Predictive Control Design Based on Neural Model of a Non-linear System

Anna Jadlovská, Nikola Kabakov, Ján Sarnovský

Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Slovak Republic
anna.jadlovska@tuke.sk, nikola.kabakov@gmail.com, jan.sarnovsky@tuke.sk

Abstract: This paper provides two approaches for design of Generalized Predictive Control (GPC) algorithm for a non-linear dynamic system. In the classical approach of the GPC strategy the recursive method of least square is considered for calculating of the linearized model parameters from the known analytic description of a particular non-linear system. The other purpose of this paper is to show an intelligent approach in which a feed-forward neural network (Multi Layer Perceptron – MLP) is used for modeling the same non-linear system within the frames of the predictive control. The possibility of on-line estimation of an actual parameters from the off-line trained neural model of the non-linear system by means method of the instantaneous linearization in each sample point is considered in GPC algorithm design. The validity of classical and neural GPC strategy is tested by computer simulations in Matlab/Simulink language using architecture of S-functions of the library PredicLib.

Keywords: non-linear dynamic system, neural model, parametric estimation, predictive control algorithm

1 Introduction

Model Predictive Control (MPC) is one of the most wide spread advanced control techniques for dynamic systems in the industry. The main idea of MPC algorithms is to solve an optimization problem in order to find the control vector trajectory that optimizes the cost function over a future prediction horizon [2], [3].

This paper provides two approaches for the design of Generalized Predictive Control (GPC) algorithm for non-linear systems. In the classical approach of GPC strategy (part 2) the recursive least square method (RLSM) is considered for the calculation of the linearized model parameters from the known analytic description of the dynamic system [8]. In the intelligent approach is a feed-forward neural network (Multi Layer Perceptron – MLP) used for modeling and the estimation of the actual parameters of the non-linear system which are considered in GPC design (part 4-6). The neural model of non-linear system is typically trained in advance, but the GPC controller is designed on-line using the parameter estimation from the neural model. The main idea of this paper is to show how on-line estimation of the actual parameters from off-line trained neural model using the gain matrix is applied in the GPC algorithm [5]. The neural model of non-linear system is linearized by means method of the instantaneous linearization in each sample point and the result of this linearization technique – the estimated parameters from neural ARX model (NARX) of the system are used for design of the GPC algorithm (part 6). The practical simulations by the language Matlab/Simulink, Neural Toolbox and the library PredicLib [6] of this paper illustrate that the classical GPC and the neural GPC strategies using linearization technique can be used for predictive control of a particular non-linear dynamic system.

2 GPC Algorithm - Classical Approach

The calculating of the cost function for optimal control for the *k*-th step can be considered as

$$J(k)=\sum_{j=N_1}^{N_2} e^2(k+j/k)+\sum_{j=1}^{N_u}\lambda\Delta u^2(k+j/k) \tag{1}$$

where for the predictive horizons holds $N_1 \geq 1$, $N_2 \geq N_1, 1 \leq N_u \leq N_2$, λ is the positive weight coefficient and $e(k+j/k)=\hat{y}(k+j/k)-r(k+j/k)$ is the prediction error, where $\hat{y}(k+j/k)$ is the predicted system output value and $r(k+j/k)$ is the system output required value. By [8] the most SISO (Single-Input/Single-Output) systems when is considered the operation around a particular set-point and after linearization can be described by the linear discrete *AutoRegresive model with eXternal input* (ARX). Using the modification of ARX model we can get CARIMA (*Controlled AutoRegresive Integrate Moving Average*) model which is used in GPC algorithm [2] and is described by eq. (2)

$$A(q^{-1})y(k)=B(q^{-1})u(k-1)+\frac{C(q^{-1})}{\mathit{\Delta}}\xi(k) \tag{2}$$

where $A(q^{-1})$, $B(q^{-1})$, $C(q^{-1})$ are the polynomials of the delay operator q^{-1}:

$$A(q^{-1})=1+a_1q^{-1}+a_2q^{-2}+\ldots+a_{na}q^{-na}, B(q^{-1})=b_0+b_1q^{-1}+b_2q^{-2}+\ldots+b_{nb}q^{-nb},$$
$$C(q^{-1})=1+c_1q^{-1}+c_2q^{-2}+\ldots+c_{nc}q^{-nc}$$

In the equation (2) $\xi(k)$ is a white noise with zero mean and $\mathit{\Delta}=1-q^{-1}$. According [2] the future output value of the system is given by (3)

$$\hat{y}(k+j/k) = G_j(q^{-1})\varDelta u(k+j-1) + \varGamma_j(q^{-1})\varDelta u(k-1) + F_j(q^{-1})y(k) \tag{3}$$

In the equation (3) the polynomials $G_j(q^{-1})$, $\varGamma_j(q^{-1})$ and $F_j(q^{-1})$ are calculated by solving of Diophantine equations:

$$C(q^{-1}) = E_j(q^{-1})A(q^{-1})\varDelta + q^{-j}F_j(q^{-1})$$

$$B(q^{-1})E_j(q^{-1}) = G_j(q^{-1})C(q^{-1}) + q^{-j}\varGamma_j(q^{-1})$$

The predictor (3) can be written as

$$\hat{\boldsymbol{y}} = \boldsymbol{G}\boldsymbol{\varDelta u} + \boldsymbol{F}(q^{-1})y(k) + \boldsymbol{\varGamma}(q^{-1})\varDelta u(k-1) \tag{4}$$

where

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}(k+N_1/k) \\ \hat{y}(k+N_1+1/k) \\ \vdots \\ \hat{y}(k+N_2/k) \end{bmatrix}, \quad \boldsymbol{\varDelta u} = \begin{bmatrix} \varDelta u(k) \\ \varDelta u(k+1) \\ \vdots \\ \varDelta u(k+N_u\text{-}1) \end{bmatrix},$$

$$\boldsymbol{G} = \begin{bmatrix} g_{N_1} & g_{N_1-1} & \cdots & g_{N_1-N_u+1} \\ g_{N_1+1} & g_{N_1} & \cdots & g_{N_1-N_u+2} \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_2} & g_{N_2-1} & \cdots & g_{N_2-N_u+1} \end{bmatrix}, \ \boldsymbol{\varGamma}(q^{-1}) = \begin{bmatrix} \varGamma_{N_1}(q^{-1}) \\ \varGamma_{N_1+1}(q^{-1}) \\ \vdots \\ \varGamma_{N_2}(q^{-1}) \end{bmatrix}, \ \boldsymbol{F}(q^{-1}) = \begin{bmatrix} F_{N_1}(q^{-1}) \\ F_{N_1+1}(q^{-1}) \\ \vdots \\ F_{N_2}(q^{-1}) \end{bmatrix}.$$

Because two last items of (4) depends only on the previous states, we can include those into one item $\boldsymbol{f}$, then the equation of the predictor is $\hat{\boldsymbol{y}} = \boldsymbol{G\varDelta u} + \boldsymbol{f}$. The cost function (1) can be written in the matrix form for the computing of an optimal control

$$J = (\boldsymbol{G\varDelta u} + \boldsymbol{f} - \boldsymbol{r})^T(\boldsymbol{G\varDelta u} + \boldsymbol{f} - \boldsymbol{r}) + \lambda\boldsymbol{\varDelta u}^T\boldsymbol{\varDelta u} \tag{5}$$

where $\boldsymbol{r} = \left[r(k+N_1), r(k+N_1+1), ..., r(k+N_2)\right]^T$ is the reference trajectory and λ is the weight coefficient.

The equation (5) can be written as

$$J = \frac{1}{2}\boldsymbol{\varDelta u}^T\boldsymbol{H\varDelta u} + \boldsymbol{b}^T\boldsymbol{\varDelta u} + \boldsymbol{f}_0 \tag{6}$$

where $\boldsymbol{H} = 2(\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I})$, $\boldsymbol{b}^T = 2(\boldsymbol{f} - \boldsymbol{r})^T\boldsymbol{G}$, $\boldsymbol{f}_0 = (\boldsymbol{f} - \boldsymbol{r})^T(\boldsymbol{f} - \boldsymbol{r})$.

The minimum of the cost function J can be found by making gradient of J equal to zero, which leads to

$$\boldsymbol{\varDelta u} = -\boldsymbol{H}^{-1}\boldsymbol{b} = -(\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I})^{-1}\boldsymbol{G}^T(\boldsymbol{f} - \boldsymbol{r})\,\boldsymbol{.} \tag{7}$$

The result of this equation is the trajectory consisting from the increments of the control signal and the first of them is applied on the system and is given by:

$\Delta u(k) = \boldsymbol{K}(\boldsymbol{r} - \boldsymbol{f})$, where $\boldsymbol{K}$ is the first row of the matrix $(\boldsymbol{G}^T\boldsymbol{G} + \lambda \boldsymbol{I})^{-1}\boldsymbol{G}^T$, so an actual controller output value is $u(k) = u(k-1) + \boldsymbol{K}(\boldsymbol{r} - \boldsymbol{f})$.

For the systems with the constrains on the controller output value, on the controller increment output value or on the system output value, the vector $\Delta \boldsymbol{u}$ is calculated by function ***quadprog*** of Optimization Toolbox of the language Matlab

$\Delta \boldsymbol{u}$ = ***quadprog*** ($\boldsymbol{H}$, $\boldsymbol{b}^{\mathrm{T}}$, $\boldsymbol{L}_{CON}$, $\boldsymbol{v}$, $\boldsymbol{U}_{MIN}$, $\boldsymbol{U}_{MAX}$), where $\boldsymbol{U}_{MIN} \leq \Delta \boldsymbol{u} \leq \boldsymbol{U}_{MAX}$, $\boldsymbol{L}_{CON}\, \Delta \boldsymbol{u} \leq \boldsymbol{v}$.

The vectors $\boldsymbol{U}_{MIN}$ and $\boldsymbol{U}_{MAX}$ are the column vectors those elements are minimal and maximal values of $\Delta \boldsymbol{u}(k)$. With using the matrix $\boldsymbol{L}_{CON}$ and the vector $\boldsymbol{v}$ can be defined the system of an inequalities which insures that constrain conditions will be satisfied [6].

$$L_{CON1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}, \; v_1 = \left[\left(u_{MAX} - u(k-1)\right) . \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right]$$

$$L_{CON2} = \begin{bmatrix} -1 & 0 & \cdots & 0 \\ -1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & -1 \end{bmatrix}, \; v_2 = \left[\left(u_{MIN} - u(k-1)\right) . \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix} \right]$$

$$L_{CON3} = G, \; v_3 = \left[y_{MAX} . \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} - \boldsymbol{F}(q^{-1}).y(k) - \boldsymbol{\Gamma}(q^{-1}).\Delta u(k-1) \right]$$

$$L_{CON4} = -G, \; v_4 = \left[y_{MIN} \cdot \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix} + \boldsymbol{F}(q^{-1}).y(k) + \boldsymbol{\Gamma}(q^{-1}).\Delta u(k-1) \right]$$

$$L_{CON} = \begin{bmatrix} L_{CON1} \\ L_{CON2} \\ L_{CON3} \\ L_{CON4} \end{bmatrix}, \; v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

In the programmable environment Matlab/Simulink was designed GPC algorithm for the dynamic non-linear SISO system using S-functions.

Algorithm GPC for the calculation of the control signal value for the *k* - th step:

1 reading of the polynomials $A(q^{-1})$, $B(q^{-1})$, $C(q^{-1})$ of the linearized discrete model (2) of the non-linear system, the reference vector trajectory $\boldsymbol{r}(k)$ and the output of the system $\boldsymbol{y}(k)$,

2 cyclical calculating of the polynomials $G_j(q^{-1})$, Γ_j (q^{-1}) and $F(q^{-1})$ by solving Diophantine equations for $j = 1, 2, .., N_2$,

3 creating of the matrixes $\boldsymbol{G}$, $\boldsymbol{\Gamma}$and $\boldsymbol{F}$,

4 if is required the constrain for the values of $u(k)$, $\Delta u(k)$ or $y(k)$, then continue by the step 8,

5 calculating of the feedback gain of the control vector $\boldsymbol{K}$,

6 calculating of the controller output increment $\Delta u(k) = \boldsymbol{K}\,(\boldsymbol{w} - \boldsymbol{f})$,

7 continue by the step 12,

8 creating of the matrix $\boldsymbol{L}_{CON}$ and vectors $\boldsymbol{v}$, $\boldsymbol{U}_{MIN}$, $\boldsymbol{U}_{MAX}$,

9 calculating of the matrixes $\boldsymbol{H}$ and $\boldsymbol{b}^{\mathrm{T}}$,

10 $\boldsymbol{\Delta u} = quadprog(\boldsymbol{H}, \boldsymbol{b}^{\mathrm{T}}, \boldsymbol{L}_{CON}, \boldsymbol{v}, \boldsymbol{U}_{MIN}, \boldsymbol{U}_{MAX})$,

11 $\Delta u(k)$ is the first element of the vector $\boldsymbol{\Delta u}$,

12 $u(k) = u(k\text{-}1) + \Delta u(k)$, $k = k+1$ and continue by the step 1.

3 Verification of GPC Algorithm for a Non-linear System by Simulation

The simulation model of the non-linear system consists of two tanks, one cylindrical and the other spherical with the free outflow. The inflow to the first tank is controlled by the relative open of the valve V_1. The outflow from the first tank is an inflow to the second one.

The physical variables and the parameters of the hydraulic system which is described in Fig. 1 are:

$\boldsymbol{h_1(t)}$ [m] - the water level in cylindrical tank,

$\boldsymbol{h_{1MAX}(t)}$ [m] - the max water level in cylindrical tank,

$\boldsymbol{h_2(t)}$ [m] - the water level in the spherical tank,

$\boldsymbol{h_{2MAX}(t)}$ [m] - the max water level in the spherical tank,

$\boldsymbol{M_1\,(t)}$ [kg/s] - the mass inflow to the cylindrical tank,

$\boldsymbol{M_2\,(t)}$ [kg/s] - the mass inflow to the spherical tank,

***M*₃ (*t*)** [kg/s] - the mass outflow from the spherical tank,

***M*_Z1 (*t*)** [kg/s] - the disturbance mass inflow to the cylindrical tank,

***M*_Z2(*t*)** [kg/s] - the disturbance mass inflow to the spherical tank,

***u*₁(*t*)** - the rise of an input outlet of the cylindrical tank,

S [m^2] - the tank's bottom area,

***S*₁** [m^2] - the area of the outflow of the cylindrical tank,

R [m] - the diameter of the spherical tank, ***S*₂** [m^2] is the outflow space of spherical tank.

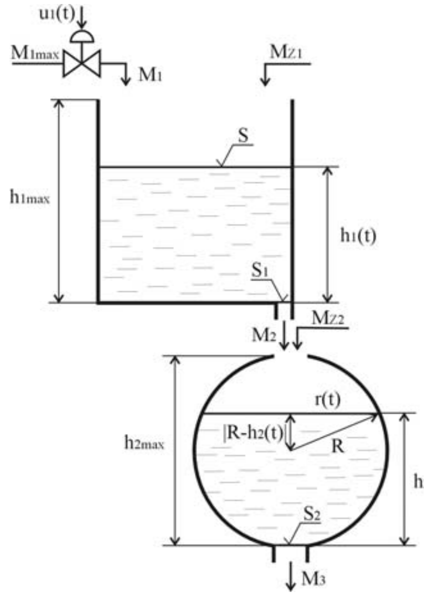

Figure 1
Model of the hydraulic system

For changing the levels $h_1(t)$ and $h_2(t)$ according to [6] it holds that

$$\begin{aligned} \rho S\frac{dh_1(t)}{dt} &= M_1(t)-M_2(t)+M_{Z1}(t) \\ \rho\pi(2Rh_2(t)-h_2^2(t))\frac{dh_2(t)}{dt} &= M_2(t)-M_3(t)+M_{Z2}(t) \end{aligned} \quad (8)$$

The mass flows $M_1(t)$, $M_2(t)$ and $M_3(t)$ can be described by the equations (9), where k_{V1} is the constructing constant of the input outlet of the cylindrical tank and $f(u_1(t))$ is characteristic function of the valve V_1

$$M_1(t)=k_{V1}f(u_1(t))M_{1MAX},\quad M_2(t)=\rho S_1\sqrt{2gh_1(t)},\quad M_3(t)=\rho S_2\sqrt{2gh_2(t)} \quad (9)$$

The mass inflow $M_1(t)$ depends on the relative open value of $u_1(t) \in \langle 0,1 \rangle$ and on the value of maximal mass inflow M_{1MAX}. Let the function $f(u_1(t))$ be linear. Then the equation for the mass inflow can be written as

$$M_1(t) = k_{V1} u_1(t) M_{1MAX} \tag{10}$$

After inducting equations (9) and (10) to the equations (8) we obtain non-linear differential equations (11) describing dynamics of the changing levels in the simulation model - two tanks without interaction

$$\begin{aligned} \frac{dh_1(t)}{dt} &= \frac{k_{V1} u_1(t) M_{1MAX} - \rho S_1 \sqrt{2gh_1(t)} + M_{Z1}(t)}{\rho S} \\ \frac{dh_2(t)}{dt} &= \frac{\rho S_1 \sqrt{2gh_1(t)} - \rho S_2 \sqrt{2gh_2(t)} + M_{Z2}(t)}{\rho \pi (2Rh_2(t) - h_2^2(t))} \end{aligned} \tag{11}$$

By expansion to Taylor series for the set-point SP = $[u_{10,}\ h_{10},\ h_{20}]$ we obtain the linearized model, which can be written by Laplace transformation as transfer functions $F_{H_2/U_1}(s)$ and then using Z-transformation can be obtained the discrete transfer function of the dynamic system $F(z) = B(z)/A(z)$.

For verification of designed GPC algorithm was used the simulation language Matlab/Simulink. The functional block of GPC controller is included into the library PredicLib [6]. The parameters of the simulation model of the hydraulic system are $M_{1MAX} = 500$kg/s, $h_{1MAX} = 6$m, $S = 7.07$m^2, $S_1 = 0.0314$m^2, $R = 1.5$m, $S_2 = 0.0314$ m^2 and $\rho = 1000$ kg/m^3.

For the calculation of the linearized model parameters the recursive method of the least square was used for the set-point SP that corresponds to the actual value of the system output $h_2(t)$. After conversion of the discretized transfer function to time area operator z is substituted with operator q. By this way the polynomials $A(q^{-1})$ and $B(q^{-1})$ of the discrete CARIMA model (2) can be acquired. The polynomial $C(q^{-1})$ is time invariant and is chosen to be equal to one. The orders of discrete model´s polynomials are $na = 2$, $nb = 2$ and $nc = 0$.

The prediction horizon for the output of the system is chosen on ten steps forward and for the control signal on five steps, so $N_1 = 1$, $N_2 = 10$, $N_u = 5$. The sample period is $T_{VZ} = 10$s.

Next is necessary to consider the constrains for the control signal value and the system output value because $u_1(t) \in \langle 0,1 \rangle$ and $h_2(t) \in \langle 0,3 \rangle$. The weight coefficient for the control signal value increment $\Delta u(t)$ is λ=0.1.

The control scheme of the simulation in an environment Matlab/Simulink is described in Fig. 2.

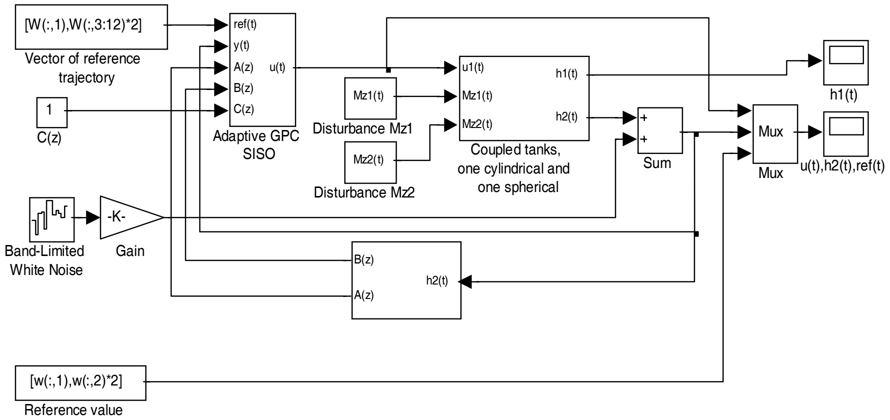

Figure 2
The simulation predictive control scheme – classical approach

On the output of the system have effect disturbance in the form of the white noise that simulates the measurement error. Also the disturbance mass inflows M_{Z1} and M_{Z2} act on the system.

The disturbance mass inflow to the cylindrical tank has value M_{Z1} = 50kg.s^{-1} and acts on in time from 2500s to 3500s. The disturbance mass inflow to the spherical tank have value M_{Z2} = 20kg.s^{-1} and acts on in time from 6500s to 7500s.

The result of tracking of the reference trajectory *ref*(*t*) by the system's output $h_2(t)$ using GPC algorithm is on Fig. 3. The output of the GPC controller – the optimal control signal $u_1(t)$ is described in Fig. 4.

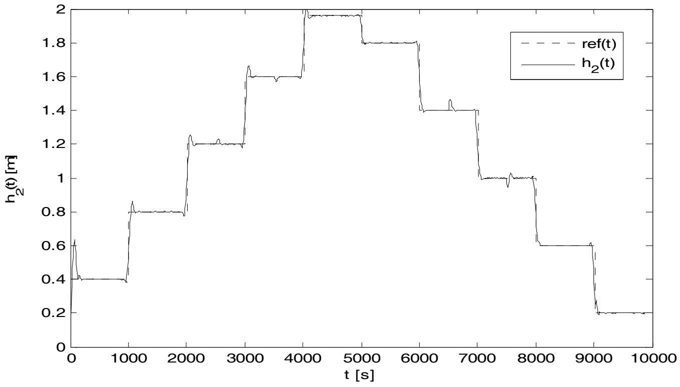

Figure 3
Tracking the reference trajectory *ref*(*t*) by the output of the non-linear system $h_2(t)$

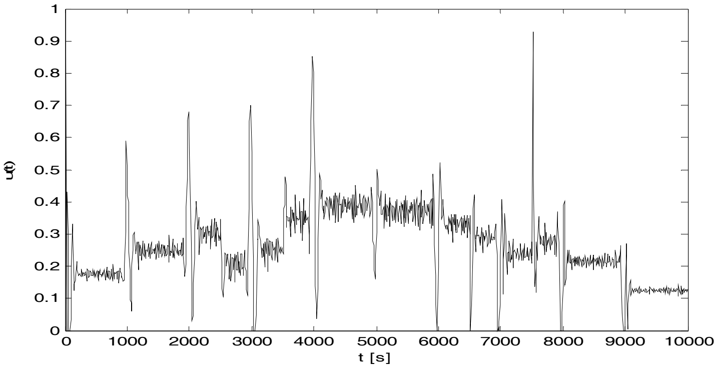

Figure 4

The GPC controller output – control signal $u_1(t)$

4 Predictive Control Structure Using Parameter Estimation from a Neural Model

Next we will consider about the GPC algorithm design for a non-linear system (hydraulic two tanks system) using the neural model of NARX structure [2], [7]. The predictive control scheme using the estimation the system parameters from off-line trained neural NARX model, which are applied in the algorithm of GPC, is illustrated on Fig. 5.

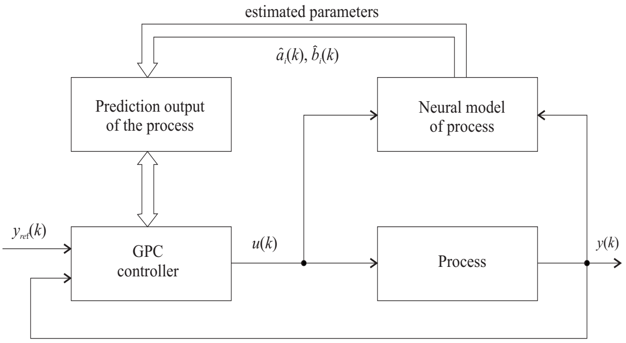

Figure 5

The predictive control scheme based on the neural model - intelligent approach

The loop consists of the controlled non-linear hydraulic system described by the equations (11), the neural process model of NARX structure and GPC controller. Because GPC algorithm needs parameters of a linear model of the dynamic system, in this paper a linear model is extracted from the non-linear neural NARX model by calculation of the gain matrix (part 5). The estimated parameters of the dynamic system from off-line trained neural model are applied for the calculation of the predictor by algebraic theory which is used in GPC algorithm [5], [6]. The optimal predictive controller output vector Δu_{opt} is determined by minimization of the cost function (1). The first term in the cost function refers to the square variation of the predicted system output from the desired reference trajectory, while the second term is added in order to limit the controller output, greater λ yields less active the controller output. The first element of the calculated optimal controller output vector is directed to the hydraulic system input. The remaining vector elements are not utilized and the entire procedure is repeated at the time $t=(k+1)T_{VZ}$, (the principle of receding horizon) [3]. The first prediction horizon N_1 is usually chosen to be 1. The choice of the second prediction horizon N_2 and the control horizon N_u are: N_2 is usually chosen as to cover the most of the control system's transient, while N_u, which denotes the significance of the future controller outputs shouldn't be greater then $N_2/2$.

5 Neural Model of a Non-linear Dynamic System

In this part we will discuss some basic aspects of the non-linear system identification using from among numerous neural networks structures only Multi-Layer Perceptron – MLP (a feed-forward neural network) [1], [9] with respect to the model based neural predictive control, where the control law is based upon the neural model. We will use in this paper a feed-forward neural network MLP with a single hidden layer. This structure is shown in the matrix notation in Fig. 6 [5], [7]. The matrix $\boldsymbol{W}_1$ represents the input weights, the matrix $\boldsymbol{W}_2$ represents the output weights, $\boldsymbol{F}_h$ represents a vector function containing the non-linear (*tanh*) neuron functions. The "1" shown in Fig. 6 together with the last column in the matrix $\boldsymbol{W}_1$ gives the offset in the network. The network input is represented by vector $\boldsymbol{Z}_{in}$ and the network output is represented by vector $\hat{\boldsymbol{Z}}_{out}$. The mismatch between the desired output $\boldsymbol{Z}_{out}$ and an aproximated output $\hat{\boldsymbol{Z}}_{out}$ is the prediction error $\boldsymbol{E}$.

The output from the neural network MLP can be written as

$$\boldsymbol{X}_2=\boldsymbol{W}_2\boldsymbol{F}_h\left(\boldsymbol{W}_1\begin{bmatrix}\boldsymbol{Y}_0\\1\end{bmatrix}\right). \tag{12}$$

From a trained MLP by Back-Propagation Error Algorithm (BPA – the first-order gradient method) a gain matrix $\boldsymbol{M}$ can be found by differentiating with respect to the input vector of the network. The gain matrix $\boldsymbol{M}$ can be calculated using (12) as

$$\boldsymbol{M} = \frac{d\hat{\boldsymbol{Z}}_{out}}{d\boldsymbol{Z}_{in}} = \frac{d\boldsymbol{X}_2}{d\boldsymbol{Y}_0^T} = \frac{d\boldsymbol{X}_2}{d\boldsymbol{Y}_1^T}.\frac{d\boldsymbol{Y}_1}{d\boldsymbol{X}_1^T}.\frac{d\boldsymbol{X}_1}{d\boldsymbol{Y}_0^T} = \boldsymbol{W}_2.\boldsymbol{F}_h{}'\left(X_1\right).\boldsymbol{W}_1^* \tag{13}$$

where $\boldsymbol{W}_1^* \square \boldsymbol{W}_1$ (excluding the last column).

The above mentioned the gain matrix $\boldsymbol{M}$ allows an on-line estimation of the actual parameters from an off-line trained neural model of the non-linear system.

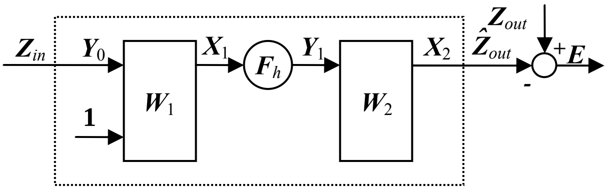

Figure 6
The matrix block diagram of the neural network MLP

With the inspiration from linear ARX model described in [8], next we will use a general model structure suitable for representing the dynamics of a wide range of non-linear system – neural ARX model (NARX) [4], [10], which is defined by

$$\hat{\boldsymbol{Y}}(k) = \boldsymbol{F}_N\left(\boldsymbol{Y}(k-1),...,\boldsymbol{Y}(k-p),\boldsymbol{U}(k-1),...,\boldsymbol{U}(k-m),\boldsymbol{\theta}\right)$$

$$\boldsymbol{Y}(k) = \hat{\boldsymbol{Y}}(k) + \boldsymbol{E}(k) \tag{14}$$

where $\boldsymbol{F}_N$ is the unknown non-linear vector function to be approximated, $\boldsymbol{E}(k)$ is the prediction error, p and m denote the number of delayed outputs and inputs. The non-linear mapping $\boldsymbol{F}_N$ can be approximated by a feed-forward neural network, e.g. MLP. The feed-forward neural network $\boldsymbol{F}_N$ is configured to represent the NARX model by applying p delayed values of the system output and m delayed values of the system input to the network inputs and assigning it's output $\hat{\boldsymbol{Y}}(k)$ to be $\boldsymbol{Y}(k)$, $\boldsymbol{\theta}$ is the vector of the network parameters.

An optimal value of the network parameters is usually obtained by using a training algorithm BPA that minimizes the following cost function:

$$J_{NN}(\boldsymbol{\theta}) = \sum_{\boldsymbol{i}=1}^{\boldsymbol{N}} \left(\boldsymbol{Y}(i) - \hat{\boldsymbol{Y}}(i,\boldsymbol{\theta})\right)^2 \tag{15}$$

on the basis of this gradient with respect to the network parameters. N in equation (15) is the length of input-output data set used for the network training.

We will consider that the neural NARX model has the input vector $\boldsymbol{Z}_{in}$ and the output vector $\hat{\boldsymbol{Z}}_{out}$:

$$\begin{aligned}&\boldsymbol{Z}_{in}(k)=\{\boldsymbol{Y}(k-1),\ldots,\boldsymbol{Y}(k-p),\ldots,\boldsymbol{U}(k-1),\ldots,\boldsymbol{U}(k-m)\}\\&\hat{\boldsymbol{Z}}_{out}(k)=\hat{\boldsymbol{Y}}(k)\end{aligned}\qquad(16)$$

After training the neural network MLP by BPA the actual gain matrix $\boldsymbol{M}(k)$ can be on-line estimated and calculated for neural ARX model as

$$\begin{aligned}\boldsymbol{M}(k)&=\frac{d\hat{\boldsymbol{Z}}_{out}(k)}{d\boldsymbol{Z}_{in}^{T}(k)}=\frac{d\hat{\boldsymbol{Y}}(k)}{d\left\{\boldsymbol{Y}(k-1)\ \ldots\ldots\ldots\ldots\ \boldsymbol{U}(k-m)\ \right\}^{T}}=\\&=\left\{-\hat{a}_1(k)\ldots-\hat{a}_p(k)\ \hat{b}_1(k)\ldots\hat{b}_m(k)\right\}\end{aligned}\qquad(17)$$

where $\hat{a}_i(k)$ for i=1,…, p, $\hat{b}_i(k)$ for i = 1, …, m are estimated parameters of the neural NARX model for step k. Using the method of an instantaneous linearization we can extract the parameters of the dynamic system from an off-line trained neural model using the gain matrix [5], [6] instead an estimation parameters of the linear model by the method of the least square. The GPC algorithm uses these estimated parameters from the neural model for control signal calculation.

6 Simulation Results of GPC Algorithm based on Neural Model – Intelligent Approach

The results of the estimation the system parameters from an off-line trained neural NARX model and their application in GPC algorithm using an algebraic theory are presented for a non-linear test SISO system – two tanks system (11). The output of the system is the water level in the second tank $y(t)=h_2(t)$. . We consider NARX model with 4 inputs and 6 neurons in the hidden layer. The activation functions in the hidden layer are „*tanh*" functions and in the output layer is selected a linear function. The actual values of the estimated parameters can be obtained from the gain matrix $M(k)$ by (17).

The GPC algorithm in the *k - th* step consists the following calculations:

1 the real output of the system (11) $y(k)$ is measured,

2 the calculation of the linearized parameters $\hat{a}_i,\hat{b}_i$ from the neural NARX model by the gain matrix (17),

3 the estimated parameters are implemented in the GPC algorithm and an optimal control value is obtained by minimized the cost function (1),

4 an optimal control value is applied on the system input,

5 $k = (k+1)T_{VZ}$ and go to step 1.

The implementation of the resulting GPC controller including a neural network model for parameter estimation is shown on Fig. 7.

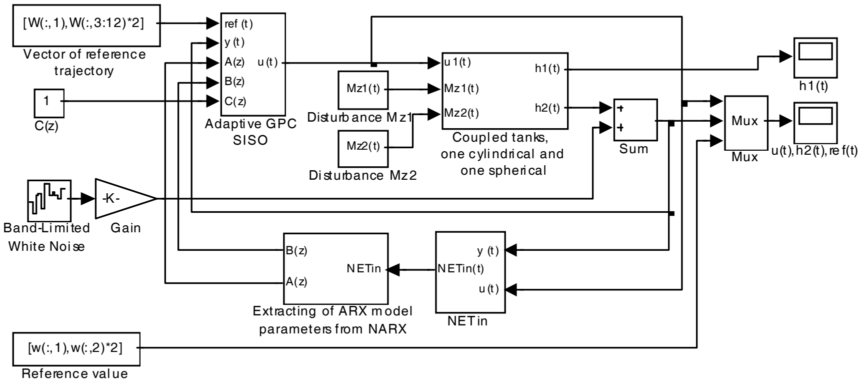

Figure 7

The GPC control scheme using neural model

The presentation of the results of GPC strategy using on-line parameter estimation from an off-line trained neural model is illustrated on Figs. 8 and 9, where the noise of measurement – the white noise is added to the output of the system (11) at $N_1 = 1, N_2 = 10, N_u = 5, \lambda = 0.1$. The disturbances input flows are added to the input of the non-linear system.

The plot on Fig. 8 compares the reference output of the system $h_{2ref}(t)$ and the actual output of the closed-loop system $h_2(t)$. A perfect model-following behaviour is achieved, although we can see an oscillating control signal on Fig. 9. This simulation example shows the possibility of an application of the neural modeling using the structure ARX known from the theory of the linear identification and also the possibility to apply GPC algorithm known from linear control theory for the control of non-linear SISO systems which have no the hard nonlinearities in the control structure in Fig. 5.

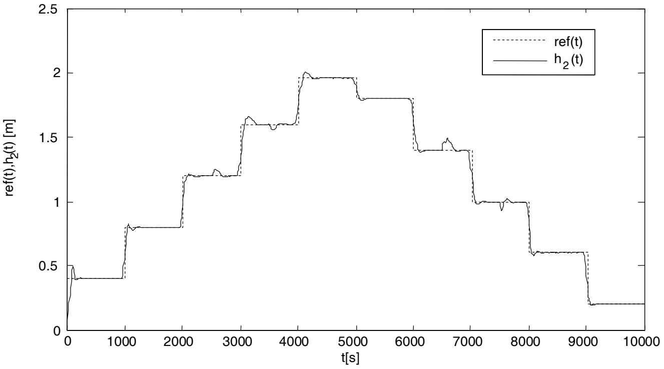

Figure 8

Tracking the reference trajectory *ref*(*t*) by the output of the non-linear system $h_2(t)$ using the parameter estimation from neural NARX model

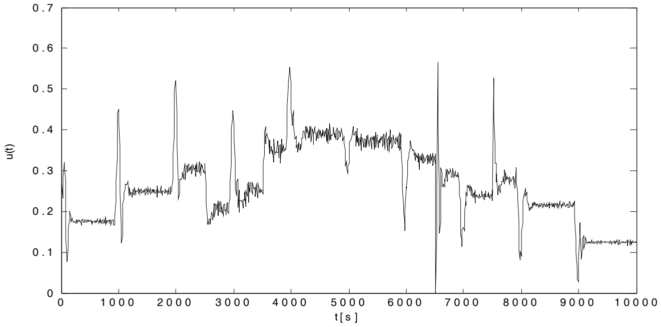

Figure 9

The control signal of the GPC controller based on parameter estimation from neural NARX model

The library *PredicLib* is created as a software tools by Simulink and Matlab language built-in functions. It contains blocks of Model Predictive Control (MPC) algorithms, concrete GPC algorithms for SISO and MIMO systems and MPC algorithms based on the state space model of the systems. PredicLib (Fig. 10) contains also the blocks for an extracting of the parameters from neural NARX model and extracting of the discrete state space model parameters from neural *Nonlinear Innovation State Space* (NISS) model of the controlled system, a block of the discrete Kalman estimator and a block for conversion of the discrete state space model to the transfer function. Some demo simulations of MPC algorithms for the control of the linear and the non-linear systems are included in the library. Some of the predictive control algorithms blocks demo applications created by using Matlab Web Server are accessible Virtual laboratory *CyberVirtLab* [*http://cyberneticsmws.fei.tuke.sk/MatlabWebServer_welcome*].

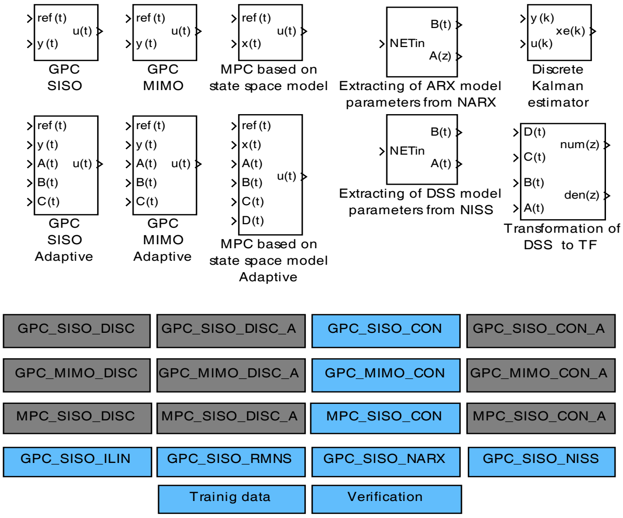

Figure 10

Screenshot of the library PredicLib from Simulink

Conclusion

In this paper are presented two approaches - classical and intelligent for the design of GPC algorithm for a non-linear system. The disadvantage of the classical approach of GPC strategy using as linearization technique for an estimate parameters of the dynamic model is that analytic description of non-linear system must be given. If analytical model of the dynamic system is unknown the better solution is to use an intelligent approach – GPC based on the neural NARX or NISS model for MIMO non-linear systems. In this paper is neural NARX model trained as an one-step predictor for a non-linear SISO system. After training this NARX model was used for on-line estimation of the system parameters which allow to calculate a linear predictor. This linear predictor of the system was used for solving of an optimization problem of GPC algorithm.

The practical simulations by the language Matlab/Simulink, Neural Toolbox and PredicLib illustrate, that this intelligent neural GPC control strategy using linearization technique by the gain matrix produces more excellent performance for control of the non-linear system as GPC strategy using of the classical approach.

Applying the principle of an instantaneous linearization to the GPC design gives tremendous advantages over the conventional nonlinear predictive control design.

Acknowledgment

This research has been supported by the Scientific Grant Agency of Slovak Republic under project Vega No. 1/0617/08 Multiagent Network Control Systems with Automatic Reconfiguration.

References

[1] S. Chen, S.Billings, P. Grant: Non-linear System Identification Using Neural Networks, International Journal of Control 51, No. 6, pp. 1191-1214, 1990

[2] D. W. Clarke, C. Monthadi, P. S. Stuffs: Generalized Predictive Control, Automatica, Vol. 23, pp. 137-148, part 1, part 2, 1987

[3] C. E. Garcia, D. M. Prett, M. Morari: Model Predictive Control: Theory and Practise – A Survey, Automatica, Vol. 25, pp. 335-348, 1989

[4] A. Jadlovská, J. Sarnovský: Using of Dynamic Neural Models in Feedforward Control Structure", in Automatizace - Journal of Automation, No. 4, Vol. 46, pp. 236-241, Praha, ISSN 0005-125X, (in Slovak), 2003

[5] A. Jadlovská: Modeling and Control of Dynamic Processes Using Neural Networks, FEI-TU, Košice, Informatech Ltd., p. 173, ISBN 80-88941-22-9, (in Slovak), 2003

[6] N. Kabakov: Predictive Control of Non-linear Processes, Thesis to Dissertation Exam, FEI-TU Košice, 53 pages, (in Slovak), 2006

[7] I. J. Leontaritis: Input-Output Parametric Models for Non-linear Systems, Part 1, Part 2, International Journal of Control, No. 2, Vol. 41, pp. 303-344, 1985

[8] L. Ljung: System Identification, Theory for User, Prentice Hall, 1987

[9] K. S. Narendra, A. U. Levin: Identification and Control of Dynamical Systems Using Neural Networks, IEEE Transaction on Neural Networks 1, No. 1, pp. 4-27, 1990

[10] M. Norgaard: NNSID Toolbox, Technical report, Department of Automation, DTU, 1997