Adaptive Generalized Predictive Control for Mechatronic Systems

KVĚTOSLAV BELDA, JOSEF BÖHM
Department of Adaptive Systems
Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 182 08 Prague 8
CZECH REPUBLIC
belda@utia.cas.cz, bohm@utia.cas.cz http://as.utia.cz/asc

Abstract: - The paper deals with the design of discrete adaptive model-based predictive control for mechatronic systems. Mechatronic systems are considered generally as systems combining mechanical elements and elements which power, monitor and control these systems themselves. They have different inputs and outputs and their number determines not only control design, but also methods of adaptation. In this paper, a combination of on-line identification and generalized predictive control is introduced. The identification is based on least squares. The predictive control arises from state-space formulation. This idea is applied to ARX models representing Input/Output formulation. The presented algorithms are derived in computationally suitable square-root form and their correctness is documented by tests on laboratory model 'ball on rod'.

Key-Words: - On-line identification, Predictive control, Input/output equations of predictions, Real-time control

1 Introduction

With steady progress of industrial production, making demands on productivity and quality, there is a necessity to simultaneously develop different ways of the control of individual components included in the production. The components usually combine some mechanical elements (mechanisms) and also elements, mostly electrical, which power (actuators – drive units), monitor (sensors) or control (control units) the component state itself; i.e. altogether, it represents combination of mechanics and electronics, in single word – mechatronics.
The main purpose of the control often consists in fulfillment of some predetermined motion or in stabilization. The question is how to achieve such purpose.
One of the possibilities, being at the beginning of industrial application, is model-based approach. It offers complex solution on global level of whole controlled system and not only of its individual elements. In this paper, as a powerful way, the predictive control is investigated [1], [4]. The main stress is laid on obtaining of suitable model and its on-line use (adaptation) within control design.
The explanation is intended for mechatronic systems, considered both Single-Input/Single-Output type and systems with higher number of inputs and outputs.
The combination of on-line identification and generalized predictive control will be introduced. The identification is based on least squares [2], [3], which identifies the parameters of autoregressive model with external input (ARX model) describing real controlled mechatronic system.
The predictive control arises from state-space formulation [1], of which idea is applied to ARX models representing Input/Output formulation. The equations of predictions – the main part of predictive design – are specifically composed in pseudo-state form from identified parameters of ARX model. This form requires only values of inputs and outputs. Their number corresponds to the order of controlled system.
The presented algorithms are derived in computationally suitable square-root form. The algorithm correctness is documented by tests on laboratory model 'ball on rod', which is illustrated by Fig.1.

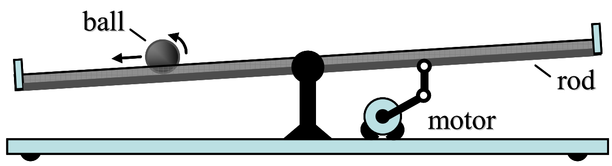


Fig.1. Scheme of model 'ball on rod'.

In general, such model represents mechatronic system of fourth order with one input and one output. At simplification, only dynamics of ball (system of second order) can be considered, since it is dominant in whole dynamics.
The paper begins by model definition (section 2) and explanation of on-line identification (section 3). The key part about generalized predictive control is in section 4. The paper ends by demonstration of tests on real laboratory model 'ball on rod' (section 5).

2 Model definition

The model, description of controlled system, represents very important part, which includes specifically processed information for design of control actions.
The best results of control process are achieved, when the model is obtained on the basis of thoroughgoing mathematical and physical analysis. It is often difficult. Therefore, different ways, how to obtain the model describing the controlled system, are investigated.
Selection of model form is determined by used model-based control, in which the model is involved. Due to digital character of automating devices, the discrete control techniques are preferred. Therefore, the resultant models for control design are also discrete in spite of the facts that controlled system may be continuous. Discrete realization is advantageous, because naturally respects finite and predefined time for computation of control actions.
As was mentioned in introduction, the design of algorithms of predictive control will arise from state-space formulation. However, only this idea will be used as inspiration for utilization of ARX models, which are Input/Output type. In the following subsections will be discussed both ARX model for Single-Input Single-Output (SISO) systems and two possible ARX model structures for Multi-Input Multi-Output (MIMO) systems. Two possibilities of model structures at MIMO systems follow from inclusion of different amount of a priori information to identification.

2.1 Model for single Input/Output systems

The SISO systems can be simply described by autoregressive model with external input (ARX model) [3], which describes relations just among inputs and outputs. Let us proceed from it.

$$y(k)=\sum_{i=0}^{n} b_i\, u(k-i)-\sum_{i=1}^{n} a_i\, y(k-i)+e(k) \qquad (1)$$

In (1), n is order of controlled system; $y(\cdot)$ and $u(\cdot)$ are values of its output and input; and $e(k)$ is error, respective, some noise of measurement of system output $y(k)$. The ARX model can be also written in the following condensed forms; either in row oriented form of parameters ϑ_k

$$y(k)=\vartheta_k\, \mathbf{f}_k+e(k) \qquad (2)$$

or in column oriented form of parameters $\theta_k=\vartheta_k^T$

$$y(k)=\mathbf{f}_k^T\, \theta_k+e(k) \qquad (3)$$

Vector of ARX parameters ϑ_k $(=\theta_k^T)$ is composed as follows

$$\vartheta_k=[b_0\ b_1\ \cdots\ b_n\ -a_1\ -a_2\ \cdots\ -a_n] \qquad (4)$$

and data vector is composed from sequence of delayed values of used control actions (inputs) and measured outputs

$$\mathbf{f}_k=[u(k)\ u(k-1)\cdots u(k-n)\ y(k-1)\cdots y(k-n)]^T \qquad (5)$$

The both forms (2) and (3) are identical and their use depends on user. In next subsection dealing with Input/Output models for MIMO systems, the situation is a little different.

2.2 Model for multi Input/Output systems

In case of models for MIMO systems, let us proceed also from ARX model (1), but let us consider MIMO character of given system

$$\mathbf{y}(k)=\sum_{i=0}^{n}\mathbf{B}_i\,\mathbf{u}(k-i)-\sum_{i=1}^{n}\mathbf{A}_i\,\mathbf{y}(k-i)+\mathbf{e}(k) \qquad (6)$$

where n is still order of controlled system; $\mathbf{y}(\cdot)$ and $\mathbf{u}(\cdot)$ are vectors of values of its outputs and inputs as follows

$$\mathbf{y}(k-i)=\begin{bmatrix} y_1(k-i)\\ \vdots \\ y_{ny}(k-i)\end{bmatrix},\quad \mathbf{u}(k-i)=\begin{bmatrix} u_1(k-i)\\ \vdots \\ u_{nu}(k-i)\end{bmatrix} \qquad (7)$$

and $\mathbf{e}(k)$ is an ny dimensional error vector corresponding to noise of measurement of output $\mathbf{y}(k)$. The parameters are included in matrices $\mathbf{B}_i$ and $\mathbf{A}_i$

$$\mathbf{B}_i=\begin{bmatrix} b_i^{11} & \cdots & b_i^{1\,nu}\\ \vdots & \ddots & \vdots\\ b_i^{ny\,1} & \cdots & b_i^{ny\,nu}\end{bmatrix},\quad \mathbf{A}_i=\begin{bmatrix} a_i^{11} & \cdots & a_i^{1\,ny}\\ \vdots & \ddots & \vdots\\ a_i^{ny\,1} & \cdots & a_i^{ny\,ny}\end{bmatrix} \qquad (8)$$

The model (6) can be rewritten for identification in two possible forms [2], which differ in properties. One of them is a multivariate linear regression form

$$\begin{aligned}\mathbf{y}(k)&=\left[\mathbf{B}_0\cdots\mathbf{B}_n\ \mathbf{A}_1\cdots\mathbf{A}_n\right]\mathbf{f}_k+\mathbf{e}(k)\\ \mathbf{y}(k)&=\qquad\qquad \vartheta_k\qquad\qquad \mathbf{f}_k+\mathbf{e}(k)\end{aligned} \qquad (9)$$

where individual coefficients (parameters) are included in horizontal rectangular matrix ϑ_k; and $\mathbf{f}_k$ is

$$\mathbf{f}_k=[\mathbf{u}^T(k)\cdots\mathbf{u}^T(k-n)\ \mathbf{y}^T(k-1)\cdots\mathbf{y}^T(k-n)]^T \qquad (10)$$

[3]; and the second is a full polynomial form [2]

$$\mathbf{y}(k)=\mathfrak{F}_k^T\,\theta_k+\mathbf{e}(k) \qquad (11)$$

where individual parameters are situated in column vector θ_k and the data are in horizontal rectangular matrix $\mathfrak{F}_k^T$. Their internal structures can be different.

Let us specify these internal structures:

$$\mathfrak{F}_k^T = \begin{bmatrix} \mathbf{f}_k^T & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{f}_k^T \end{bmatrix}, \; \mathfrak{F}_k^T \in \Re^{ny \times n(nu \cdot ny + ny \cdot ny)} \qquad (12)$$

$$\boldsymbol{\theta}_k = \begin{bmatrix} \theta_k^1 \\ \vdots \\ \theta_k^{ny} \end{bmatrix}, \left(\begin{bmatrix} (\theta_k^1)^T \\ \vdots \\ (\theta_k^{ny})^T \end{bmatrix} = \left[\mathbf{B}_0 \cdots \mathbf{B}_n \; \mathbf{A}_1 \cdots \mathbf{A}_n \right] \right)$$
$$\theta_k^1 = \left[b_0^{11} \cdot\cdot\, b_0^{1\,nu} \cdot\cdot\, b_n^{11} \cdot\cdot\, b_n^{1\,nu} \; a_1^{11} \cdot\cdot\, a_1^{1\,ny} \cdot\cdot\, a_n^{11} \cdot\cdot\, a_n^{1\,ny} \right]^T \qquad (13)$$
$$\vdots$$
$$\theta_k^{ny} = \left[b_0^{ny\,1} \;\; \cdots \;\; b_n^{ny\,nu} \; a_1^{ny\,1} \;\; \cdots \;\; a_n^{ny\,ny} \right]^T$$

The model (11) in view of structures (12) and (13) can be written separately for each system output

$$\begin{aligned} y_1(k) &= \mathbf{f}_k^T \, \theta_k^1 + e_1(k) \\ \vdots \quad & \quad \vdots \qquad\quad \vdots \\ y_{ny}(k) &= \mathbf{f}_k^T \, \theta_k^{ny} + e_{ny}(k) \end{aligned} \qquad (14)$$

This form means decomposition to set of equations expressing relation of individual current outputs to all appropriate inputs (current and delayed) and appropriate delayed outputs. Then, the identification can be realized formally as well as in SISO case.

The difference of both forms for multi Input/Output systems becomes evident in initialization of identification and influence of evolution of identified parameters by a priori information.

The ARX model (1) or (6) will be used for construction of equations of predictions of predictive design (subsection 4.1). The multi Input/Output character does not represent any obstacle only increase of computational demands.

Finally, the condensed forms (2) or (3), respectively, (9) or (14) are suitable for identification by least squares (section 3).

3 Identification

The sufficient and well known method of identification is method of least squares. In this paper will be briefly summed up in square-root form [2], [3].

For simplicity, let us consider ARX model (2), where $e(k)$ represents in view of least squares model error expressed as follows:

$$e(k) = y(k) - \vartheta_k \, \mathbf{f}_k \qquad (15)$$

The expression (15) is not sufficient for identification, since it is only one equation for determination of $2n+1$ (or $2n$, if $b_0 = 0$) unknown parameters ϑ_k.

Therefore, on the assumption, that the parameters are close to constants or they are varied only slightly during real control process, then it is possible to write needful number of equations of errors with changeless vector of parameters ϑ_k

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{F}_k \, \vartheta_k^T = [\mathbf{F}_k \;\; \mathbf{y}_k] \begin{bmatrix} -\vartheta_k^T \\ 1 \end{bmatrix} \qquad (16)$$

where $\mathbf{F}_k$ is a matrix of past data, of which rows are composed from data vectors $\mathbf{f}_i^T$, $i = 1, \cdots, 2n$.

The criterion for identification is

$$J_k = \mathbf{e}_k^T \, \mathbf{e}_k \qquad (17)$$

alternatively

$$J_k = [-\vartheta_k \;\; 1] \begin{bmatrix} \mathbf{F}_k^T \\ \mathbf{y}_k^T \end{bmatrix} [\mathbf{F}_k \;\; \mathbf{y}_k] \begin{bmatrix} -\vartheta_k^T \\ 1 \end{bmatrix} \qquad (18)$$

To minimize the criterion, it is sufficient to minimize only its square-root $\mathbf{J}$ following from (18)

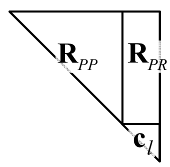
$$\min J_k = \| \mathbf{J} \|^2 = \left\| [\mathbf{F}_k \;\; \mathbf{y}_k] \begin{bmatrix} -\vartheta_k^T \\ 1 \end{bmatrix} \right\|^2 \qquad (19)$$

The computationally effective minimization is provided by orthogonal-triangular decomposition (e.g. house-holder algorithm [5])

$$\mathbf{Q} [\mathbf{F}_k \;\; \mathbf{y}_k] \begin{bmatrix} -\vartheta_k^T \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{c}_l \end{bmatrix} \qquad (20)$$

which transforms extended matrix $[\mathbf{F}_k \;\; \mathbf{y}_k]$ to upper triangular matrix.

$$\mathbf{Q} [\mathbf{F}_k \;\; \mathbf{y}_k] = \mathbf{R} = \qquad (21)$$



This matrix consists of sub-matrices partly corresponding to the unknown parameters and partly to square-root $\mathbf{c}_l$ of loss of the criterion.

By considering sub-matrices related to unknown parameters, the following equation is obtained

$$-\mathbf{R}_{PP} \vartheta_k^T + \mathbf{R}_{PR} = \mathbf{0} \qquad (22)$$

from which, the unknown parameters can be determined by backward substitution (due to triangular form of matrix $\mathbf{R}_{PP}$). This process is provided on-line in each time step with connecting refreshed data $\mathbf{f}_k$ and $y(k)$ to current of triangular matrix $\mathbf{R}$.

Thus appropriate part of matrix $\mathbf{R}$ is restored to new upper triangular matrix $\mathbf{R}_{new}$.

$$\mathbf{Q}\begin{bmatrix} \mathbf{R}_{PP}^{prev} & \mathbf{R}_{PR}^{prev} \\ \mathbf{f}_k^T & y(k) \end{bmatrix} = \mathbf{R}_{new} \tag{23}$$

The initial filling of matrix $\mathbf{R}$, when the identification of parameters should not start from zeros (a priori parameter setting) can be done as

$$\begin{array}{l} \mathbf{R} = K\mathbf{I}_{(2n+n_y)},\ approx.\ K = 10^{-6},\ n_y \text{ - number of y} \\ \mathbf{R}_{(1:2n,2n+1:2n+n_y)} = \mathbf{R}_{(1:2n,1:2n)}\vartheta_0^T\ (= \mathbf{R}_{PR} = \mathbf{R}_{PP}\vartheta_0^T) \end{array} \tag{24}$$

This selection (diagonal elements of $\mathbf{R}$) influence progress of evolution of identified parameters. In case of parameters of SISO systems and in case parameterization (14) for MIMO systems these diagonal elements correspond to individual parameters. By setting of some element to zero means that appropriate parameter is fixed, i.e. it keeps its initial value. This property is useful e.g. for a priori setting of the dependency of individual outputs on inputs and other outputs; i.e. presence of the appropriate parameter or not. The parameterization in multivariate linear regression form has limited this property, but on the other hand it is more computationally effective.

During the process of identification, to increase the weight of the newest data, the addition of exponential forgetting factor $fi_{(=0.9-1)}$ $\mathbf{R} = fi\,\mathbf{R}$ is useful. It is realized after obtaining current parameters.

4 Algorithm of predictive control

Predictive control is a multi-step approach, combining feedforward and feedback control design [1]. Feedforward is represented by predictions based on mathematical model. This part is dominant component of control actions. Feedback from measured outputs serves for compensation of some bounded model inaccuracies and low external disturbances.

The design consists in local minimization of quadratic criterion, in which the predictions of future outputs $y(\cdot)$ are involved. The predictions are determined from specific equations of predictions forming the basis of predictive control. The minimization is repeated in each time step.

The section firstly focuses on reorganization of ARX model (Input/Output formulation) in order to use the idea of effective state-space formulation. Then, the reorganized model is used in the construction of equations of predictions. Finally, the efficient square-root criterion minimization is explained.

4.1 Model reorganization

To compose equations of predictions from available ARX model (defined and identified in sections 2 and 3), there are several possibilities how to do it. One of the possibilities is to express the equations directly from ARX model. It is possible according to [4], however such way requires solution of Diofantic equation and storing previous values of inputs and outputs as in ‘pseudo state-space’ possibilities. These further possibilities are generally called as state-space forms with non-minimal state. They will be helpful for the use of idea of state-space formulation in predictive control, in which forming needed equations of predictions consists in repetitive insertion of state-space model.

Let us firstly renew the ARX model: let us suppose, that parameters are stabilized or slightly changed around some constant values, further, that the mean values of stochastic components are zeros and that the controller reacts immediately against the model of controlled system, in which one time period delay is assumed as indicated in (25)

$$y(k) = \sum_{i=0}^{n} b_i\,u(k-i) - \sum_{i=1}^{n} a_i\,y(k-i),\ b_0 = 0$$

$$i.e.\ \ y(k) = \sum_{i=1}^{n} b_i\,u(k-i) - \sum_{i=1}^{n} a_i\,y(k-i) \tag{25}$$

Usual state-space form with non-minimal state corresponding to ARX model (25) is given in this way

$$\begin{bmatrix} u(k) \\ \vdots \\ u(k-n+2) \\ y(k+1) \\ y(k) \\ \vdots \\ y(k-n+2) \end{bmatrix} = \left[\begin{array}{ccc|c|ccc} 0 & & 0 & 0 & 0 & \cdots & 0 \\ \ddots & & & \vdots & \vdots & \ddots & \vdots \\ & & 1 & 0 & 0 & \cdots & 0 \\ \hline b_2 & \cdots & b_n & -a_1 & & \cdots & -a_n \\ \hline 0 & \cdots & 0 & 1 & & & 0 \\ \vdots & \ddots & \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & & & 1 & 0 \end{array}\right] \times \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-n+1) \\ y(k) \\ y(k-1) \\ \vdots \\ y(k-n+1) \end{bmatrix} + \begin{bmatrix} 1 \\ \vdots \\ 0 \\ b_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(k),\quad \mathbf{X}(k) = \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-n+1) \\ y(k) \\ y(k-1) \\ \vdots \\ y(k-n+1) \end{bmatrix}$$

$$\text{i.e.}\qquad \mathbf{X}(k+1) = \mathbf{A}\,\mathbf{X}(k) + \mathbf{B}\,u(k) \tag{26}$$

The state-space form (26) is complemented moreover by output equation (27)

$$
\begin{aligned}
y(k) &= [0\;0 \quad \cdots 0\;1 \quad \cdots 0]\,\mathbf{X}(k) \\
y(k) &= \qquad\quad \mathbf{C} \qquad\quad\; \mathbf{X}(k)
\end{aligned}
\tag{27}
$$

This form is standardly used for LQ adaptive control [3]. However, for predictive design, this form causes perceptible increase of matrix dimensions in equations of predictions.
To compose suitable form for equations of predictions, let us arise from ARX model in one-ahead predictive style.

$$
y(k+1) = \sum_{i=1}^{n} b_i\, u(k-i+1) - \sum_{i=1}^{n} a_i\, y(k-i+1) \tag{28}
$$

(Note: The ARX model (6) is possible to transform to this style and then to work with it considering matrices of parameters instead of scalar parameters.)

Then the suitable form can be structured as follows

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

$$
\begin{aligned}
\begin{bmatrix} y(k-n+2) \\ \vdots \\ y(k+1) \end{bmatrix}
&= \mathbf{A}
\begin{bmatrix} y(k-n+1) \\ \vdots \\ y(k) \end{bmatrix}
+ \mathbf{B}_0
\begin{bmatrix} u(k-n+1) \\ \vdots \\ u(k) \end{bmatrix} \\
\mathbf{X}(k+1) \quad &= \mathbf{A} \qquad \mathbf{X}(k) \qquad + \mathbf{B}_0 \qquad \mathbf{u}(k)
\end{aligned}
\tag{29}
$$

$$
\begin{aligned}
y(k) &= [0 \quad \cdots \quad 0\;\;1]\,\mathbf{X}(k) \\
y(k) &= \qquad\quad \mathbf{C} \qquad\quad \mathbf{X}(k)
\end{aligned}
\tag{30}
$$

State-space model (equations (29) and (30)) is equivalent to model (equations (26) and (27)), however thereby, that it was decomposed in two smaller pseudo state-space matrices $\mathbf{A}$ and $\mathbf{B}_0$ causes similar dimensions of matrices of equations of prediction as in the use of usual pure state-space models. Subscript of matrix $\mathbf{B}_0$ will be significant in real composition of equations of prediction in next subsection.

4.2 Equations of predictions

Usual composition of equations of predictions follows from ordinary state-space model

$$
\begin{aligned}
\mathbf{X}(k+1) &= \mathbf{A}_k\,\mathbf{X}(k) + \mathbf{B}_k\,\mathbf{u}(k) \\
\mathbf{y}(k) &= \mathbf{C}_k\,\mathbf{X}(k) + \mathbf{D}_k\,\mathbf{u}(k) \qquad (\mathbf{D}_k = \mathbf{0})
\end{aligned}
\tag{31}
$$

which is a model with minimal state. It maps interval of one sampling period. Principle of the equations is expression (prediction) of future values of outputs $\mathbf{y}$ from current measured state $\mathbf{X}(k)$ as follows [1]:

$$
\begin{aligned}
\hat{\mathbf{X}}(k+1) &= \mathbf{A}\;\mathbf{X}(k) + \mathbf{B}\mathbf{u}(k) \\
\hat{\mathbf{y}}(k+1) &= \mathbf{CA}\;\mathbf{X}(k) + \mathbf{C}\,\mathbf{B}\mathbf{u}(k) \\
&\vdots \qquad\qquad \vdots \qquad\qquad \ddots \\
\hat{\mathbf{X}}(k+N) &= \mathbf{A}^N\mathbf{X}(k) + \mathbf{A}^{N-1}\mathbf{B}\mathbf{u}(k) \cdots + \mathbf{B}\mathbf{u}(k+N-1) \\
\hat{\mathbf{y}}(k+N) &= \mathbf{CA}^N\mathbf{X}(k) + \mathbf{CA}^{N-1}\mathbf{B}\mathbf{u}(k) \cdots + \mathbf{CB}\mathbf{u}(k+N-1)
\end{aligned}
\tag{32}
$$

Equation (32) can be condensed in matrix notation

$$
\begin{aligned}
\hat{\mathbf{y}} = \mathbf{f} + \mathbf{G}\,\mathbf{u},\;\; \hat{\mathbf{y}} &= [\hat{\mathbf{y}}(k+1)\;\hat{\mathbf{y}}(k+2)\cdots\hat{\mathbf{y}}(k+N)]^T \\
\mathbf{u} &= [\mathbf{u}(k)\;\mathbf{u}(k+1)\cdots\mathbf{u}(k+N-1)]^T
\end{aligned}
\tag{33}
$$

$$
\mathbf{f} = \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^N \end{bmatrix} \mathbf{X}(k)
$$

$$
\mathbf{G} = \begin{bmatrix} \mathbf{C} & \mathbf{B} & \\ \vdots & & \ddots \\ \mathbf{CA}^{N-1} & \mathbf{B} \cdots & \mathbf{CB} \end{bmatrix}
\tag{34}
$$

Considering the state-space model (29) with output equation (30), the equations of prediction can be composed according to similar idea as was indicated by (32)

$$
\begin{aligned}
»\hat{\mathbf{X}}_{(k+1)} &= \mathbf{A}\mathbf{X}_{(k)} + \mathbf{B}_0\mathbf{u}_{(k)} \\
\hat{\mathbf{X}}_{(k+2)} &= \mathbf{A}^2\mathbf{X}_{(k)} + \mathbf{AB}_0\mathbf{u}_{(k)} + \mathbf{B}_0\bar{\mathbf{u}}_{(k+1)} \\
»\hat{\mathbf{X}}_{(k+2)} &= \mathbf{A}^2\mathbf{X}_{(k)} + \underbrace{([\mathbf{AB}_0\;\mathbf{0}] + [\mathbf{0}\;\mathbf{B}_0])}_{\mathbf{B}_1}\,\mathbf{u}_{(k+1)} \\
\hat{\mathbf{X}}_{(k+3)} &= \mathbf{A}^3\mathbf{X}_{(k)} + \mathbf{AB}_1\mathbf{u}_{(k+1)} + \mathbf{B}_0\bar{\mathbf{u}}_{(k+2)} \\
»\hat{\mathbf{X}}_{(k+3)} &= \mathbf{A}^3\mathbf{X}_{(k)} + \underbrace{([\mathbf{AB}_1\;\mathbf{0}] + [\mathbf{0}\;\mathbf{0}\;\mathbf{B}_0])}_{\mathbf{B}_2}\,\mathbf{u}_{(k+2)} \\
&\ddots \\
\hat{\mathbf{X}}_{(k+N)} &= \mathbf{A}^N\mathbf{X}_{(k)} + \mathbf{AB}_{N-2}\mathbf{u}_{(k+N-2)} + \mathbf{B}_0\bar{\mathbf{u}}_{(k+N-1)} \\
»\hat{\mathbf{X}}_{(k+N)} &= \mathbf{A}^N\mathbf{X}_{(k)} + \underbrace{([\mathbf{AB}_{N-2}\;\mathbf{0}] + [\mathbf{0}_{n,N-1}\;\mathbf{B}_0])}_{\mathbf{B}_{N-1}}\,\mathbf{u}_{(k+N-1)}
\end{aligned}
\tag{35}
$$

The support notation has the following meaning

$$\begin{aligned}
\bar{\mathbf{u}}_{(k+1)} &= [u_{(k-n+2)} \cdots u_{(k+1)}]^T \\
\mathbf{u}_{(k+1)} &= [u_{(k-n+1)}\, u_{(k-n+2)} \cdots u_{(k)}\, u_{(k+1)}]^T \\
&= [\mathbf{u}^T_{(k)}, u_{(k+1)}]^T \\
&\vdots \\
\bar{\mathbf{u}}_{(k+N-1)} &= [u_{(k-n+N)} \cdots u_{(k+N-1)}]^T \\
\mathbf{u}_{(k+N-1)} &= [u_{(k-n+1)} \cdots u_{(k+N-2)}\, u_{(k+N-1)}]^T \\
&= [\mathbf{u}^T_{(k+N-2)}, u_{(k+N-1)}]^T
\end{aligned} \tag{36}$$

The equations of predictions (35) can be also rewritten to appropriate matrix notation:

$$\begin{aligned}
\begin{bmatrix} \hat{y}_{(k+1)} \\ \vdots \\ \hat{y}_{(k+N)} \end{bmatrix} &= \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^N \end{bmatrix} \begin{bmatrix} y_{(k-n+1)} \\ \vdots \\ y_{(k)} \end{bmatrix} + \begin{bmatrix} \mathbf{CB}_0 & \cdots & \mathbf{0} \\ & \ddots & \vdots \\ & \mathbf{CB}_{N-1} & \end{bmatrix} \begin{bmatrix} u_{(k-n+1)} \\ \vdots \\ u_{(k+N-1)} \end{bmatrix} \\
\hat{\mathbf{y}} \quad &= \quad \mathbf{f} \quad + \quad \bar{\mathbf{G}} \quad \mathbf{u}_{(k+N+1)} \\
\hat{\mathbf{y}} \quad &= \mathbf{f} + \bar{\mathbf{G}}_{(:,1:n-1)} \begin{bmatrix} u_{(k-n+1)} \\ \vdots \\ u_{(k-1)} \end{bmatrix} + \bar{\mathbf{G}}_{(:,n:N-1)} \begin{bmatrix} u_{(k)} \\ \vdots \\ u_{(k+N-1)} \end{bmatrix} \\
\hat{\mathbf{y}} \quad &= \quad \bar{\mathbf{f}} \quad + \quad \mathbf{G} \quad \mathbf{u}
\end{aligned} \tag{37}$$

representing: free response + forced response. Such composed equations of predictions have the same dimension as the equations (34), which are based on state-space model with minimal state.

4.3 Square-root minimization

The control actions are obtained by minimization of quadratic criterion

$$J_k = \sum_{j=No+1}^{N} \left\| (\hat{y}_{(k+j)} - w_{(k+j)}) Q_y \right\|^2 + \sum_{j=1}^{Nu} \left\| u_{(k+j-1)} Q_u \right\|^2 \tag{38}$$

where N, No and Nu are horizons; Q_y and Q_u are penalizations; and $w(k+j)$ are desired values. That criterion can be again condensed in matrix notation

$$J_k = [(\hat{\mathbf{y}} - \mathbf{w})^T, \mathbf{u}^T] \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix}^T \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ \mathbf{u} \end{bmatrix} \tag{39}$$

from which, only one part (square-root) is sufficient to minimize [6].

$$\mathbf{J} = \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{Q_y\, G} \\ \mathbf{Q_u} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{Q_y}(\mathbf{w} - \bar{\mathbf{f}}) \\ \mathbf{0} \end{bmatrix} \tag{40}$$

The minimization leads to the solution of algebraic equations for unknown control actions

$$\begin{aligned}
\begin{bmatrix} \mathbf{Q_y\, G} \\ \mathbf{Q_u} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{Q_y}(\mathbf{w} - \bar{\mathbf{f}}) \\ \mathbf{0} \end{bmatrix} &= \mathbf{0} \\
\mathbf{A} \quad \mathbf{u} - \quad \mathbf{b} \quad &= \mathbf{0}
\end{aligned} \tag{41}$$

This system of algebraic equations can be effective evaluated by orthogonal-triangular decomposition (e.g. using house-holder algorithm [5]).

$$\begin{aligned}
\mathbf{A}\,\mathbf{u} &= \mathbf{b} \quad / \times \mathbf{Q}^T \\
\mathbf{R}_1 \mathbf{u} &= \mathbf{c}_1
\end{aligned} \tag{42}$$

Orthogonal matrix $\mathbf{Q}^T$ transforms the system matrix $\mathbf{A}$ to upper triangle $\mathbf{R}_1$. Unknown control actions from the algebraic system (42) can be determined by backward substitution.

Finally, from obtained vector $\mathbf{u}$, which represents designed control actions for whole horizon N, only first appropriate actions are really applied to controlled system. This process is repeated in every time step.

5 Tests with model 'ball on rod'

For real-time tests, simple laboratory model (Fig.2) was used. From mathematical-physical analysis, this model represents system of fourth order: electrical motor is second order and the ball dynamics is expressed also by second order.

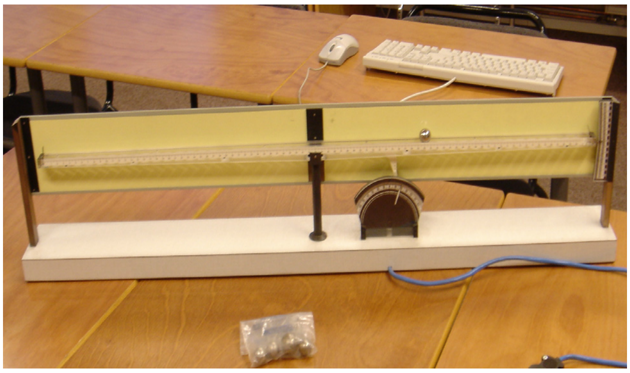


Fig.2. Laboratory model 'ball on rod'.

For simplicity, the dynamics of the motor can be omitted, because it is negligible against dynamics of the ball (in view of time constant versus sampling period used for computation of control actions). The model of the ball can be obtained according to force diagram see Fig.3.

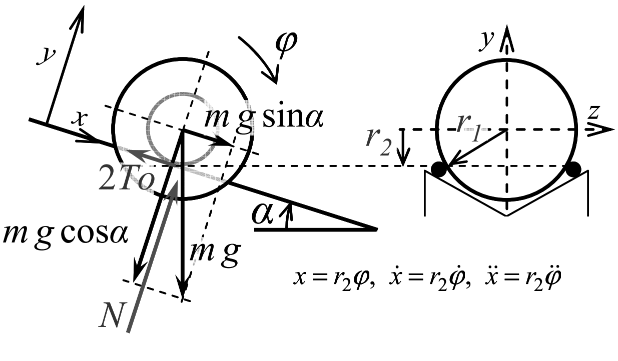


Fig.3. Force diagram of ball on rod.

From the diagram (Fig.3) the following differential equation follows

$$\ddot{x} = \frac{g}{1+\frac{2}{5}\left(\frac{r_1}{r_2}\right)^2}\sin\alpha \qquad (43)$$

This equation represents pure equation of motion. It can be furthermore simplified, when it is supposed, that angle α fulfils $-5° < \alpha < 5°$ then $\sin\alpha \approx \alpha$ and the equation becomes linear

$$\ddot{x} = k\,\alpha \qquad (44)$$

The equation (44) can be assumed as a suitably simplified model for control design due to negligible time constant of the motor against the constant of the ball. Moreover let us note that the model is independent of weight of ball.

The real laboratory model was controlled by Real Time Toolbox for MATLAB via M-file S-functions Level 2 of the predictive controller and algorithm of identification implemented in Simulink blocks; see schemes in Fig.4 and Fig.7.

The presented predictive control was partly tested in adaptive mode using algorithm of on-line identification (Fig.4).

For comparison, the control was partly tested in non-adaptive mode (Fig.7) with constant model obtained from introduced simple mathematical-physical analysis (Fig.3 and equations (43), (44)).

The aim of the tests was stabilization of ball in different positions x of desired rectangular signal w with frequency $0.04 s^{-1}$ and amplitude $0.4 m$.

The control process with model identification requires relatively energetic control actions for induction of identification. It is achieved by penalization Q_u, which is smaller, than in case of using physical model (compare Fig.5 and Fig.8). Therefore the control actions are quite jittered.

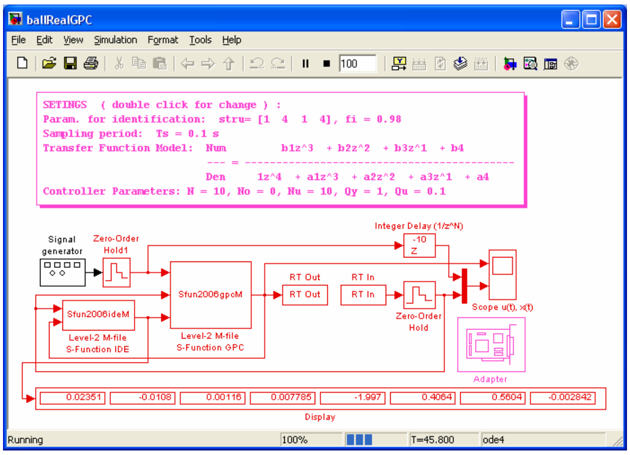


Fig.4. Simulink scheme of adaptive predictive control.

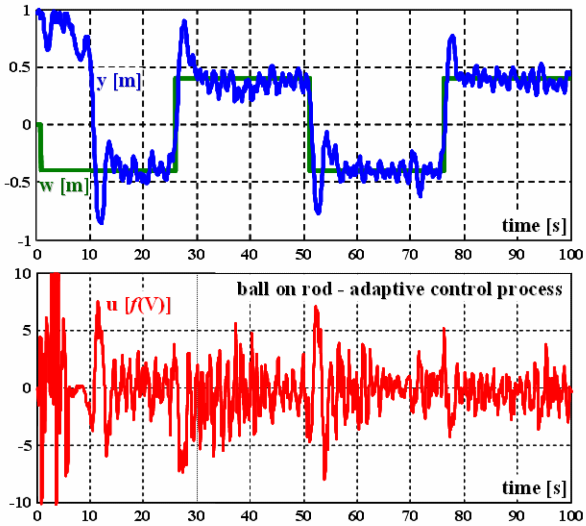


Fig.5. Time histories of adaptive predictive control.

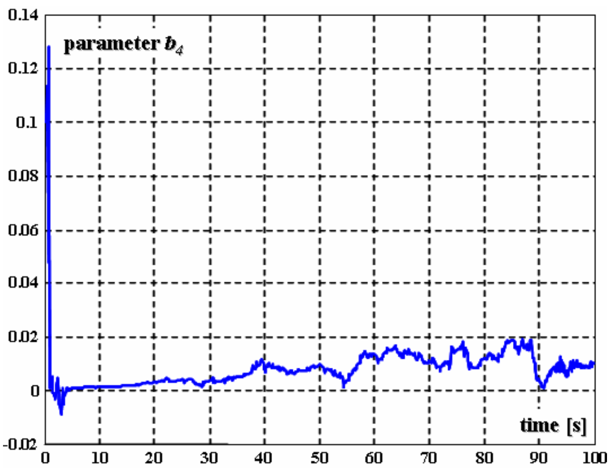


Fig.6. Example of parameter evolution (parameter b_4 corresponding to input $u(k-4)$ in ARX model (1) ($b_0 = 0$), i.e. model (25)).

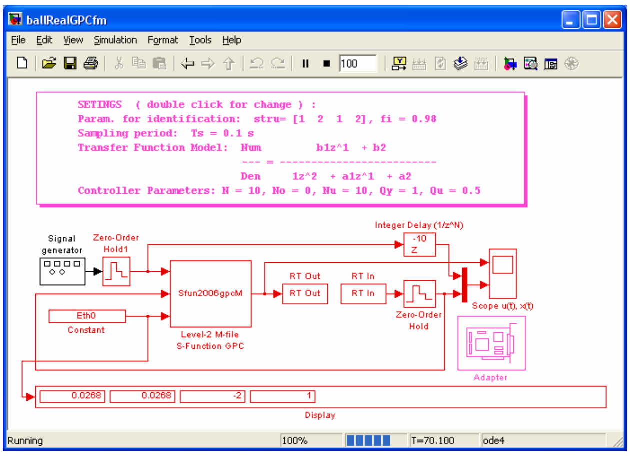


Fig.7. Simulink scheme of predictive control with mathematical-physical model.

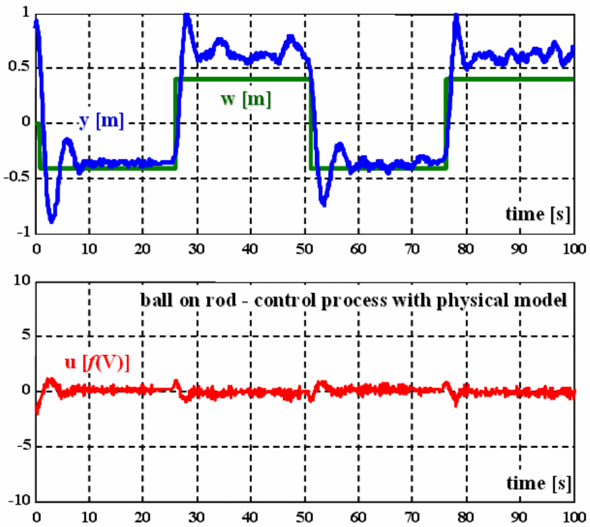


Fig.8. Time histories of predictive control with mathematical-physical model.

The Fig.8 shows positive results, however the predictive controller with mathematical model can not compensate inaccuracies given by real conditions. Therefore, in spite of integral character of the system 'ball on rod', the control error does not tend to zero or at least to error with zero mean as in case of adaptive predictive control.

The real conditions for model in Fig.2, which was used for real experiments, were determined by asymmetric connection of drive unit. This asymmetry causes different sensitivity of the rod dynamics to actuator, when the ball is located on the right-side or let-side of the rod. Such changes in sensitivity are difficult to be covered by mathematical-physical model. Therefore predictive control in adaptive mode achieves better results but it generates more jittered control actions.

6 Conclusion

In the paper, one possible solution of adaptive predictive controller was shown. Input/Output ARX model was used for composition of equations of predictions in specific pseudo state-space formulation. This formulation requires comparable matrices of the same dimensions in spite of non-minimal state; i.e. it has similar computational demands as usual state-space formulation with minimal system state. The real experiments with laboratory model 'ball on rod' illustrate differences between adaptive and non-adaptive mode of generalized predictive control.

References:

[1] Ordis, A., Clarke, D.: A state – Space Description for GPC Controllers. *INT. J. Systems SCI.*, Vol. 24, No. 9, 1993, pp. 1727-1744.

[2] Söderström, T., Stoica, P.: *System Identification*, Prentice Hall, 1989, 612 p.

[3] Bobál, V., Böhm, J. et al.: *Digital Self-tuning Controllers Algorithms, implementation & applications,* Springer, 2005, 318 p.

[4] Maciejowski, J. M.: *Predictive Control with Constraints*. Prentice Hall, London 2002.

[5] Golub, H. G., Loan, Van Ch. F.: *Matrix computations.* The Johns Hopkins Univ. Press, 1989.

[6] Belda, K., Böhm, J., Píša, P.: Predictive control for modern industrial robots. *Proc. of Int. Conf. on Informatics in Control, Automation and Robotics*, INSTICC Press, 2005, pp. 1-8.

Acknowledgments

The authors appreciates kind support of Grant Agency of the Czech Republic by the grant (102/06/P275, 2006/08) 'Model-based Control of Mechatronic Systems for Robotics' and (102/05/0271, 2005/07) 'Methods of Predictive Control, Algorithms and Implementation'.