the number of nodes in the hidden layer and the number of the output layer. $\hat{y}_i(k)$ is the output of the ith neuron in the output layer. $s_j(k)$ is the output of the jth neuron of hidden layer. w_{ij} denotes the weight from the jth neuron in the hidden layer to the ith neuron in the output layer and the thresholds are denoted by w_{i0} for $i = 1, \ldots, m$.

$$s_j(k) = f(v_j + \sum_{z=1}^{r} v_{jz} I_z(k)) \tag{7}$$

where r represents the number of inputs to the network. v_{jz} and v_j, for $j = 1, \ldots, m$, are the connection weights from inputs to the hidden neurons and biases, respectively. $f(.)$ represents the non-linear activation function of the hidden layer, which should be a continuous differentiable function. In this work, the tangent hyperbolic function is used as a neural network activation function in the hidden layer.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{8}$$

$I(k)$ is the input vector, which can be constructed by the following elements:

$$I(k) = [I_1(k), \ldots, I_r(k)] \tag{9}$$

Hence, the total number of the parameters in the neural networks model is equal to $g = rL + L + Lm + m$.

In order to estimate the parameters of the FNN model, the quadratic error of the neural model to be optimized is given by:

$$E = \frac{1}{2} \sum_{k=1}^{N_m} \sum_{i=1}^{m} (y_i(k) - \hat{y}_i(k))^2 \tag{10}$$

where $y_i(k)$ and $\hat{y}_i(k)$ are, respectively, the desired output and the predicted output of the ith node in the output layer of the neural networks model computed at time k and N_m is the total number of measurements.

In order to handle a large class of non-linear discrete-time systems, we suppose that the outputs layer's weights belong to some prespecified set Ω. The connection weights of the output layer of the FNNs are modelled by the structured uncertainties, which are given by the following relation:

$$w_{ij} \in \left[\underline{w}_{ij}, \bar{w}_{ij}\right], \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, L \tag{11}$$

where $\underline{w}_{ij}$ and $\bar{w}_{ij}$ are the upper and lower bounds values of each connection weight of the output layer of the FNNs. Then, each parameter has uncertain variations about its nominal value. So, it can be represented by means of a median value and an absolute value of the maximum deviation with respect to its median value.

$$w_{ij} = w^0_{ij} + \varepsilon_{ij}\delta w_{ij}, \ \left|\varepsilon_{ij}\right| \leq 1 \tag{12}$$

$$for\, i = 1, \ldots, m \text{ and } j = 1, \ldots, L$$

The feasible region Ω can be given by:

$$\Omega = \left\{ w^0_{ij} + \varepsilon_{ij}\delta w_{ij}, \ \left|\varepsilon_{ij}\right| \leq 1 \right\} \tag{13}$$

The uncertainties are assumed to be bounded, and $\varepsilon_{ij}\delta w_{ij}$ represents the uncertainty of coefficient w_{ij}.

Neural networks-based robust predictive controller design

The main contribution is devoted to the robust MPC controller design for non-linear MIMO systems based on uncertain FNNs. So, the basic idea is to control the multivariable system for each sample time by using a FNN having parametric uncertainty output layer weights. The control law represents the best solution for the worst case defined by the set of uncertain models (Alessio and Bemporad, 2009; Ben-Tal et al., 2009; Lofberg, 2003; Magni et al., 2009; Raimondo and Magni, 2006). Robust model predictive control (RMPC) involves online optimization of a minimax objective problem.

The control law is computed by solving the minimax optimization.

The optimization problem is expressed as:

$$\min_{\Delta u_n} \max_{W \in \Omega} J(\Delta u_n, W), \text{ for } n = 1, \ldots, p \tag{14}$$

Subject to constraints:

$$u_{n\min} \le u_n(k+h-1) \le u_{n\max}, \quad h = 1, \ldots, N_u$$

$$\Delta u_{n\min} \le \Delta u_n(k+h-1) \le \Delta u_{n\max}, \quad h = 1, \ldots, N_u$$

$$W_{\min} \le W \le W_{\max}$$

where $W_{\min}$ and $W_{\max}$ are, respectively, the lower and upper bounded vectors of the connection weights of the output layer of the neural model. $u_{n\min}$ and $u_{n\max}$ are the minimum and maximum values of the nth manipulated input, respectively. $\Delta u_{n\min}$ and $\Delta u_{n\max}$ represent the upper and lower bounds values of the increments control. $J(\Delta u_n, W)$ denotes the desired performance criterion that can be defined by the quadratic function (2) in which

$$W = [w_{11}, \ldots, w_{1L}, \ldots, w_{m1}, \ldots, w_{mL}]^T \in IR^{mL}$$

is the vector of the connection weights of the output layer of the neural networks model.

For the sake of brevity, we consider the following vectors:

$$U(k) = [\, U_1(k) \quad \ldots \quad U_p(k)\,]^T \in IR^{pN_u}$$

$$U_n(k) = [u_n(k), \ \ldots, \ u_n(k+N_u-1)]^T \in IR^{N_u}$$

$U_{\min}(k)$ and $U_{\max}(k)$ are the minimum and maximum vectors of the input control, respectively.

$$U_{\min}(k) = [\, U_{1\min}(k) \quad \cdots \quad U_{p\min}(k)\,]^T \in IR^{pN_u}$$

$$U_{\max}(k) = [\, U_{1\max}(k) \quad \cdots \quad U_{p\max}(k)\,]^T \in IR^{pN_u}$$

$$U_{n\min}(k) = [\, u_{n\min}(k) \quad \cdots \quad u_{n\min}(k+N_u-1)\,]^T \in IR^{N_u}$$

$$U_{n\max}(k) = [\, u_{n\max}(k) \quad \cdots \quad u_{n\max}(k+N_u-1)\,]^T \in IR^{N_u}$$

for $n = 1, \ldots, p$, where $u_{n\min}(k)$ and $u_{n\max}(k)$ are the minimum and maximum values of the nth control input, respectively.

$$W_{\min} = [\underline{w}_{11}, \ldots, \underline{w}_{1L}, \ldots, \underline{w}_{m1}, \ldots, \underline{w}_{mL}]^T \in IR^{mL}$$

$$W_{\max} = [\bar{w}_{11}, \ldots, \bar{w}_{1L}, \ldots, \bar{w}_{m1}, \ldots, \bar{w}_{mL}]^T \in IR^{mL}$$

$\underline{w}_{ij}$ and $\bar{w}_{ij}$ are, respectively, the lower and upper limits weights of each neurons in the output layer of the neural network model.

$\Delta U_{\min}(k)$ and $\Delta U_{\max}(k)$ represent, respectively, the upper and lower bounded vectors of the increments control.

$$\Delta U_{\min}(k) = [\Delta U_{1\min}(k) \quad \ldots \quad \Delta U_{p\min}(k)]^T \in IR^{pN_u}$$

$$\Delta U_{\max}(k) = [\Delta U_{1\max}(k) \quad \ldots \quad \Delta U_{p\max}(k)]^T \in IR^{pN_u}$$

$$\Delta U_{n\min}(k) = [\Delta u_{n\min}(k) \quad \ldots \quad \Delta u_{n\min}(k+N_u-1)]^T \in IR^{N_u}$$

$$\Delta U_{n\max}(k) = [\Delta u_{n\max}(k) \quad \ldots \quad \Delta u_{n\max}(k+N_u-1)]^T \in IR^{N_u}$$

for $n = 1, \ldots, p$, $\Delta u_{n\min}(k)$ and $\Delta u_{n\max}(k)$ represent, respectively, the upper and lower bounds values of the nth control increments.

Then, the optimization problem is formulated as:

$$\Delta U(k) = \min_{\Delta U(k)} \max_{W \in \Omega} J(\Delta U(k), W) \tag{15}$$

Subject to constraints:

$$U_{\min}(k) \le U(k) \le U_{\max}(k)$$

$$\Delta U_{\min}(k) \le \Delta U(k) \le \Delta U_{\max}(k)$$

$$W_{\min} \le W \le W_{\max}$$

$$\Delta U(k) = [\Delta U_1(k), \ \ldots, \ \Delta U_p(k)]^T \in IR^{pN_u}$$

denotes the variables vector of the control increments, where:

$$\Delta U_n(k) = [\Delta u_n(k), \ \ldots, \ \Delta u_n(k+N_u-1)]^T \in IR^{N_u}$$

for $n = 1, \ldots, p$. To apply the minimax problem, let us unifies the inequalities constraints in a matrix form.

By using (3), the nth control signal can be expressed as:

$$\begin{aligned} u_n(k) &= u_n(k-1) + \Delta u_n(k) \\ &\vdots \\ u_n(k+N_u-1) &= u_n(k-1) + \Delta u_n(k) \\ &\quad + \cdots + \Delta u_n(k+N_u-2) + \Delta u_n(k+N_u-1) \end{aligned} \tag{16}$$

Denote:

$$P = \begin{bmatrix} I_p & 0 & 0 & \cdots & 0 \\ I_p & I_p & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ I_p & I_p & \cdots & I_p & I_p \end{bmatrix} \in IR^{pN_u \times pN_u}$$

where I_p and 0 denotes the identity and zero matrices of appropriate dimension.

Therefore, the optimization problem (15) can be expressed as a non-linear programming problem:

$$\min_{\Delta U(k)} \max_{W \in \Omega} J(\Delta U(k), W) \tag{17}$$

Subject to

$$\Delta U_{\min}(k) \le \Delta U(k) \le \Delta U_{\max}(k)$$

$$U_{\min}(k) \le U(k-1) + P\Delta U(k) \le U_{\max}(k)$$

$$W_{\min} \le W \le W_{\max}$$

The problem (17) becomes:

$$\min_{\Delta U(k)} \max_{W \in \Omega} J(\Delta U(k), W) \tag{18}$$

Subject to

$$\alpha \leq \sigma[\Delta U \quad W]^T \leq \beta$$

where

$$\sigma = \begin{bmatrix} I_{pN_u} & P \,|\, 0 \\ 0 & 0 \,|\, I_{mL} \end{bmatrix}^T \in IR^{(2pN_u+mL)\times(pN_u+mL)}$$

$$\alpha = [\Delta U_{\min}(k) \quad U_{\min}(k) - U(k-1) \quad |W_{\min}\ ]^T$$
$$\beta = [\Delta U_{\max}(k) \quad U_{\max}(k) - U(k-1) \quad |W_{\max}\ ]^T$$

$$\alpha \in IR^{(2pN_u+mL)}, \ \beta \in IR^{(2pN_u+mL)}$$

The optimization problem (18) is formulated into two sub-problems. In the first step, an initial input vector is used to compute the maximum of the cost function for all the models within the family of models described by the uncertainty set Ω. So, the first problem is given by:

$$\begin{aligned} P_1 &= \max_{W\in\Omega} J(\Delta U(k), W) \\ &= -\min_{W\in\Omega} -J(\Delta U(k), W) \end{aligned} \tag{19}$$

As a result, the values of the connection weights of the output layer of the neural network are obtained. We will refer to W^*, the solution of the optimization problem (19). In the second step, the unknown variables are the vector composed of (pN_u) future increment control signals. The optimization problem (18) can equivalently be written as an optimization problem of the form:

$$P_2 = \min_{\Delta U(k)} J(\Delta U(k), W^*) \tag{20}$$

The optimal control input increment vector is the solution of the constrained minimization problem (20) that minimizes the worst case of the objective function (2) at time instant k over the prediction horizon N_2.

The algorithm

Based on the non-linear function of the system model and the non-linear relation between the connection weights of the output layer of the neural network and the control values, the optimization problem (19) is non-convex. The gradient method is used to find the minimum solution of the optimization problems. This method is an iterative optimization technique and attempts to find a constrained minimum of a non-linear function. It is based on calculus of the derivative of the objective function to be optimized. We will refer to $J(k)$ as the objective function and to ϕ as the optimized variables. Based on the gradient method, the optimal solution of the problem $\min_{\phi} J(k)$ is computed by:

$$\phi_{j+1} = \phi_j - \beta \frac{\partial J(k)}{\partial \phi_j} \tag{21}$$

where $j=1, \ldots, N_{iter}$. N_{iter} denotes the maximal number used to smooth the optimal value. The gradient algorithm depends on the rate of the steepest gradient β that aims to accelerate the convergence to the optimal solution. In the optimization toolbox of Matlab, we can use the predefined function 'fmincon' to solve the constrained optimization problem. The scheme of the proposed controller based on uncertain non-linear model is depicted in Figure 2.

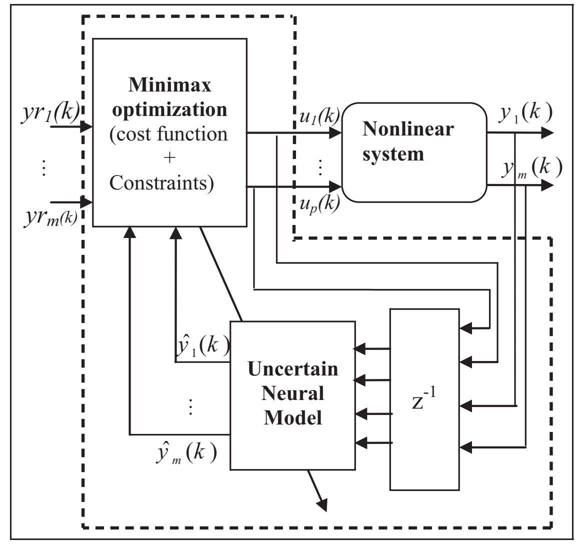

Figure 2. Block diagram of the multi-input–multi-output (MIMO) model predictive control (MPC) controller based on a neural network (NN) model.

The elimination of the steady-state error for the RMPC can be obtained by applying the method presented in Jazayeri et al. (2008). This method is based on the use of disturbance model, which is an iterative learning to reduce the tracking error. The disturbance model is trained by a gradient descent method with adaptive weighting that distinguished external disturbances and model mismatches.

Thereby, the disturbance model is added to the main neural network model. We will refer to w_{ij}^* as the optimal connection weights of the output layer of the neural network by solving the optimization problem (19). Hence, the output model is given by the following equation:

$$\hat{y}_i(k) = NN(w_{ij}^*, X(k)) + d_i(k), \tag{22}$$

for $i=1, \ldots, m$ and $j=1, \ldots, L$, and $d_i(k)$ is the output of the disturbance model, which is described by:

$$d_i(k) = k_{1i}e_i(k) + k_{2i} \tag{23}$$

where $e_i(k)$ is the difference between the output of the neural model and the system output. k_{1i} and k_{2i} are disturbance model weightings, which are adjusted at each sample time.

The rules to adapt the disturbance model parameters are given by Jazayeri et al. (2008), which are based on the proportional integral learning rule:

$$\begin{aligned} k_{1i}(k)_{j+1} &= k_{1i}(k)_j + \eta e_i(k) + k_p(e_i(k) - e_i(k-1)) \\ k_{2i}(k)_{j+1} &= k_{2i}(k)_j + \eta e_i(k)^2 \end{aligned} \tag{24}$$

for $j=1, \ldots, N_{iter}$ and η is the learning rate. Then, the algorithm used to resolve the minimax optimization problem (18) is summarized as follows:

Step 1. Let $k=1$. Set the values of the connection weights from inputs to the hidden layer, the upper and lower bounds weights vectors of the output layer ($W_{\min}, W_{\max}$) of the NN model, the final control time T, the set point trajectory $yr_1(k), \ldots, yr_m(k)$ the prediction and control horizons N_2, N_u, and the control weighting factor λ.
Step 2. Find the connection weights of the output layer of the neural network, which are the solution of the optimization problem (19).
Step 3. Use the solution of the optimization problem (19) for computing the control sequence Δu_n, for $n=1, \ldots, p$, by solving the next optimization problem (20); then, the first control of the calculated sequence is applied to the system.
Step 4. If $k<T$, $k=k+1$, take the new measurements and go to step 2, otherwise end.

Simulation results

The objective of this section is to examine the effectiveness of the proposed robust controller for MIMO non-linear systems based on uncertain FNNs. The process has two inputs and two outputs and it is represented by the discrete equations (25a) and (25b) (Petlenkov, 2007; Song and Li, 2006).

$$y_1(k) = \frac{a_1 y_1(k-1) y_1(k-2)}{1 + a_2 y_1(k-1)^2 + a_3 y_2(k-2)^2} + a_4 u_1(k-2) + a_5 u_1(k-1) + a_6 u_2(k-2) \quad (25a)$$

$$y_2(k) = \frac{b_1 y_2(k-1) sin(y_2(k-2))}{1 + b_2 y_2(k-1)^2 + b_3 y_1(k-2)^2} + b_4 u_2(k-2) + b_5 u_2(k-1) + b_6 u_1(k-2) \quad (25b)$$

Assume that the parameter variations of the system a_i and b_i are known along of trajectory and three situations of the dynamic behaviour of the full system are defined in Table 1 as follows.

Table 1. Parameters values of the full system.

Parameter values	Nominal	Situation 1 (±10%)	Situation 2 (±20%)
a_1	0.7	0.77	0.56
a_2	1	0.9	1.2
a_3	1	1.1	0.8
a_4	0.3	0.33	0.36
a_5	1	0.9	1.2
a_6	0.2	0.22	0.16
b_1	0.5	0.55	0.4
b_2	1	1.1	1.2
b_3	1	0.9	1.2
b_4	0.5	0.55	0.4
b_5	1	1.1	0.8
b_6	0.2	0.18	0.24

Non-linear system modelling

In the modelling stage, we have considered three situations. In the first one, the system behaviour is described by the nominal values parameters. In the second and third situations, the system's behaviour will be changed and 10% and 20% level variations about the nominal parameters values are introduced, respectively. Various NN models with different sizes are trained in order to find the best one with the minimum value of the performance criterion.

The control inputs sequences u_1 and u_2 for the training of the model are formed of pulses of positives random amplitude in the interval [0; 1]. The finite length of the control inputs sequences is $N_m=300$. So, the final structure of the neural network model is formed by 10 neurons in the hidden layer with tangent hyperbolic as activation function and two neurons in the output layer. The input elements to the neural network model are defined by the following vector:

$$X(k) = \{u_1(k-1), u_1(k-2), u_2(k-1), u_2(k-2), y_1(k-1), y_1(k-2), y_2(k-1), y_2(k-2), \}$$

The back-propagation algorithm is used to estimate the weights of the neural networks model and a reduced value of sum square error E on the training sequences is assumed.

$$E = \frac{1}{2}\sum_{i=1}^{2}\sum_{k=1}^{300}(y_i(k) - \hat{y}_i(k)) = 2.67\ 10^{-2} \quad (26)$$

Thereby, the same size was adopted for the two other neural models and we assume that the hidden layer weights are fixed but only the parameters of the outputs layers are determined. The MIMO neural network model is obtained for each situation of the multivariable system.

The real process can be characterized by a single neural network model formed by a one hidden layer. The proposed model is determined by the use of fixed weights in the hidden layer and by structuring the uncertain connection weights of the output layer. The validation of the NN models are captured by using a random control inputs sequences.

Non-linear system control

In order to control the non-linear system, which has known parameter variations, an uncertain non-linear model is used and the control design shown in Figure 2 can be implemented. The efficiency of the robust predictive control method is related to the quality of the model, which is used to approximate the behaviour of the real system. The robustness of the centralized MPC based on the FNN model is illustrated with the simulated process. The set-point sequence consists of pulses with amplitudes in the interval [0; 1.5]. The finite length of the sequence is $S=180$. The system behaviour is described by the nominal model for the 65 first sample times; thereafter, the parameters of the nominal system's behaviour will be varied on the wide range of time, which are shown in Table 1. Assume that the time of system behaviour change is

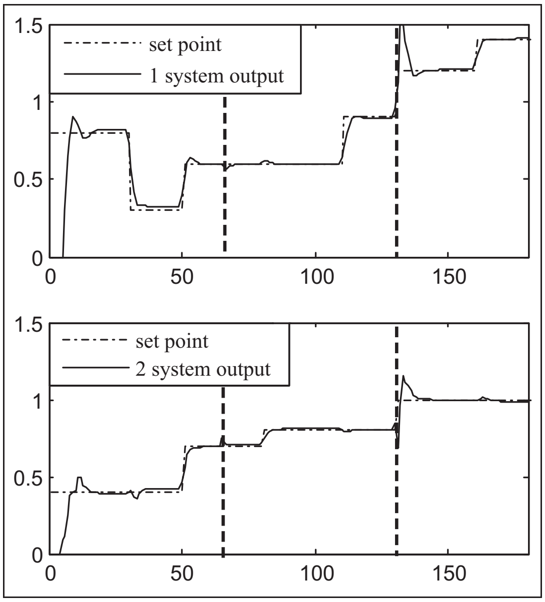

Figure 3. Evolutions of the system outputs y_1 and subject to an acceptable control action.

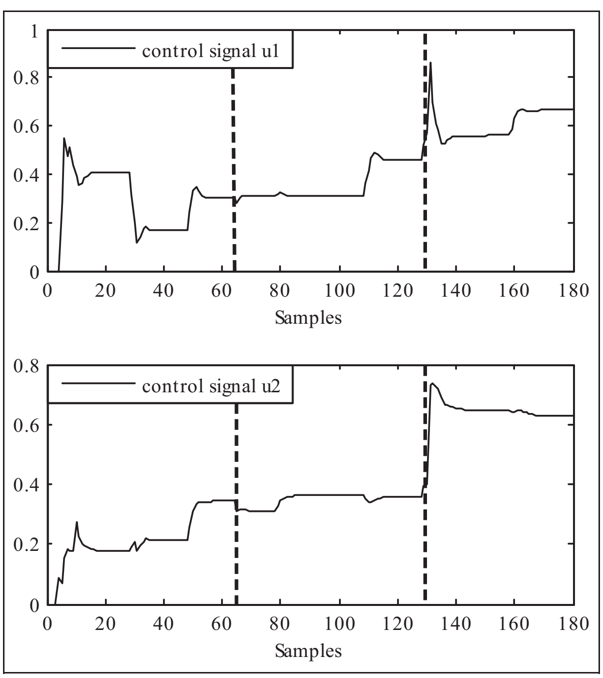

Figure 4. Evolutions of the control signals u_1 and u_2 without constraints.

unknown *a priori*. Assume that our information about the weights of the output layer w_{ij}, for $i = 1$, 2 and $j = 1, \ldots 10$, is that these parameters are given in a compact set $W \in [W_{\min}, \quad W_{\max}] \in IR^{20}$, in which $W_{\min}$ and $W_{\max}$ are, respectively, the lower and upper bounds of the connection weights of the output layer w_{ij} of the NN.

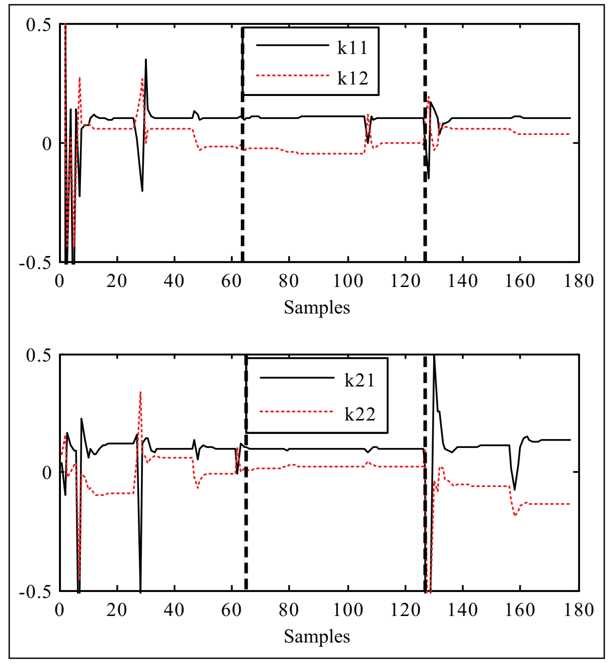

Figure 5. Evolutions of the weights of the disturbance model corresponding to the output system y_1 and y_2 without constraints.

In order to take this knowledge about the uncertainty W into account, the robust predictive control is applied. Hence, the optimal control signals $u_1(k)$ and $u_2(k)$ are computed by minimizing the worst possible value of the objective function $J(\Delta u_n, W)$ for $n = 1, 2$. This kind of problem is bi-level: firstly, the maximization problem with respect to the parameter uncertainties weights of the output layer W is solved in order to find the worst model parameters and secondly, the optimum sequence of control signals $u_1(k)$ and $u_2(k)$ are computed by solving the problem (20). The centralized predictive controller parameter values used in the control system are given by: $N_2 = 4$, $N_u = 1$ and $\lambda = 1$.

Figure 3 shows the reference trajectory and the closed-loop responses obtained with the MIMO predictive controller. We can observe that the output system tracks the reference trajectory.

Figure 4 depicts the corresponding control signals u_1 and u_2. It can be noted that we have improved the tracking response of the closed-loop system and the proposed neural controller satisfied good performances in controlling multivariable system. Figure 5 shows that by adjusting the weighting of the disturbance model, we can overcome the steady error tracking and the satisfactory output tracking subject to an acceptable control action.

In addition, the ability of MPC to handle constraints on controls is also tested in this example. Therefore, the constraints imposed on the control signals of the multivariable systems are given by:

$$\begin{cases} 0.15 \le u_1(k) \le 0.42 \\ 0 \le u_2(k) \le 1 \end{cases} \text{for } k < 65 \tag{27a}$$

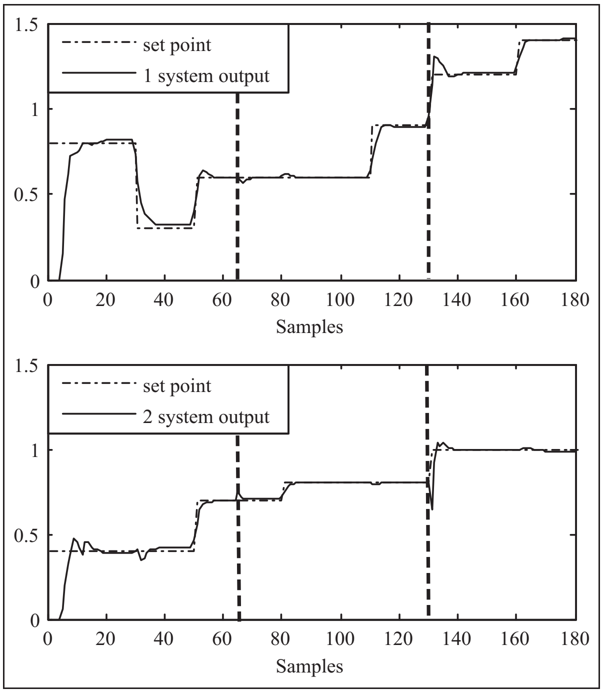

Figure 6. Evolutions of the system outputs y_1 and y_2 with constraints.

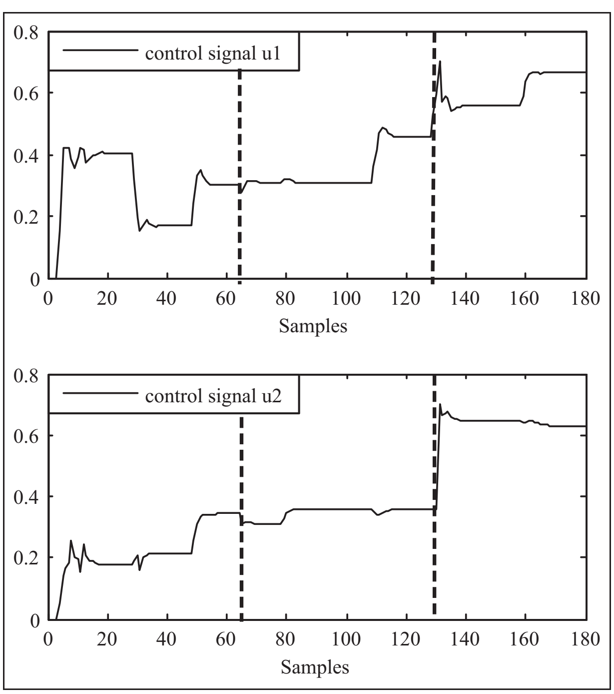

Figure 7. Evolutions of the control signals u_1 and u_2 with constraints.

$$\begin{cases} 0.25 \le u_1(k) \le 1 \\ 0.25 \le u_2(k) \le 0.36 \end{cases} \text{for } 65 \le k < 130 \tag{27b}$$

$$\begin{cases} 0 \le u_1(k) \le 0.7 \\ 0 \le u_2(k) \le 0.7 \end{cases} \text{for } k \ge 130 \tag{27c}$$

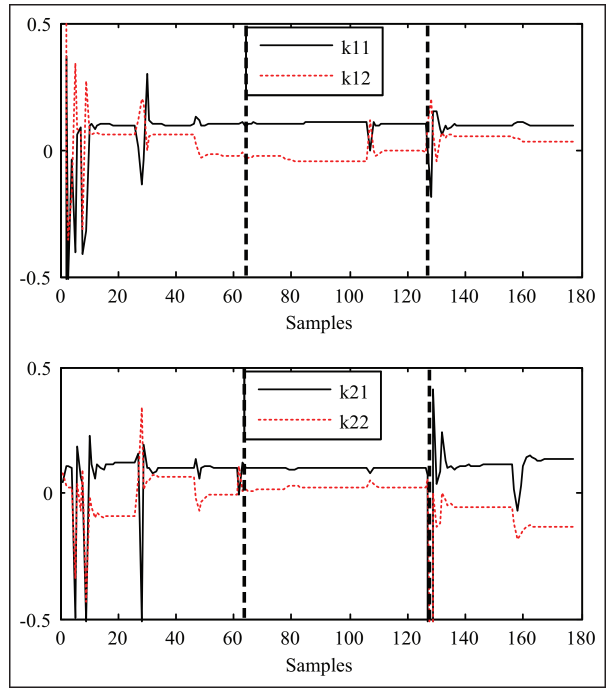

Figure 8. Evolutions of the weights of the disturbance model corresponding to the output system y_1 and y_2 with constraints.

The simulation results in the presence of the uncertainties of the neural networks model and the control signals constraints are depicted in Figures 6 and 7. From these simulation results, it is shown that the system outputs successfully track the set-points. On the other hand, it is shown that the control signals constraints slowly lead to a closed-loop system.

Moreover, Figure 7 illustrates that the imposed control signals constraints are satisfied. Figure 8 shows that by adjusting the weighting of the disturbance model, we can overcome the steady error tracking and the output tracking subject to an acceptable control action. The smooth trends of the system outputs indicate good performance with the proposed method. The proposed controller based on neural networks model with uncertain parameters in the output layer presents a good tracking performance of the closed-loop system.

Conclusion

A centralized predictive controller based on an uncertain neural network for multivariable systems is developed in this paper. Thanks to the ability to approximate the dynamic behaviour of the system, only a single FNN model is used at each operating point of the MIMO system. The system nonlinear model is only determined by parametric uncertain weights of the output layer of the neural network model, whereas the connection weights from inputs to the hidden neurons are fixed. The control law is computed by solving a minimax optimization problem at each sampling time. The obtained simulation results have illustrated the efficiency of the proposed controller.

Conflict of interest

The authors declare that there is no conflict of interest.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

Alessio A and Bemporad A (2009) A survey on explicit model predictive control. *Lecture Notes in Control and Information Sciences* 384: 345–369.

Behera L and Kar I (2010) *Intelligent Systems and Control Principles and Applications*. New York: Oxford University Press.

Bemporad A, Borrelli F and Morari M (2003) Min–max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on Automatic Control* 48(9): 1600–1606.

Ben-Tal A, El-Ghaoui L and Nemirovski A (2009) Robust optimization. Princeton, NJ: Princeton University Press.

Bouzouita B, Bouani F and Ksouri M (2007) Robust MPC for nonlinear multivariable systems. *Proceedings of the 15th Mediterranean Conference on Control and Automation*, Crete, Greece.

Cheng L, Hou Z and Tan M (2007) Constrained multi-variable generalized predictive control using a dual neural network. *Neural Computing and Applications* 16(6): 505–512.

Fei M, Zhang J, Hu H, et al. (2006) A novel linear recurrent neural network for multivariable system identification. *Transactions of the Institute of Measurement and Control* 28(3): 229–242.

Hernandez E and Arkun Y (1993) Control of nonlinear systems using polynomial ARMA models. *AIChE Journal* 39(3): 446–460.

Ho YK, Mjalli FS and Yeoh HK (2012) Centralized vs decentralized adaptive generalized predictive control of a biodiesel reactor. *Asia Pacific Journal of Chemical Engineering* 8(1): 137–143.

Holkar KS and Waghmare LM (2010) An overview of model predictive control. *International Journal of Control and Automation* 5, 47–64.

Hornik K, Stinchcombe M and White H (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* 2(5): 359–366.

Jazayeri A, Fatehi A, Sadjadian H, et al. (2008) Disturbance rejection in neural network model predictive control. *Proceeding of the 17th IFAC World Congress* Seoul, Korea.

Kheriji A, Bouani F and Ksouri M (2011) A GGP approach to solve nonconvex min-max predictive controller for a class of constrained MIMO systems described by state space models. *International Journal of Control, Automation and Systems* 9(3): 452–460.

Kotta Ü, Chowdhury FN and Nõmm S (2006) On realizability of neural-networks-based input–output models in the state space form. *Automatica* 42(7): 1211–1216.

Lawrynczuk M and Tatjewski P (2010) Nonlinear predictive control based on neural multi-models. *International Journal of Applied Mathematics and Computer Science* 20(1): 7–21.

Lu CH, Tsai CC, Liu CM, et al. (2010) Neural network based predictive controller design: An application to temperature control of a plastic injection molding process. *Asian Journal of Control* 12(6): 680–691.

Lofberg J (2003) Minimax approaches to robust model predictive control. *Linkoping Studies in Sciences and Technology* 812.

Magni L, Raimondo DM and Allgwer F (2009) Nonlinear model predictive control: Towards new challenging applications. *Lecture Notes in Control and Information Sciences* 384.

Morari M (2009) *Nonlinear Model Predictive Control: Towards New Challenging Applications*. Berlin: Springer.

Mnasser A, Bouani F and Ksouri M (2013) Neural networks predictive control using an adaptive control rate. *International Conference on Control, Decision and Information Technologies*, Hammamet, Tunisia.

M'Salhi F, Ben Abdennour R and Ksouri M (2001) Identification and predictive control of a nonlinear process using a parametric Volterra model. *International Journal of Computational Engineering Science* 2(4): 633–651.

Narendra KS and Parthasarathy K (1990) Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Transactions on Neural Networks* 1(1): 4–27.

Pan Y and Wang J (2012) Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks. *IEEE Transactions on Industrial Electronics* 59(8): 3089–3101.

Petlenkov E (2007) NN-ANARX structure based dynamic output feedback linearization for control of nonlinear MIMO systems. *Proceedings of the 15th Mediterranean Conference on Control and Automation*, Athens, Greece.

Raimondo DM, Limon D, Lazar M, et al. (2009) Min-max model predictive control of nonlinear systems: A unifying overview on stability. *European Journal of Control* 15(1): 5–21.

Raimondo DM and Magni L (2006) A robust model predictive control algorithm for nonlinear systems with low computational burden. *IFAC Workshop on Nonlinear Model Predictive Control for Fast Systems*, Grenoble, France.

Song F and Li P (2006) MIMO Decoupling Control Based on Support Vector Machines αth-order Inversion. *Proceeding of the 6th World Congress on Intelligent Control and Automation*, Dalian, China.

Wang D, Bai Y and Zhao J (2012) Robot manipulator calibration using neural network and a camera based measurement system. *Transactions of the Institute of Measurement and Control* 34(1): 105–121

Yan Z and Wang J (2014) Robust MPC of nonlinear systems with unmodeled dynamics and bounded uncertainties based on neural networks. *IEEE Transactions on Neural Network and Learning Systems* 25(3): 457–469.

Zhang R, Lan Y, Huang GB, et al. (2012) Universal approximation of extreme learning machine with adaptive growth of hidden nodes. *IEEE Transactions on Neural Network and Learning Systems* 23(2): 365–371.