

GENERALIZED PREDICTIVE CONTROL DESIGN FOR A NONLINEAR HYDRAULIC SYSTEM

Anna JADLOVSKÁ, Štefan JAJČIŠIN

Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, e-mail: anna.jadlovaska@tuke.sk, stefan.jajcisin@tuke.sk

ABSTRACT

The paper deals with the design of optimization predictive control algorithms, which are based on the input-output ARX and CARIMA models of the dynamic systems. The paper also deals with the simulation verification of predictive control algorithms on a nonlinear hydraulic system with different settings of various control algorithms parameters, with a system subject to noise, or disturbance. There is mentioned a brief theoretical derivation and the particular formulas for concrete predictive control methods in the paper. The paper includes a flow chart diagram for predictive control methods programming and the control structure, where predictive control algorithms were used in nonlinear hydraulic system control. Nonlinear differential equations, which express hydraulic system's dynamic are presented, too. The results of generalized predictive control algorithms comparison are expressed by the control action and the hydraulic system's output in a graphs form.

Keywords: optimal control, generalized predictive control, criterion minimization, quadratic programming, nonlinear simulation model

1. INTRODUCTION

In the paper we are engaged in designing of generalized predictive control (GPC) algorithms, that are based on the dynamic systems' models input-output description, on the transfer functions.

Particularly, we are concerned in the GPC algorithm, which is based on the ARX dynamic system's model and was used in adaptive predictive control in [1] and [2] and the GPC algorithm based on the CARIMA dynamic system's model, which contains typically an integrator in linear model expression [3].

In the second part we mention a brief theoretical derivation of single algorithm and the particular formulas.

We analyze in detail the GPC algorithms design in a function form, which serves for control action computing in the dynamic systems control process in the part 3.

In the fourth part we are concerned in simulation control of a nonlinear hydraulic system using designed GPC algorithms in the particular control structure.

We are presenting the results acquired by GPC algorithms comparing with different settings of the various algorithms' parameters, possibly with noisy system's output or with an impact of the disturbance on the control action in the paper.

2. GENERALIZED PREDICTIVE CONTROL ALGORITHMS OVERVIEW

Generalized predictive control algorithms belong to optimization tasks. In general they minimize the criterion

$$J = \sum_{i=N_1}^{N_p} Q(i) [\hat{y}(k+i) - w(k+i)]^2 + \sum_{i=1}^{N_u} R(i) [u(k+i-1)]^2 \quad (1)$$

where: $u(k)$ is an control action, $\hat{y}(k)$ is a predicted output value and $w(k)$ is a reference trajectory.

Values N_1 and N_p represent a prediction horizon, N_u denotes a control horizon with $N_u \leq N_p$. If reduction of

degrees of freedom is used in a predictive control algorithm, $N_u < N_p$ [4]. $Q(i)$ represents weighting coefficients of a deviation between the system output and the reference trajectory on the prediction horizon and $R(i)$ represents weighting coefficients of the control action on the control horizon.

Predictive control algorithm design can be divided in two steps:

- I) predictor derivation,
- II) computing optimal sequence of control actions.

Advantage of predictive control algorithms consists in possibility to compose various constraints in computation of the optimal control sequence. Most commonly it is carried out by quadratic programming. In our case we used the *quadprog* function, that is one of functions of the *Optimization Toolbox* in the Matlab and computes a vector of the optimal values \mathbf{u} by formula

$$\min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + \mathbf{g}^T \mathbf{u} \quad \text{subject to} \quad \mathbf{A}_{con} \mathbf{u} \leq \mathbf{b}_{con}, \quad (2)$$

where the form of the matrix \mathbf{H} and the vector \mathbf{g} depend on used algorithm and matrix \mathbf{A}_{con} and vector \mathbf{b}_{con} are composed in compliance with required constraints.

2.1. GPC algorithm based on the ARX model

This algorithm is based on the regression ARX model of the dynamic systems:

$$\mathbf{A}_z(z^{-1})y(k) = \mathbf{B}_z(z^{-1})u(k) + \xi(k), \quad (3)$$

where $\mathbf{B}_z(z^{-1})$ is m ordered polynomial numerator with b_i coefficients and $\mathbf{A}_z(z^{-1})$ is n ordered polynomial denominator with a_i coefficients.

$u(k)$ is input, $y(k)$ is output and $\xi(k)$ is a system output error or output noise measurement [1].

I) Predictor derivation

According to [1] we can express the output of the dynamic system in sample $k + 1$ from the ARX model (3) by relation

$$y(k+1) = \sum_{i=1}^n b_i u(k-i+1) - \sum_{i=1}^n a_i y(k-i+1). \quad (4)$$

We are able to arrange (4) into a matrix form

$$\begin{pmatrix} y(k-n+2) \\ \vdots \\ y(k) \\ y(k+1) \end{pmatrix} = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ 0 & & & 1 \\ -a_n & \cdots & & -a_1 \end{pmatrix} \begin{pmatrix} y(k-n+1) \\ \vdots \\ y(k-1) \\ y(k) \end{pmatrix} + \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 0 \\ b_n & \cdots & b_1 \end{pmatrix} \begin{pmatrix} u(k-n+1) \\ \vdots \\ u(k-1) \\ u(k) \end{pmatrix} + \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 0 \\ b_n & \cdots & b_1 \end{pmatrix} \begin{pmatrix} u(k-n+1) \\ \vdots \\ u(k-1) \\ u(k) \end{pmatrix} \quad (5)$$

$$\begin{aligned} y(k) &= (0 \quad \cdots \quad 0 \quad 1) X(k) \\ y(k) &= \mathbf{C} X(k) \end{aligned}$$

which represents the „pseudostate“ space model of the dynamic system [1].

By derivation mentioned in [1] or [8] it is possible to express the predictor from the pseudostate space model in form of

$$\begin{pmatrix} \hat{y}(k+1) \\ \vdots \\ \hat{y}(k+N_p) \end{pmatrix} = \begin{pmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{N_p} \end{pmatrix} \begin{pmatrix} y(k-n+1) \\ \vdots \\ y(k) \end{pmatrix} + \begin{pmatrix} \hat{y} \\ \vdots \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \bar{y}_0 \\ \vdots \\ \bar{y}_0 \end{pmatrix} + \begin{pmatrix} \mathbf{CB}_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \mathbf{CB}_{N_p-1} & & \end{pmatrix} \begin{pmatrix} u(k-n+1) \\ \vdots \\ u(k+N_p-1) \end{pmatrix} + \begin{pmatrix} \bar{\mathbf{G}} \\ \vdots \\ \bar{\mathbf{G}} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{u}} \\ \vdots \\ \bar{\mathbf{u}} \end{pmatrix} \quad (6)$$

$$\hat{\mathbf{y}} = \bar{\mathbf{y}}_0 + \bar{\mathbf{G}}_{(:,1:n-1)} \begin{pmatrix} u(k-n+1) \\ \vdots \\ u(k-1) \end{pmatrix} + \bar{\mathbf{G}}_{(:,n:n+N_p-1)} \begin{pmatrix} u(k) \\ \vdots \\ u(k+N_p-1) \end{pmatrix}$$

$$\hat{\mathbf{y}} = \mathbf{y}_0 + \mathbf{G}_{N_p} \mathbf{u}$$

where \mathbf{y}_0 introduces the *free response* and $\mathbf{G}_{N_p} \mathbf{u}$ represents the *forced response* of the system. Following the control horizon length it is necessary to create a $N_p \times N_u$ matrix

$$\mathbf{U} = \begin{pmatrix} \mathbf{I} & & & \\ & \ddots & & \\ & & \mathbf{I} & \\ & & & \vdots \\ & & & \mathbf{I} \end{pmatrix}. \quad (7)$$

It is recommended to right multiply \mathbf{U} with \mathbf{G}_{N_p} , whereby we obtain a matrix $\mathbf{G} = \mathbf{G}_{N_p} \mathbf{U}$, that ensures considering the real control horizon in computation of the optimal control sequence [1]. Thus, the final matrix form of the predictor will be

$$\hat{\mathbf{y}} = \mathbf{y}_0 + \mathbf{G} \mathbf{u}. \quad (8)$$

II) Computation of the optimal control sequence

The algorithm minimizes the criterion

$$J_{\text{ARX}} = \sum_{i=N_1}^{N_p} \{Q(i) [\hat{y}(k+i) - w(k+i)]\}^2 + \sum_{i=1}^{N_u} \{R(i) [u(k+i-1)]\}^2 \quad (9)$$

which in contrast to (1) includes weighting coefficients.

A matrix form of the criterion J_{ARX} is

$$J_{\text{ARX}} = ((\hat{\mathbf{y}} - \mathbf{w})^T \quad \mathbf{u}^T) \begin{pmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \begin{pmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ \mathbf{u} \end{pmatrix}, \quad (10)$$

whereby according to [2] it is sufficient to minimize only one part

$$J_k = \begin{pmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{Q} \mathbf{G} \\ \mathbf{R} \end{pmatrix} \mathbf{u} - \begin{pmatrix} \mathbf{Q}(\mathbf{w} - \mathbf{y}_0) \\ \mathbf{0} \end{pmatrix} \quad (11)$$

by solving algebraic equations

$$\begin{pmatrix} \mathbf{Q} \mathbf{G} \\ \mathbf{R} \end{pmatrix} \mathbf{u} - \begin{pmatrix} \mathbf{Q}(\mathbf{w} - \mathbf{y}_0) \\ \mathbf{0} \end{pmatrix} = \mathbf{0} \\ \mathbf{S} \mathbf{u} - \mathbf{T} = \mathbf{0} \quad (12)$$

Since the matrix \mathbf{S} is not square, according to [5], for computing the optimal control sequence \mathbf{u} without considering required constraints it is possible to use the QR-decomposition or the pseudoinversion of the matrix \mathbf{S} :

$$\mathbf{u} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \mathbf{T}. \quad (13)$$

The matrix \mathbf{H} and the vector \mathbf{g} , which are needed in computation of the optimal control sequence \mathbf{u} by *quadprog* function with required constraints respecting, have with control action value \mathbf{u} weighted in criterion (10) form

$$\begin{aligned} \mathbf{H} &= \mathbf{G}^T \mathbf{Q}^T \mathbf{Q} \mathbf{G} + \mathbf{R}^T \mathbf{R} \\ \mathbf{g}^T &= (\mathbf{y}_0 - \mathbf{w})^T \mathbf{Q}^T \mathbf{Q} \mathbf{G} \end{aligned} \quad (14)$$

In the case, where the control move $\Delta \mathbf{u}$ is weighted in the criterion (10) it is necessary to modify equations (11), (12) and (14). This technique is introduced in [8].

2.2. GPC algorithm based on the CARIMA model

The algorithm is derived from the CARIMA model of the dynamic system

$$A_z(z^{-1})y(k) = B_z(z^{-1})u(k-1) + \frac{C_z(z^{-1})}{\Delta} \xi(k), \quad (15)$$

where in contrast to the ARX model $C_z(z^{-1})$ is multi-nominal and $\Delta = 1 - z^{-1}$ introduces an integrator [3].

According to [3] the criterion, that is minimized in this GPC algorithm has form

$$J_{\text{CARIMA}} = \sum_{i=N_1}^{N_p} [\hat{y}(k+i) - w(k+i)]^2 + \lambda \sum_{i=1}^{N_u} [\Delta u(k+i-1)]^2, \quad (16)$$

where $\lambda = R/Q$ is relative weighting quotient.

I) Predictor derivation

According to [3] the output of a SISO dynamic system in sample $k+1$ is given by equation (without z^{-1}):

$$y(k+j) = \frac{B_z}{A_z} u(k+j-1) + \frac{C_z}{\Delta A_z} \xi(k+j). \quad (17)$$

Solving diophantine equations

$$C_z = E_j \Delta A_z + z^{-j} F_j \quad \text{a} \quad B_z E_j = G_j C_z + z^{-j} \Gamma_j \quad (18)$$

it is possible to express the predictor for j steps in form

$$\hat{y} = \mathbf{G} \Delta \mathbf{u} + \mathbf{y}_0, \quad (19)$$

where \mathbf{y}_0 is system's free response and coefficients of matrix \mathbf{G} are given by matrix dividing $B/\Delta A$,

$$\mathbf{G} = \begin{pmatrix} g_0 & 0 & \cdots & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \\ \vdots & & & g_0 & 0 \\ g_{N_p-1} & \cdots & & & g_0 \end{pmatrix}.$$

According to [3] we can reduce the matrix \mathbf{G} to size $(N_p - N_1 + 1) \times N_u$ on the basis of values N_1 and N_u .

II) Computation of the optimal control sequence

From the matrix form of the criterion (16)

$$\begin{aligned} J_{\text{CARIMA}} &= (\hat{\mathbf{y}} - \mathbf{w})^T (\hat{\mathbf{y}} - \mathbf{w}) + \lambda \Delta \mathbf{u}^T \Delta \mathbf{u} = \\ &= (\mathbf{G} \Delta \mathbf{u} + \mathbf{y}_0 - \mathbf{w})^T (\mathbf{G} \Delta \mathbf{u} + \mathbf{y}_0 - \mathbf{w}) + \lambda \Delta \mathbf{u}^T \Delta \mathbf{u} =, \quad (20) \\ &= c + 2\mathbf{g}^T \Delta \mathbf{u} + \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} \end{aligned}$$

where $\mathbf{H} = \mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}$,
 $\mathbf{g}^T = (\mathbf{y}_0 - \mathbf{w})^T \mathbf{G}$,

it is possible to express a formula for the control action without required constraints computing by minimization condition

$$\frac{\partial J_{\text{CARIMA}}}{\partial \Delta \mathbf{u}} = \mathbf{0} : \quad \Delta \mathbf{u} = -\mathbf{H}^{-1} \mathbf{g}. \quad (21)$$

For computing the optimal control action sequence with required constraints it is possible to use quadratic programming, where input parameters of the *quadprog* function are the matrix \mathbf{H} , vector \mathbf{g} , matrix and vector \mathbf{A}_{con} and \mathbf{b}_{con} representing the constraints.

It is important to note, that control move $\Delta \mathbf{u}$ is result of GPC algorithm computing, which is based on the CARIMA model. In the case the control action value \mathbf{u} is weighted in criterion (20), the optimal sequence of the control action is given by

$$\mathbf{u} = -\mathbf{H}^{-1} \mathbf{g}, \quad (22)$$

where $\mathbf{H} = \mathbf{G}^T \mathbf{G} + \lambda \mathbf{D}_i^{-T} \mathbf{D}_i^{-1}$, $\mathbf{g}^T = (\mathbf{y}_0 - \mathbf{w})^T \mathbf{G} + \lambda \mathbf{u}_k^T \mathbf{D}_i^{-T} \mathbf{D}_i^{-1}$.

3. GPC ALGORITHMS DESIGN

We programmed GPC algorithms mentioned in the previous part as Matlab function form, which compute the control action value on the basis of input parameters.

In Fig. 1 the flow chart, which corresponds to GPC algorithms programming is depicted. The equation (21) computes the control move, thus in the case of GPC algorithm, based on the CARIMA model, the previous value of the control action must be added to the function's output.

The algorithms are designed in such way that it is possible to carry out computation of the optimal control sequence by receding horizon principle. It means the next procedure is evaluated in mentioned functions within every control step:

1. assigning of the reference trajectory \mathbf{w} over the prediction horizon,
2. detection of the actual state $\mathbf{x}(k)$ or output $\mathbf{y}(k)$ of the system in specific sample,
3. prediction of system response on the prediction horizon based on the actual values of the control action $u_{\text{opt}}(k)$ and state $\mathbf{x}(k)$ in previous samples without influence of next control action, so-called system free response,
4. the optimal control sequence \mathbf{u}_{opt} computing by the criterion J minimization with known parameters N_1 , N_p , N_u , $Q(i)$ and $R(i)$, using $u_{\text{opt}}(k)$ to the system input.

4. SIMULATION VERIFICATION OF GPC ALGORITHMS ON HYDRAULIC SYSTEM

We carried out simulation control of the nonlinear hydraulic system with designed GPC algorithms on the basis of the control structure, that is shown in Fig. 2.

The algorithm, which we used for feedback control structure programming has several steps:

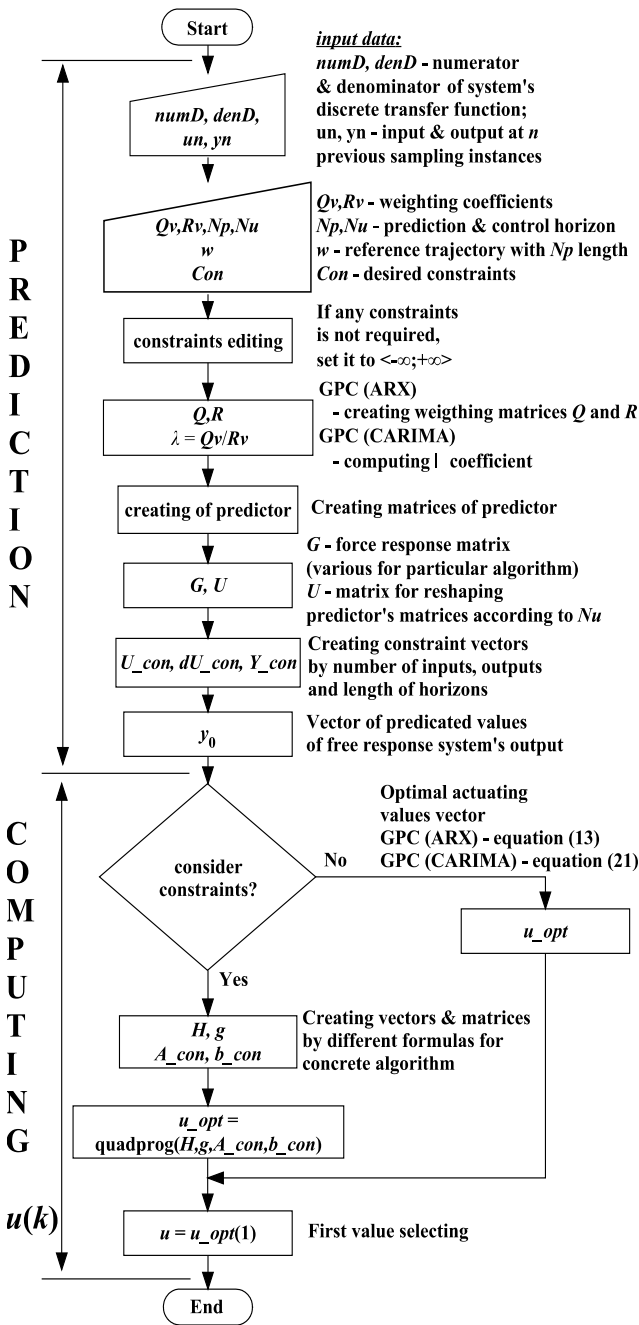


Fig. 1 Flow chart diagram for GPC algorithms

1. Loading input data
 - numerator and denominator of a discrete transfer function,
 - sample time T_{vz} ,
 - time of control (simulation) T_{sim} ,
 - vector of reference trajectory w ,
 - noise and disturbance $d(k)$ setting.
2. Data initialization
 - model's initial output $y(0)$ setting,
 - zeroising the data, which represents past values in input-output description of dynamic system,
 - number of samples establishing: $n_v = T_{sim} / T_{vz} + 1$,
 - counter of samples setting: $k \leftarrow n$, where n is system order.

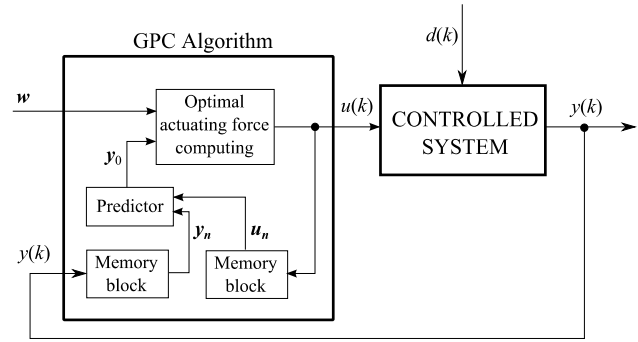


Fig. 2 Control structure with GPC algorithm

3. Computing the deviation $e(k) = w(k) - y(k)$.
4. Computing the control action $u(k)$ by specific equation for used GPC algorithm and eventually the control action saturation.
5. Disturbance value addition to the control action (if it is required).
6. New output $y(k)$ computing with using computed control action value $u(k)$ on basis of the system discrete transfer function.
7. Noise addition to the output (if it is required).
8. Setting the output of the dynamic system linear model by output of the controlled system.
9. $k \leftarrow k + 1$.
10. If $k \leq n_v + n$, then jump to 3. step.
11. Clearing of samples in inverse time.
12. Visual display of simulation results.

4.1. Nonlinear hydraulic system

The hydraulic system (adapted from [6]), which we used in GPC algorithms verification and comparing, is composed of two tanks with areas S_2 and S_3 and heights h_{2max} and h_{3max} . The tanks are joined by the valve V_1 with relative opening L_{V1} and with area S_{V1} . A liquid is ducted to the first tank from a receiver through a feed-pipe with area S_1 and height h_{1max} by a pump and it flow out from the second tank through the valve V_2 with relative opening L_{V2} and area S_{V2} back to the receiver. The control input to the hydraulic system is the motor's voltage of pump U_{mot} , disturbance value is the opening of the second valve L_{V2} , we consider height of the second tank $h_3(t)$ as an output. The area of the pump's untightness is S_c , U_{insens} represents a voltage of motor's insensitivity.

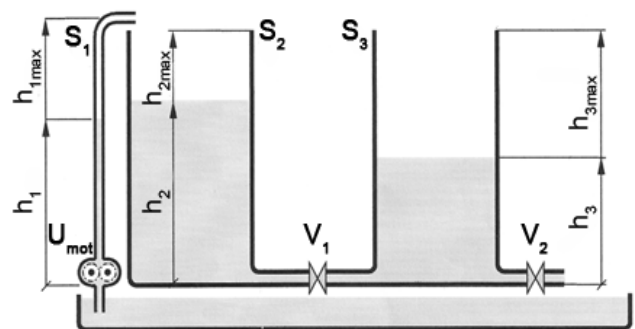


Fig. 3 Hydraulic system

The mathematical description of the hydraulic system's model is created by the nonlinear differential equations

$$\begin{aligned} \dot{h}_2(t) &= \frac{1}{S_2} [k(U_{mot} - U_{insens}) - S_c \sqrt{2gh_1(t)} - S_{v1} L_{v1} \sqrt{2g[h_2(t) - h_3(t)]}] , \text{ if } h_1(t) \geq h_{1max}, \\ \dot{h}_2(t) &= \frac{1}{S_2} [0 - S_{v1} L_{v1} \sqrt{2g[h_2(t) - h_3(t)]}] , \text{ if } h_1(t) < h_{1max}, \\ \dot{h}_3(t) &= \frac{1}{S_3} [S_{v1} L_{v1} \sqrt{2g[h_2(t) - h_3(t)]} - S_{v2} L_{v2} \sqrt{2gh_3(t)}] . \end{aligned} \quad (23)$$

Detailed derivation of the mathematical description and numerical parameters of this system are presented in [7].

We programmed a numerical method Runge-Kutta of 4th order as a function in Matlab to solve the nonlinear differential equations (23), which describe the dynamics of the hydraulic system. Within this function we programmed also physical constraints of the hydraulic system:

$$\begin{aligned} 0V &\leq U_{mot} \leq 10V, \\ 0m &\leq h_2(t) \leq 0.49m, & 0 &\leq L_{v1} \leq 1, \\ 0m &\leq h_3(t) \leq 0.49m, & 0 &\leq L_{v2} \leq 1. \end{aligned} \quad (24)$$

We used the physical constraints of hydraulic system (24) noted above in designed GPC algorithms in computation of the optimal control sequence by the *quadprog* function.

Although we verified designed GPC algorithms on the nonlinear hydraulic system, it is necessary to note, that we stated the predictor on the basis of linear model for each algorithm. We established the linear model of the hydraulic system by the Taylor linearization approach of the nonlinear differential equations (23) in the operating point: $U_{mot} = 8V$; $L_{v1} = 0.4$; $L_{v2} = 0.2$. This operating point ensures a state of hydraulic system, where water levels in both tanks are around the middle height of tanks.

4.2. Simulation control of hydraulic system results

In Fig. 4 the results of simulation control of the hydraulic system are shown, where GPC algorithms was dealt with settings: $N_p = 10$, $N_u = 2$, Q is a diagonal matrix with values $Q_v = 1000$, R is a diagonal matrix with values $R_v = 0.001$, the noise and the disturbance value were not considered in this case, the control move $\Delta u(k)$ was weighted in the criterion (1).

The fundamental difference is evident between GPC algorithms, particularly the GPC algorithm based on the ARX model does not introduce any integration character into the control process.

This mentioned difference between GPC algorithms is more visible with settings $N_p = 50$, $N_u = 1$, $Q_v = 1000$, $R_v = 0.01$, the results are shown in Fig. 6.

It is needed to note, in that case, when the control action value $u(k)$ was weighted in the criterion (1), the deviation between the system output $y(k)$ and the

reference trajectory $w(k)$ existed in the control process with both GPC algorithms.

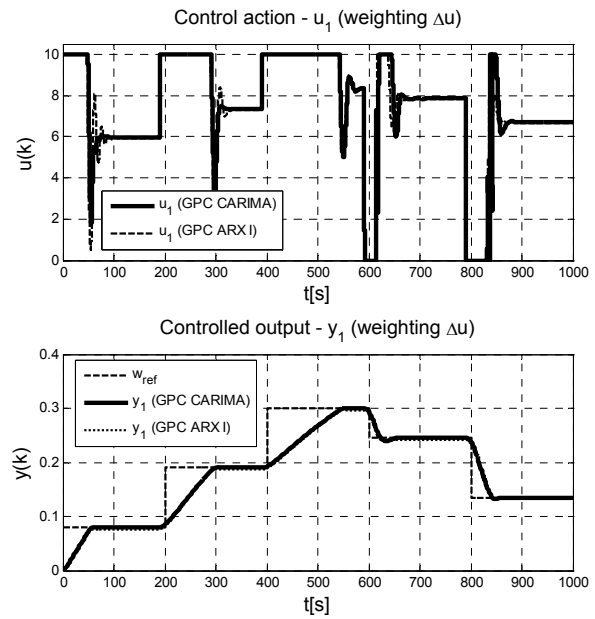


Fig. 4 GPC algorithms comparing without noise and disturbance value ($N_p = 10$, $N_u = 2$, $Q_v = 1000$, $R_v = 0.001$)

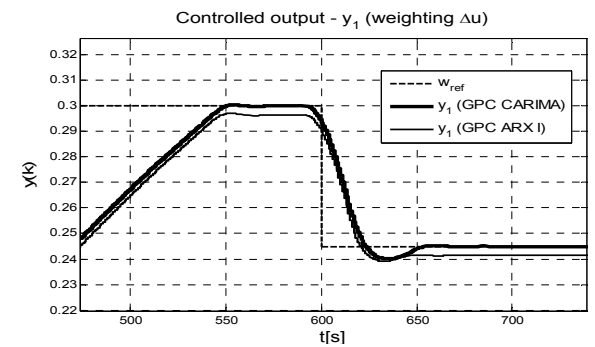


Fig. 5 Detail of time response from Fig. 4

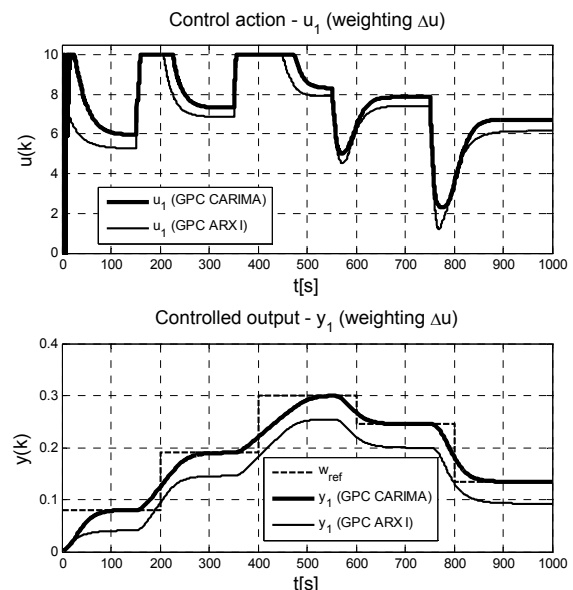


Fig. 6 GPC algorithms comparing without noise and disturbance value ($N_p = 50$, $N_u = 1$, $Q_v = 1000$, $R_v = 0.01$)

The results of the simulation control with noise, which mean value was 0 and standard deviation was 0.005 and 0.001 are shown in Fig. 7 and Fig. 8. In Fig. 8 it is seen, that the control action is with noisy system output more oscillatory for both GPC algorithms. Comparing Fig. 7 and Fig. 8 we can clearly see a massive control downgrade when the GPC algorithm based on the CARIMA model was used in the control with increasing noise, particularly its standard deviation. We assume the noise affects much more the linear predictor based on the CARIMA model than on the ARX model. The integrator $\Delta = 1 - z^{-1}$ in the CARIMA based predictor plays some role, too.

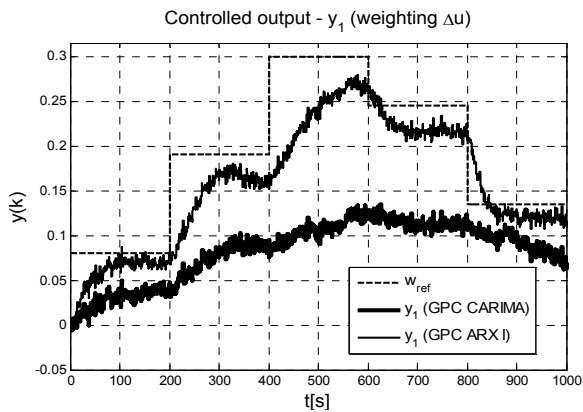


Fig. 7 GPC algorithms comparing with noise ($N_p = 10, N_u = 2, Q_v = 1000, R_v = 0.001, \sigma = 0.005$)

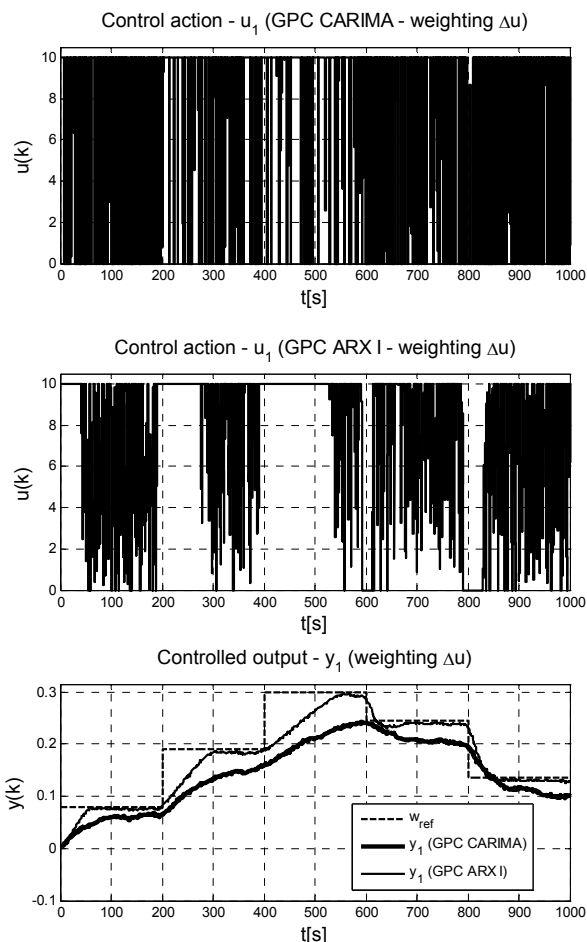


Fig. 8 GPC algorithms comparing with noise ($N_p = 10, N_u = 2, Q_v = 1000, R_v = 0.001, \sigma = 0.001$)

The results of simulation control, where a disturbance value was added into the control structure as increased or decreased value 6V in time 300s or 600s of the control action are visible in Fig. 9. It is seen from Figure, that both of GPC algorithms respond almost equally to disturbance impact, but with the GPC algorithm based on the ARX model above mentioned deviation between the system output $y(k)$ and the reference trajectory $w(k)$ appears.

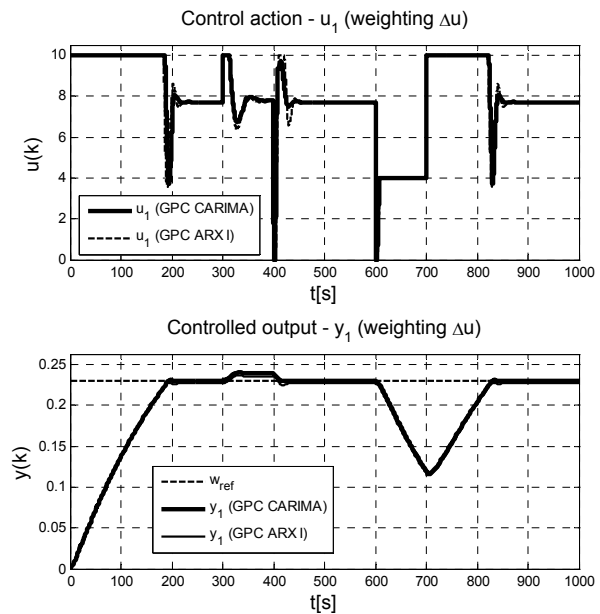


Fig. 9 GPC algorithms comparing with disturbance influence in time 300s and 600s ($N_p = 10, N_u = 2, Q_v = 1000, R_v = 0.001$)

We consider that is necessary to note for concluding, that the control has better performance with the prediction horizon increasing for both of GPC algorithms. Moreover, changing of the weighting coefficients in the matrices Q and R has much bigger influence on the control with the GPC algorithm based on the ARX model than with the GPC algorithm based on the CARIMA model. Both of mentioned facts can be deduced from Fig. 4 and Fig. 6 comparison.

5. CONCLUSIONS

We introduced two different generalized predictive control algorithms, which vary each other by the dynamic system model's type using and also by the form of the predictor expression. We were engaged in the basic theoretical fundamentals of GPC algorithms, the method for programming them in the function form, moreover the control structure programming. We concentrated on GPC algorithms comparing in nonlinear hydraulic system simulation control with various settings, whether in the control algorithms, the noise or the disturbance impact adjusting in the control process.

We can point out from the obtained results, that it is preferable to use the GPC algorithm based on the CARIMA model in the case, where the request for deviation between the output and the reference trajectory removal is present. On the other side, in the case the system is noisy, the GPC algorithm based on the ARX

model of the dynamic system seems to be more profitable. However, in this case the deviation between the output and desired value will exist.

Later we will verify designed GPC algorithms on a real laboratory hydraulic model, similar to the hydraulic system mentioned in this paper, which is owned by Department of Cybernetics and Artificial Intelligence. Like it is applied in this paper, we will carry out predictive control algorithms as Matlab functions, which will be called by a main program. The main program will provide a communication with a PLC, which controls a frequency converter of a pump's motor, by DDE or OPC protocol.

We prefer computing the control action in Matlab because its huge advantage is that it contains optimized and ready for use algorithms in toolboxes. In addition, GPC algorithms work with matrices a lot and they require the model of dynamic system during computing. We assume that it is more suitable to handle these problems in Matlab than in other structured programming languages.

Since computing the control action with constraints is very time-consuming, our goal is to use an explicit predictive control. It means computing the control action in advance by multi-parametric programming and use it in control loop. We expected it should be more convenient for implementation of GPC algorithms in PLC.

ACKNOWLEDGMENTS

This work was supported by Agency of the Ministry of Education of the Slovak Republic for the Structural Funds of the EU under the project Centre of Information and Communication Technologies for Knowledge Systems (project number: 26220120020).

REFERENCES

- [1] BELDA, K. – BÖHM, J.: Adaptive Predictive Control for Simple Mechatronic Systems. *Proceedings of the WSEAS CSCC & EE International Conferences*. WSEAS Press, Athens, Greece 2006, pp. 307–312.
- [2] BELDA, K. – BÖHM, J.: Adaptive Generalized Predictive Control for Mechatronic Systems. *WSEAS Transactions on Systems*. Vol. 5, Issue 8, August 2006, Athens, Greece 2006, pp. 1830–1837.
- [3] FIKAR, M.: Predictive Control – An Introduction. Slovak Technical University – FCT, Bratislava 1998.
- [4] ROUBAL, J.: Prediktívni regulátor (Příklady ze cvičení – Moderní teorie řízení). Available on the internet: http://support.dce.felk.cvut.cz/pub/roubalj/teaching/MTR/seminars/MTR_cv8_mpc.pdf
- [5] BELDA, K.: Control of Parallel Robotic Structures Driven by Electromotors. (Research Report). Czech Technical University, FEE, Prague 2004, p. 32.
- [6] RESCHKE, D.: Matlab Web Server ve výuce (in Czech). Diploma thesis (Supervisor: Ing. František Vaněk) Czech Technical University, FE, Prague 2002.
- [7] JAJČIŠIN, Š.: Modeling and Control of Dynamical Systems by Matlab/Simulink (in Slovak). Bachelor thesis (Supervisor: doc. Ing. Anna Jadlovská, PhD.) TU-FEI, Košice, 2008.
- [8] JAJČIŠIN, Š.: Application Modern Methods in Control of Non-linear Educational Models (in Slovak). Diploma thesis (Supervisor: doc. Ing. Anna Jadlovská, PhD.) TU-FEI, Košice 2010.

Received November 2, 2010, accepted February 1, 2011

BIOGRAPHIES

Anna Jadlovská was born October 29th, 1960. She received her MSc. degree in the field of Technical Cybernetics at the Faculty of Electrical Engineering of the Technical University in Košice in 1984. She defended her PhD thesis in the domain of Automatization and Control in 2001 at the same University; her thesis title was “Modelling and Control of Non-linear Processes Using Neural Networks”. Since 1993 she worked in the Department of Cybernetics and Artificial Intelligence Faculty of Electrical Engineering and Informatics Technical University in Košice as an Associate Assistant and since 2004 she has been working as an Associate Professor. Her main research activities include the problems of adaptive and optimal control – in particular predictive control with constraints for non-linear processes using neural networks and methods of artificial intelligence (Intelligent Control Design). She is the author of scientific articles and contributions to various journals and international conference proceedings, as well as being the co-author of some monographs.

Štefan Jajčišín was born August 22th, 1986. In 2010 he graduated (MSc.) with distinction at the Department of Cybernetics and Artificial Intelligence of the Faculty of Electrical Engineering and Informatics at Technical University in Košice. Since September 2010 he has been internal PhD. student at the Department of Cybernetics and Artificial Intelligence. The topic of his dissertation thesis is focused on predictive control algorithms for nonlinear dynamic systems. In addition, the sphere of his interest is physical dynamic system modeling, optimal control algorithms with constraints design and their verification on simulation and real laboratory models by Matlab/Simulink.