Predictive Control with Filtered Input and Output Variables in Prediction Equations

MAREK KUBALČÍK, VLADIMÍR BOBÁL
Tomas Bata University in Zlín
Department of Process Control
Nad Stráněmi 4511, 76005, Zlín
CZECH REPUBLIC
kubalcik@fai.utb.cz, bobal@fai.utb.cz, http://web.fai.utb.cz/

Abstract: - The paper is focused on an implementation of a predictive controller with a colouring filter *C* in a disturbance model. Both single-input-single-output (SISO) and multi-input-multi-output (MIMO) cases were considered and analysed. The filter is often essential for practical applications of predictive control based on input-output models. It is commonly considered as a design parameter because it has direct effects on closed loop performance. In this paper a computation of predictions for the case with the colouring polynomial is introduced. The computation is based on particular models of the controlled systems in the form of transfer function in case of SISO system and matrix fraction in case of MIMO system which are commonly used for description of a range of processes. Performances of closed loop systems with and without the colouring polynomial in the disturbance model were also compared.

Key-Words: - predictive control, disturbance modelling, colouring polynomial, filtering of variables, transfer function models, matrix fraction models

1 Introduction

Model Predictive Control (MPC) or only Predictive Control [1], [2], [3] is one of the control methods which have developed considerably over a few past years. Predictive control is essentially based on discrete or sampled models of processes. Computation of appropriate control algorithms is then realized especially in the discrete domain. When using most of other approaches, the control actions are taken based on past errors. MPC uses also future values of the reference signals. The basic idea of the generalized predictive control [4], [5] is to use a model of a controlled process to predict a number of future outputs of the process. A trajectory of future manipulated variables is given by solving an optimization problem incorporating a suitable cost function and constraints. Only the first element of the obtained control sequence is applied. The whole procedure is repeated in following sampling period. This principle is known as the receding horizon strategy.

Typical technological processes require the simultaneous control of several variables related to one system. Each input may influence all system outputs. The design of a controller for such a system must be quite sophisticated if the system is to be controlled adequately. Simple decentralized PI or PID controllers largely do not yield satisfactory results. There are many different advanced methods of controlling multi-input–multi-output (MIMO) systems. The problem of selecting an appropriate control technique often arises. Perhaps the most popular way of controlling MIMO processes is by designing decoupling compensators to suppress the interactions [6] and the designing multiple SISO controllers [7]. This requires determining how to pair the controlled and manipulated variables. One of the most effective approaches to control of multivariable systems is model predictive control. An advantage of model predictive control is that multivariable systems can be handled in a straightforward manner.

Implementation of predictive controllers based on input-output models with a colouring filter *C* in a disturbance model is described in this paper. Both single-input-single-output (SISO) and multi-input-multi-output (MIMO) cases were considered and analysed. The filter is often essential for practical applications of predictive control based on input-output models. Surveys of practical applications of predictive control are presented in [8], [9], [10]. It is commonly considered as a design parameter because it has direct effects on closed loop performance. A computation of predictions for the case with the colouring polynomial is introduced. The computation is based on particular models of the controlled systems in the form of transfer function in case of SISO system and matrix fraction in case of MIMO system which are commonly used for description of a range of processes. The filtering of variables is the equivalent of the colouring polynomial in the noise model. It is practically very difficult to estimate the coefficients of the colouring polynomial. A model with the *C*-polynomial is then utilized as an example with filtering of input and output variables when the polynomial *C* is a tuning parameter. In the paper are derived prediction equations for both SISO and MIMO input-output models both for the case with the *C*-filter and without the *C*-filter. Performances of closed loop systems with and without the colouring polynomial in the disturbance model were also compared.

2 Model of the Controlled System

2.1 Model of SISO System

A model of the second order which is widely used in practice and has proved to be effective for control of a range of various processes was applied. It can be expressed by following transfer function

$$G\left(z^{-1}\right)=\frac{B\left(z^{-1}\right)}{A\left(z^{-1}\right)}=\frac{b_1 z^{-1}+b_2 z^{-2}}{1+a_1 z^{-1}+a_2 z^{-2}} \tag{1}$$

A widely used model in general model predictive control is the CARIMA (controller autoregressive integrated moving average) model which we can obtain by adding a disturbance model as

$$A\left(z^{-1}\right)y(k)=B\left(z^{-1}\right)u(k)+\frac{C\left(z^{-1}\right)}{\Delta}n(k) \tag{2}$$

where n is a non-measurable random disturbance that is assumed to have zero mean value and constant covariance and $\Delta=1-z^{-1}$. C is the colouring polynomial. For purpose of simplification it is often supposed to be equal to 1[1]. In Model Predictive Control it is also common to treat C as a design parameter [4], [5], [13]. In this paper will be compared cases when C=1 and when C is supposed as the design parameter.

2.2 Model of MIMO System

Let us consider a two input – two output system. The two – input/two – output (TITO) processes are the most often encountered multivariable processes in practice and many processes with inputs/outputs beyond two can be treated as several TITO subsystems [11].

A general transfer matrix of a two-input–two-output system with significant cross-coupling between the control loops is expressed as:

$$\boldsymbol{G}(z)=\begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix} \tag{3}$$

$$\boldsymbol{Y}(z)=\boldsymbol{G}(z)\boldsymbol{U}(z) \tag{4}$$

where $\boldsymbol{U}(z)$ and $\boldsymbol{Y}(z)$ are vectors of the manipulated variables and the controlled variables, respectively.

$$\boldsymbol{U}(z)=\left[u_1(z), u_2(z)\right]^T \ \boldsymbol{Y}(z)=\left[y_1(z), y_2(z)\right]^T \tag{5}$$

It may be assumed that the transfer matrix can be transcribed to the following form of the matrix fraction:

$$\boldsymbol{G}(z)=\boldsymbol{A}^{-1}\left(z^{-1}\right)\boldsymbol{B}\left(z^{-1}\right)=\boldsymbol{B}_1\left(z^{-1}\right)\boldsymbol{A}_1^{-1}\left(z^{-1}\right) \tag{6}$$

where the polynomial matrices $\boldsymbol{A}\in R_{22}\left[z^{-1}\right], \boldsymbol{B}\in R_{22}\left[z^{-1}\right]$ are the left coprime factorizations of matrix $\boldsymbol{G}(z)$ and the matrices $\boldsymbol{A}_1\in R_{22}\left[z^{-1}\right], \boldsymbol{B}_1\in R_{22}\left[z^{-1}\right]$ are the right coprime factorizations of $\boldsymbol{G}(z)$. The model can be also written in the form

$$\boldsymbol{A}\left(z^{-1}\right)\boldsymbol{Y}(z)=\boldsymbol{B}\left(z^{-1}\right)\boldsymbol{U}(z) \tag{7}$$

As an example a model with polynomials of second degree was chosen. This model proved to be effective for control of several TITO laboratory processes [12], where controllers based on a model with polynomials of the first degree failed. The model has sixteen parameters. The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are defined as follows

$$\boldsymbol{A}\left(z^{-1}\right)=\begin{bmatrix} 1+a_1 z^{-1}+a_2 z^{-2} & a_3 z^{-1}+a_4 z^{-2} \\ a_5 z^{-1}+a_6 z^{-2} & 1+a_7 z^{-1}+a_8 z^{-2} \end{bmatrix} \tag{8}$$

$$\boldsymbol{B}\left(z^{-1}\right)=\begin{bmatrix} b_1 z^{-1}+b_2 z^{-2} & b_3 z^{-1}+b_4 z^{-2} \\ b_5 z^{-1}+b_6 z^{-2} & b_7 z^{-1}+b_8 z^{-2} \end{bmatrix} \tag{9}$$

the CARIMA model in the MIMO case is as follows

$$\boldsymbol{A}\left(z^{-1}\right)\boldsymbol{y}(k)=\boldsymbol{B}\left(z^{-1}\right)\boldsymbol{u}(k)+\boldsymbol{C}\left(z^{-1}\right)\boldsymbol{\Delta}^{-1}\left(z^{-1}\right)\boldsymbol{n}(k) \tag{10}$$

where

$$\boldsymbol{\Delta}\left(z^{-1}\right)=\begin{bmatrix} 1-z^{-1} & 0 \\ 0 & 1-z^{-1} \end{bmatrix} \tag{11}$$

In case of TITO system. $\boldsymbol{C}$ is the colouring polynomial matrix. For purpose of simplification it is often supposed to be equal to the identity matrix [1]. In a single input – single output case we have a colouring polynomial C instead of the matrix $\boldsymbol{C}$. Analogically the polynomial matrix $\boldsymbol{C}$ could be expected as the design parameters in a multivariable case. Nevertheless considering the polynomial matrix $\boldsymbol{C}$ as the design parameter is computationally unsolvable and practically inapplicable. A simplified model when the non-measurable random disturbance was a scalar was then considered

$$\Delta \boldsymbol{A}(z^{-1})\boldsymbol{y}(k) = \boldsymbol{B}(z^{-1})\Delta \boldsymbol{u}(k) + C(z^{-1})n(k) \tag{12}$$

Further will be compared cases when $\boldsymbol{C}$ is the identity matrix and when the input and output variables are filtered with a colouring polynomial C which is supposed as the design parameter.

3 Implementation of predictive controller

The basic idea of MPC is to use a model of a controlled process to predict N future outputs of the process. A trajectory of future manipulated variables is given by solving an optimization problem incorporating a suitable cost function and constraints. Only the first element of the obtained control sequence is applied. The whole procedure is repeated in following sampling period. This principle is known as the receding horizon strategy. The computation of a control law of MPC is based on minimization of the following criterion

$$J(k) = \sum_{j=N_1}^{N} \boldsymbol{e}(k+j)^2 + \lambda \sum_{j=1}^{N_u} \Delta \boldsymbol{u}(k+j)^2 \tag{13}$$

where $\boldsymbol{e}(k+j)$ is a vector of predicted control errors, $\Delta\boldsymbol{u}(k+j)$ is a vector of future increments of the manipulated variable (for the system with two inputs and two outputs each vector has two elements), N is a length of the prediction horizon, N_u is a length of the control horizon and λ is a weighting factor of control increments.
A predictor in a vector form is given by

$$\hat{\boldsymbol{y}} = \boldsymbol{G}\Delta\boldsymbol{u} + \boldsymbol{y}_0 \tag{14}$$

where $\hat{y}$ is a vector of system predictions along the horizon of the length N, $\Delta\boldsymbol{u}$ is a vector of control increments, $\boldsymbol{y}_0$ is the free response vector. $\boldsymbol{G}$ is a matrix of the dynamics. It contains values of the step sequence. In SISO case it is given as

$$\boldsymbol{G} = \begin{bmatrix} g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ g_3 & g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_N & g_{N-1} & g_{N-2} & \cdots & g_{N-N_u+1} \end{bmatrix} \tag{15}$$

In TITO case the matrix $\boldsymbol{G}$ takes the following form

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_0 & 0 & \cdots & \cdots & 0 \\ \boldsymbol{G}_1 & \boldsymbol{G}_0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \boldsymbol{G}_0 & 0 \\ \boldsymbol{G}_{N-1} & \cdots & \cdots & \cdots & \boldsymbol{G}_0 \end{bmatrix} \tag{16}$$

where sub-matrices $\boldsymbol{G}_i$ have dimension 2x2 and contain values of the step sequence.
The criterion (12) can be written in a general vector form

$$J = (\hat{\boldsymbol{y}} - \boldsymbol{w})^T (\hat{\boldsymbol{y}} - \boldsymbol{w}) + \lambda \Delta\boldsymbol{u}^T \Delta\boldsymbol{u} \tag{17}$$

where $\boldsymbol{w}$ is a vector of the reference trajectory. The criterion can be modified using the expression (15) to

$$J = 2\boldsymbol{g}^T \Delta\boldsymbol{u} + \Delta\boldsymbol{u}^T \boldsymbol{H} \Delta\boldsymbol{u} \tag{18}$$

where the gradient $\boldsymbol{g}$ and the Hess matrix $\boldsymbol{H}$ are defined by following expressions

$$\boldsymbol{g}^T = \boldsymbol{G}^T (\boldsymbol{y}_0 - \boldsymbol{w}) \tag{19}$$

$$\boldsymbol{H} = \boldsymbol{G}^T \boldsymbol{G} + \lambda \boldsymbol{I} \tag{20}$$

Handling of constraints is one of main advantages of predictive control. General formulation of predictive control with constraints is then as follows

$$\min_{\Delta\boldsymbol{u}} 2\boldsymbol{g}^T \Delta\boldsymbol{u} + \Delta\boldsymbol{u}^T \boldsymbol{H} \Delta\boldsymbol{u} \tag{21}$$

owing to

$$\boldsymbol{A}\Delta\boldsymbol{u} \le \boldsymbol{b} \tag{22}$$

The inequality (22) expresses the constraints in a compact form.

4 Computation of predictions – unfiltered variables

4.1 Computation of predictions for SISO system

An important task in predictive control is computation of predictions for arbitrary prediction and control horizons.

The difference equation of the CARIMA model without the unknown term can be expressed as:

$$\begin{aligned} y(k) &= (1-a_1)y(k-1) + (a_1 - a_2)y(k-2) + \\ &+ a_2 y(k-3) + b_1 \Delta u(k-1) + b_2 \Delta u(k-2) \end{aligned} \tag{23}$$

It was necessary to directly compute three steps-ahead predictions in a straightforward way by establishing of previous predictions to later predictions. The model order defines that computation of one step-ahead prediction is based on the three past values of the system output.

$$\begin{aligned}
&\hat{y}(k+1) = (1-a_1)y(k) + (a_1 - a_2)y(k-1) + a_2 y(k-2) + \\
&+ b_1 \Delta u(k) + b_2 \Delta u(k-1) \\
&\hat{y}(k+2) = (1-a_1)\hat{y}(k+1) + (a_1 - a_2)y(k) + a_2 y(k-1) + \\
&+ b_1 \Delta u(k+1) + b_2 \Delta u(k) \\
&\hat{y}(k+3) = (1-a_1)\hat{y}(k+2) + (a_1 - a_2)\hat{y}(k+1) + a_2 y(k) + \\
&+ b_1 \Delta u(k+2) + b_2 \Delta u(k+1)
\end{aligned} \tag{24}$$

The predictions after modification can be written in a matrix form

$$\begin{aligned}
&\begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \end{bmatrix} = \begin{bmatrix} g_1 & 0 \\ g_2 & g_1 \\ g_3 & g_2 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix} + \\
&+ \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \Delta u(k-1) + \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \end{bmatrix}
\end{aligned} \tag{25}$$

$$\begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \end{bmatrix} = \boldsymbol{G} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix} + \boldsymbol{P} \Delta u(k-1) + \boldsymbol{Q} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \end{bmatrix} \tag{26}$$

$$\begin{aligned}
&\boldsymbol{y}(k+1) = \boldsymbol{G} \Delta \boldsymbol{u}(k+j-1) + \boldsymbol{P} \Delta \boldsymbol{u}(k-1) + \boldsymbol{Q} \boldsymbol{y}(k+j-1) \\
&j \le N
\end{aligned} \tag{27}$$

where

$$\begin{aligned}
&\boldsymbol{G} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix} = \\
&= \begin{bmatrix} b_1 & 0 \\ b_1(1-a_1) + b_2 & b_1 \\ (a_1-a_2)b_1 + (1-a_1)^2 b_1 + (1-a_1)b_2 & (1-a_1)b_1 + b_2 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix}
\end{aligned} \tag{28}$$

$$\boldsymbol{P} \Delta u(k-1) = \begin{bmatrix} b_2 \\ (1-a_1)b_2 \\ (1-a_1)^2 b_2 + (a_1 - a_2) b_2 \end{bmatrix} \Delta u(k-1) \tag{29}$$

$$\begin{aligned}
&\boldsymbol{Q} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \end{bmatrix} = \\
&= \begin{bmatrix} (1-a_1) & (a_1-a_2) & a_2 \\ (1-a_1)^2 + (a_1-a_2) & (1-a_1)(a_1-a_2) + a_2 & a_2(1-a_1) \\ (1-a_1)^3 + 2(1-a_1)(a_1-a_2) + a_2 & (1-a_1)^2(a_1-a_2) + a_2(1-a_1) + (a_1-a_2)^2 & a_2(1-a_1)^2 + (a_1-a_2)a_2 \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \end{bmatrix}
\end{aligned} \tag{30}$$

The coefficients of the matrices ***G***, ***P*** and ***Q*** for further predictions are computed recursively. Based on the three previous predictions it is repeatedly computed the next row of the matrices ***P*** and ***Q*** in the following way:

$$p_4 = (1-a_1)p_3 + (a_1 - a_2)p_2 + a_2 p_1 \tag{31}$$

$$\begin{aligned}
q_{41} &= (1-a_1)q_{31} + (a_1 - a_2)q_{21} + a_2 q_{11} \\
q_{42} &= (1-a_1)q_{32} + (a_1 - a_2)q_{22} + a_2 q_{12} \\
q_{43} &= (1-a_1)q_{33} + (a_1 - a_2)q_{23} + a_2 q_{13}
\end{aligned} \tag{32}$$

The recursion of the matrix ***G*** is similar. The next element of the first column is repeatedly computed and the remaining columns are shifted. This procedure is performed repeatedly until the prediction horizon is achieved. If the control horizon is lower than the prediction horizon a number of columns in the matrix is reduced. Computation of a new element is performed as follows:

$$g_4 = (1-a_1)g_3 + (a_1 - a_2)g_2 + a_2 g_1 \tag{33}$$

4.2 Computation of predictions for MIMO System

The difference equation of the CARIMA model without the unknown term can be expressed as:

$$\begin{aligned}
&y_1(k+1) = (1-a_1)y_1(k) + (a_1 - a_2)y_1(k-1) + a_2 y_1(k-2) - \\
&- a_3 y_2(k) + (a_3 - a_4)y_2(k-1) + a_4 y_2(k-2) + \\
&+ b_1 \Delta u_1(k) + b_2 \Delta u_1(k-1) + b_3 \Delta u_2(k) + b_4 \Delta u_2(k-1) \\
&y_2(k+1) = (1-a_7)y_2(k) + (a_7 - a_8)y_2(k-1) + a_8 y_2(k-2) - \\
&- a_5 y_1(k) + (a_5 - a_6)y_1(k-1) + a_6 y_1(k-2) + \\
&+ b_5 \Delta u_1(k) + b_6 \Delta u_1(k-1) + b_7 \Delta u_2(k) + b_8 \Delta u_2(k-1)
\end{aligned} \tag{34}$$

These equations can be written into a matrix form

$$\begin{aligned}
&\boldsymbol{y}(k+1) = \boldsymbol{A}_1 \boldsymbol{y}(k) + \boldsymbol{A}_2 \boldsymbol{y}(k-1) + \boldsymbol{A}_3 \boldsymbol{y}(k-2) + \\
&+ \boldsymbol{B}_1 \Delta \boldsymbol{u}(k) + \boldsymbol{B}_2 \Delta \boldsymbol{u}(k-1)
\end{aligned} \tag{35}$$

where

$$\begin{aligned}
&\boldsymbol{A}_1 = \begin{bmatrix} 1-a_1 & -a_3 \\ -a_5 & 1-a_7 \end{bmatrix} \ \boldsymbol{A}_2 = \begin{bmatrix} a_1 - a_2 & a_3 - a_4 \\ a_5 - a_6 & a_7 - a_8 \end{bmatrix} \\
&\boldsymbol{A}_3 = \begin{bmatrix} a_2 & a_4 \\ a_6 & a_8 \end{bmatrix}
\end{aligned} \tag{36}$$

$$\boldsymbol{B}_1 = \begin{bmatrix} b_1 & b_3 \\ b_5 & b_7 \end{bmatrix} \ \boldsymbol{B}_2 = \begin{bmatrix} b_2 & b_4 \\ b_6 & b_8 \end{bmatrix} \tag{37}$$

It was again necessary to directly compute three steps-ahead predictions in a straightforward way by establishing of previous predictions to later predictions.

$$\begin{aligned}
&\hat{\boldsymbol{y}}(k+1) = \boldsymbol{A}_1 \boldsymbol{y}(k) + \boldsymbol{A}_2 \boldsymbol{y}(k-1) + \boldsymbol{A}_3 \boldsymbol{y}(k-2) + \\
&+ \boldsymbol{B}_1 \Delta \boldsymbol{u}(k) + \boldsymbol{B}_2 \Delta \boldsymbol{u}(k-1)
\end{aligned}$$

$$\hat{\boldsymbol{y}}(k+2) = \boldsymbol{A}_1 \boldsymbol{y}(k+1) + \boldsymbol{A}_2 \boldsymbol{y}(k) + \boldsymbol{A}_3 \boldsymbol{y}(k-1) + \\ + \boldsymbol{B}_1 \Delta \boldsymbol{u}(k+1) + \boldsymbol{B}_2 \Delta \boldsymbol{u}(k) \tag{38}$$

$$\hat{\boldsymbol{y}}(k+3) = \boldsymbol{A}_1 \boldsymbol{y}(k+2) + \boldsymbol{A}_2 \boldsymbol{y}(k+1) + \boldsymbol{A}_3 \boldsymbol{y}(k) + \\ + \boldsymbol{B}_1 \Delta \boldsymbol{u}(k+2) + \boldsymbol{B}_2 \Delta \boldsymbol{u}(k+1)$$

It is possible to divide computation of the predictions to recursion of the free response and recursion of the matrix of the dynamics. The free response vector can be expressed as:

$$\boldsymbol{y}_0 = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & q_{22} \\ \hline p_{31} & p_{32} \\ p_{41} & p_{42} \\ \hline p_{51} & p_{52} \\ p_{61} & p_{62} \end{bmatrix} \begin{bmatrix} \Delta u_1(k-1) \\ \Delta u_2(k-1) \end{bmatrix} + \left[\begin{array}{cc|cc|cc} q_{11} & q_{12} & q_{13} & q_{14} & q_{15} & q_{16} \\ q_{21} & q_{22} & q_{23} & q_{24} & q_{25} & q_{26} \\ \hline q_{31} & q_{32} & q_{33} & q_{34} & q_{35} & q_{36} \\ q_{41} & q_{42} & q_{43} & q_{44} & q_{45} & q_{46} \\ \hline q_{51} & q_{52} & q_{53} & q_{54} & q_{55} & q_{56} \\ q_{61} & q_{62} & q_{63} & q_{64} & q_{65} & q_{66} \end{array}\right] \begin{bmatrix} y_1(k) \\ y_2(k) \\ \hline y_1(k-1) \\ y_2(k-1) \\ \hline y_1(k-2) \\ y_2(k-2) \end{bmatrix} =$$

$$= \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \end{bmatrix} \Delta \boldsymbol{u}(k-1) + \begin{bmatrix} \boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} & \boldsymbol{Q}_{13} \\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22} & \boldsymbol{Q}_{23} \\ \boldsymbol{Q}_{31} & \boldsymbol{Q}_{32} & \boldsymbol{Q}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}(k) \\ \boldsymbol{y}(k-1) \\ \boldsymbol{y}(k-2) \end{bmatrix} = \boldsymbol{P} \Delta \boldsymbol{u}(k-1) + \boldsymbol{Q} \begin{bmatrix} \boldsymbol{y}(k) \\ \boldsymbol{y}(k-1) \\ \boldsymbol{y}(k-2) \end{bmatrix} \tag{39}$$

The coefficients of the matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ for further predictions are computed recursively. Based on the three previous predictions it is repeatedly computed the next row of the matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ in the following way:

$$\boldsymbol{P}_4 = \begin{bmatrix} p_{71} & p_{72} \\ p_{81} & p_{82} \end{bmatrix} = \boldsymbol{A}_1 \boldsymbol{P}_{31} + \boldsymbol{A}_2 \boldsymbol{P}_{21} + \boldsymbol{A}_3 \boldsymbol{P}_{11} \tag{40}$$

$$\boldsymbol{Q}_{41} = \begin{bmatrix} q_{71} & q_{72} \\ q_{81} & q_{82} \end{bmatrix} = \boldsymbol{A}_1 \boldsymbol{Q}_{31} + \boldsymbol{A}_2 \boldsymbol{Q}_{21} + \boldsymbol{A}_3 \boldsymbol{Q}_{11} \tag{41}$$

$$\boldsymbol{Q}_{42} = \begin{bmatrix} q_{73} & q_{74} \\ q_{83} & q_{84} \end{bmatrix} = \boldsymbol{A}_1 \boldsymbol{Q}_{32} + \boldsymbol{A}_2 \boldsymbol{Q}_{22} + \boldsymbol{A}_3 \boldsymbol{Q}_{12} \tag{42}$$

$$\boldsymbol{Q}_{43} = \begin{bmatrix} q_{75} & q_{76} \\ q_{85} & q_{86} \end{bmatrix} = \boldsymbol{A}_1 \boldsymbol{Q}_{33} + \boldsymbol{A}_2 \boldsymbol{Q}_{23} + \boldsymbol{A}_3 \boldsymbol{Q}_{13} \tag{43}$$

The recursion of the matrix $\boldsymbol{G}$ is analogical to the SISO case. The technique is apparent from the equations (44) and (45).

$$\boldsymbol{G} \Delta \boldsymbol{u} = \left[\begin{array}{cc|cc} g(1,1) & g(1,2) & 0 & 0 \\ g(2,1) & g(2,2) & 0 & 0 \\ \hline g(3,1) & g(3,2) & g(1,1) & g(1,2) \\ g(4,1) & g(4,2) & g(2,1) & g(2,2) \\ \hline g(5,1) & g(5,2) & g(3,1) & g(3,2) \\ g(6,1) & g(6,2) & g(4,1) & g(4,2) \end{array}\right] \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \hline \Delta u_1(k+1) \\ \Delta u_2(k+1) \end{bmatrix} = \\ = \begin{bmatrix} G(1,1) & 0 \\ G(2,1) & G(1,1) \\ G(3,1) & G(2,1) \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u}(k) \\ \Delta \boldsymbol{u}(k+1) \end{bmatrix} \tag{44}$$

$$\boldsymbol{G}_{41} = \begin{bmatrix} g_{71} & g_{72} \\ g_{81} & g_{82} \end{bmatrix} = \boldsymbol{A}_1 \boldsymbol{G}_{31} + \boldsymbol{A}_2 \boldsymbol{G}_{21} + \boldsymbol{A}_3 \boldsymbol{G}_{11} \tag{45}$$

The predictions can be written in a compact matrix form

$$\hat{\boldsymbol{y}}(k+j) = \boldsymbol{G} \varDelta \boldsymbol{u}(k+j-1) + \boldsymbol{P} \varDelta \boldsymbol{u}(k-1) + \boldsymbol{Q} \boldsymbol{y}(k-j+1) \\ j \le N \tag{46}$$

5 Computation of predictions with colouring filter *C*

5.1 Computation of predictions for SISO system

Computation of predictions for C≠1 is solved for example in [14]. Including the C-filter the CARIMA model takes the form

$$\varDelta A(z^{-1}) y(k) = B(z^{-1}) \varDelta u(k) + C(z^{-1}) n(k) \tag{47}$$

Equation (49) can be modified to

$$\varDelta A(z^{-1}) \frac{y(k)}{C(z^{-1})} = B(z^{-1}) \frac{\varDelta u(k)}{C(z^{-1})} + n(k) \tag{48}$$

Where the unknown term is supposed to be the white noise and the input and output variables are filtered. Using of (48) for prediction improves prediction accuracy.

The filtered variables are defined as

$$y_f(k) = \frac{y(k)}{C(z^{-1})} \tag{49}$$

$$u_f(k) = \frac{u(k)}{C(z^{-1})} \tag{50}$$

In this case the polynomial C is a design parameter. It is a stable polynomial. In case of the system (1) it was chosen to be of the second order as

$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} \tag{51}$$

The input and output data are filtered before prediction. $1/C$ is a low-pass filter which reduces high frequency noise. It is easy to prove by simulation that the cases when the noise is coloured (47) and when the noise is white and the input and output variables are filtered (48) are equal.

The prediction equation for filtered variables takes the following form

$$\hat{\boldsymbol{y}}_f(k+j) = \boldsymbol{G} \Delta \boldsymbol{u}_f(k+j-1) + \boldsymbol{P} \Delta \boldsymbol{u}_f(k-1) + \boldsymbol{Q} \boldsymbol{y}_f(k-j+1) \\ j \le N \tag{52}$$

For practical application the equation (52) is inapplicable. Prediction of the unfiltered output must be expressed by means of future control increments.

The relationship between filtered and unfiltered variables can be expressed as follows

$$y_f(k) = \frac{y(k)}{1 + c_1 z^{-1} + c_2 z^{-2}} \tag{53}$$

$$y(k) = y_f(k) + c_1 y_f(k-1) + c_2 y_f(k-2) \tag{54}$$

For three step ahead predictions

$$\begin{aligned} \hat{y}(k+1) &= y_f(k+1) + c_1 y_f(k) + c_2 y_f(k-1) \\ \hat{y}(k+2) &= y_f(k+2) + c_1 y_f(k+1) + c_2 y_f(k) \\ \hat{y}(k+3) &= y_f(k+3) + c_1 y_f(k+2) + c_2 y_f(k+1) \end{aligned} \tag{55}$$

In a matrix form the equations (31) can be expressed as follows

$$\begin{aligned} \begin{pmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 \\ c_1 & 1 & 0 \\ c_2 & c_1 & 1 \end{pmatrix} \begin{pmatrix} y_f(k+1) \\ y_f(k+2) \\ y_f(k+3) \end{pmatrix} + \\ &+ \begin{pmatrix} c_1 & c_2 & 0 \\ c_2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_f(k) \\ y_f(k-1) \\ y_f(k-2) \end{pmatrix} \end{aligned} \tag{56}$$

The relationship between filtered and unfiltered control increments can be expressed similarly. Using matrix notation we can define following equations

$$\hat{\boldsymbol{y}}(k+j) = \boldsymbol{C}_C \boldsymbol{y}_f(k+j) + \boldsymbol{H}_C \boldsymbol{y}_f(k-j+1) \tag{57}$$

$$\Delta \boldsymbol{u}(k+j) = \boldsymbol{C}_C \Delta \boldsymbol{u}_f(k+j) + \boldsymbol{H}_C \Delta \boldsymbol{u}_f(k-j+1) \tag{58}$$

Where matrices $\boldsymbol{C}_c$ and $\boldsymbol{H}_c$ are defined as follows

$$\boldsymbol{C}_C = \begin{pmatrix} 1 & 0 & 0 \\ c_1 & 1 & 0 \\ c_2 & c_1 & 1 \end{pmatrix} \tag{59}$$

$$\boldsymbol{H}_C = \begin{pmatrix} c_1 & c_2 & 0 \\ c_2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{60}$$

From equations (57) and (58) we can express the filtered variables

$$\boldsymbol{y}_f(k+j) = \boldsymbol{C}_C^{-1}\left(\hat{\boldsymbol{y}}(k+j) - \boldsymbol{H}_C \boldsymbol{y}_f(k-j+1)\right) \tag{61}$$

$$\Delta \boldsymbol{u}_f(k+j) = \boldsymbol{C}_C^{-1}\left(\Delta \boldsymbol{u}(k+j) - \boldsymbol{H}_C \Delta \boldsymbol{u}_f(k-j+1)\right) \tag{62}$$

After substitution of equations (61) and (62) to equation (52) we obtain

$$\begin{aligned} &\boldsymbol{C}_C^{-1}\left(\hat{\boldsymbol{y}}(k+j) - \boldsymbol{H}_C \boldsymbol{y}_f(k-j+1)\right) = \\ &\boldsymbol{G}\boldsymbol{C}_C^{-1}\left(\Delta \boldsymbol{u}(k+j-1) - \boldsymbol{H}_C \Delta \boldsymbol{u}_f(k-1)\right) + \\ &+ \boldsymbol{P}\Delta \boldsymbol{u}_f(k-1) + \boldsymbol{Q}\boldsymbol{y}_f(k-j+1) \end{aligned} \tag{63}$$

After modification we obtain resulting equation of the predictor

$$\begin{aligned} \hat{\boldsymbol{y}}(k+j) &= \boldsymbol{G}\Delta \boldsymbol{u}(k+j-1) + [\boldsymbol{C}_C \boldsymbol{P} - \boldsymbol{G}\boldsymbol{H}_C]\Delta \boldsymbol{u}_f(k-1) + \\ &+ [\boldsymbol{H}_C + \boldsymbol{C}_C \boldsymbol{Q}]\boldsymbol{y}_f(k-j+1) \end{aligned} \tag{64}$$

We can establish following substitutions

$$\tilde{\boldsymbol{P}} = [\boldsymbol{C}_C \boldsymbol{P} - \boldsymbol{G}\boldsymbol{H}_C] \tag{65}$$

$$\tilde{\boldsymbol{Q}} = [\boldsymbol{H}_C + \boldsymbol{C}_C \boldsymbol{Q}] \tag{66}$$

The prediction equation then can be written in the form

$$\hat{\boldsymbol{y}}(k+j) = \boldsymbol{G}\Delta \boldsymbol{u}(k+j-1) + \tilde{\boldsymbol{P}}\Delta \boldsymbol{u}_f(k-1) + \tilde{\boldsymbol{Q}}\boldsymbol{y}_f(k-j+1) \tag{67}$$

5.2 Computation of predictions for MIMO System

Including the *C*-filter the multivariable CARIMA model takes the form

$$\Delta \boldsymbol{A}(z^{-1})\boldsymbol{y}(k) = \boldsymbol{B}(z^{-1})\Delta \boldsymbol{u}(k) + C(z^{-1})n(k) \tag{68}$$

As it was previously mentioned, a simplified model when the non-measurable random disturbance was a scalar was considered.

Equation (70) can be modified to

$$\Delta \boldsymbol{A}(z^{-1})\boldsymbol{y}(k)\frac{1}{C(z^{-1})} = \boldsymbol{B}(z^{-1})\Delta \boldsymbol{u}(k)\frac{1}{C(z^{-1})} + n(k) \tag{69}$$

where the unknown term is supposed to be the white noise and the input and output variables are filtered. Using of (69) for prediction improves prediction accuracy.

The filtered variables are defined as

$$\begin{pmatrix} y_{1f}(k) \\ y_{2f}(k) \end{pmatrix} = \begin{pmatrix} y_1(k) \\ y_2(k) \end{pmatrix}\frac{1}{C(z^{-1})} \tag{70}$$

$$\begin{pmatrix} u_{1f}(k) \\ u_{2f}(k) \end{pmatrix} = \begin{pmatrix} u_1(k) \\ u_2(k) \end{pmatrix}\frac{1}{C(z^{-1})} \tag{71}$$

The polynomial C is a design parameter. It is a stable polynomial. For the system with polynomials of the second degree (8), (9) it was chosen to be of the second degree as well

$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} \tag{72}$$

The prediction equation for filtered variables takes the following form

$$\begin{aligned} &\hat{\boldsymbol{y}}_f(k+j) = \boldsymbol{G}\Delta \boldsymbol{u}_f(k+j-1) + \boldsymbol{P}\Delta \boldsymbol{u}_f(k-1) + \boldsymbol{Q}\boldsymbol{y}_f(k-j+1) \\ &j \le N \end{aligned} \tag{73}$$

The relationship between filtered and unfiltered variables can be expressed as follows

$$\begin{pmatrix} y_{1f}(k) \\ y_{2f}(k) \end{pmatrix} = \begin{pmatrix} y_1(k) \\ y_2(k) \end{pmatrix} \frac{1}{1 + c_1 z^{-1} + c_2 z^{-2}} \tag{74}$$

$$\begin{pmatrix} y_1(k) \\ y_2(k) \end{pmatrix} = \begin{pmatrix} y_{1f}(k) + c_1 y_{1f}(k-1) + c_2 y_{1f}(k-2) \\ y_{2f}(k) + c_1 y_{2f}(k-1) + c_2 y_{2f}(k-2) \end{pmatrix} \tag{75}$$

For three step ahead predictions

$$\begin{aligned} \hat{\boldsymbol{y}}(k+1) &= \boldsymbol{y}_f(k+1) + c_1 \boldsymbol{y}_f(k) + c_2 \boldsymbol{y}_f(k-1) \\ \hat{\boldsymbol{y}}(k+2) &= \boldsymbol{y}_f(k+2) + c_1 \boldsymbol{y}_f(k+1) + c_2 \boldsymbol{y}_f(k) \\ \hat{\boldsymbol{y}}(k+3) &= \boldsymbol{y}_f(k+3) + c_1 \boldsymbol{y}_f(k+2) + c_2 \boldsymbol{y}_f(k+1) \end{aligned} \tag{76}$$

In a matrix form the equations (76) can be expressed as follows

$$\begin{pmatrix} \hat{y}_1(k+1) \\ \hat{y}_2(k+1) \\ \hat{y}_1(k+2) \\ \hat{y}_2(k+2) \\ \hat{y}_1(k+3) \\ \hat{y}_2(k+3) \end{pmatrix} = \overbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ c_1 & 0 & 1 & 0 & 0 & 0 \\ 0 & c_1 & 0 & 1 & 0 & 0 \\ c_2 & 0 & c_1 & 0 & 1 & 0 \\ 0 & c_2 & 0 & c_1 & 0 & 1 \end{pmatrix}}^{\boldsymbol{C}_c} \begin{pmatrix} \hat{y}_{1f}(k+1) \\ \hat{y}_{2f}(k+1) \\ \hat{y}_{1f}(k+2) \\ \hat{y}_{2f}(k+2) \\ \hat{y}_{1f}(k+3) \\ \hat{y}_{2f}(k+3) \end{pmatrix} +$$

$$+ \overbrace{\begin{pmatrix} c_1 & 0 & c_2 & 0 & 0 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & 0 \\ c_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}^{\boldsymbol{H}_c} \begin{pmatrix} \hat{y}_{1f}(k) \\ \hat{y}_{2f}(k) \\ \hat{y}_{1f}(k-1) \\ \hat{y}_{2f}(k-1) \\ \hat{y}_{1f}(k-2) \\ \hat{y}_{2f}(k-2) \end{pmatrix} \tag{77}$$

The relationship between filtered and unfiltered control increments can be expressed similarly. Using matrix notation we can define following equations

$$\hat{\boldsymbol{y}}(k+j) = \boldsymbol{C}_C \boldsymbol{y}_f(k+j) + \boldsymbol{H}_C \boldsymbol{y}_f(k-j+1) \tag{78}$$

$$\Delta\boldsymbol{u}(k+j) = \boldsymbol{C}_C \Delta\boldsymbol{u}_f(k+j) + \boldsymbol{H}_C \Delta\boldsymbol{u}_f(k-j+1) \tag{79}$$

where matrices $\boldsymbol{C}_c$ and $\boldsymbol{H}_c$ are defined as follows

$$\boldsymbol{C}_C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ c_1 & 0 & 1 & 0 & 0 & 0 \\ 0 & c_1 & 0 & 1 & 0 & 0 \\ c_2 & 0 & c_1 & 0 & 1 & 0 \\ 0 & c_2 & 0 & c_1 & 0 & 1 \end{pmatrix} \tag{80}$$

$$\boldsymbol{H}_C = \begin{pmatrix} c_1 & 0 & c_2 & 0 & 0 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & 0 \\ c_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{81}$$

From equations (78) and (79) we can express the filtered variables

$$\hat{\boldsymbol{y}}(k+j) = \boldsymbol{C}_C \boldsymbol{y}_f(k+j) + \boldsymbol{H}_C \boldsymbol{y}_f(k-j+1) \tag{82}$$

$$\Delta\boldsymbol{u}(k+j) = \boldsymbol{C}_C \Delta\boldsymbol{u}_f(k+j) + \boldsymbol{H}_C \Delta\boldsymbol{u}_f(k-j+1) \tag{83}$$

After substitution of equations (82) and (83) to equation (73) we obtain

$$\begin{aligned} &\boldsymbol{C}_C^{-1}\left(\hat{\boldsymbol{y}}(k+j) - \boldsymbol{H}_C \boldsymbol{y}_f(k-j+1)\right) = \\ &= \boldsymbol{G}\boldsymbol{C}_C^{-1}\left(\Delta\boldsymbol{u}(k+j-1) - \boldsymbol{H}_C \Delta\boldsymbol{u}_f(k-1)\right) + \\ &+ \boldsymbol{P}\Delta\boldsymbol{u}_f(k-1) + \boldsymbol{Q}\boldsymbol{y}_f(k-j+1) \end{aligned} \tag{84}$$

After modification we obtain resulting equation of the predictor

$$\begin{aligned} \hat{\boldsymbol{y}}(k+j) &= \boldsymbol{G}\Delta\boldsymbol{u}(k+j-1) + [\boldsymbol{C}_C \boldsymbol{P} - \boldsymbol{G}\boldsymbol{H}_C]\Delta\boldsymbol{u}_f(k-1) + \\ &+ [\boldsymbol{H}_C + \boldsymbol{C}_C \boldsymbol{Q}]\boldsymbol{y}_f(k-j+1) \end{aligned} \tag{85}$$

We can establish following substitutions

$$\tilde{\boldsymbol{P}} = [\boldsymbol{C}_C \boldsymbol{P} - \boldsymbol{G}\boldsymbol{H}_C] \tag{86}$$

$$\tilde{\boldsymbol{Q}} = [\boldsymbol{H}_C + \boldsymbol{C}_C \boldsymbol{Q}] \tag{87}$$

The prediction equation then can be written in the form

$$\hat{\boldsymbol{y}}(k+j) = \boldsymbol{G}\Delta\boldsymbol{u}(k+j-1) + \tilde{\boldsymbol{P}}\Delta\boldsymbol{u}_f(k-1) + \tilde{\boldsymbol{Q}}\boldsymbol{y}_f(k-j+1) \tag{88}$$

6 Simulation Verification

6.1 SISO control

Verification by simulation was carried out on a range of plants with various dynamics. The control of the model below is given here as an example.

$$G(s) = \frac{3}{5s^2 + 6s + 1} \tag{89}$$

It does not exist a systematic way for selection of the filter *C*. Its selection is mostly based on intuition. In our example the filter was chosen as

$$C(z^{-1}) = 1 + 0{,}8z^{-1} + 0{,}05z^{-2} \tag{90}$$

The sampling period was tuned experimentally and the best value was $T_0 = 2$ *s*. The controlled variable was affected by a noise with zero mean value and constant covariance. Simulation sampling of noise was 0,1 *s*.

In figures 1-6 are simulation results. Figures 1,3 and 5 show time responses of the control without the filtering of the variables introduced in section 4. Figures 2, 4 and 6 show time responses of the control with the filtering of variables described in section 5.

In figures 1 and 2 there is the response of the controlled variable taken by 0,1 *s*. It means with the same sampling period as the simulation noise. Simulation results in this figure are the closest to the reality. In figures 3 and 4 there is the controlled variable taken by 2 *s*. It means with the same sampling period which is used for the control. The data then simulates measured values. In figures 5 and 6 is the manipulated variable.

The tuning parameters that are lengths of the prediction and control horizons and the weighting coefficient λ were tuned experimentally. There is a lack of clear theory relating to the closed loop behavior to design parameters. The length of the prediction horizon, which should cover the important part of the step response, was set to $N = 5$. The length of the control horizon was also set to $N_u = 5$. The coefficient λ was taken as equal to 0,1.

It is necessary to emphasize that the displayed inputs and outputs in the graphs are not filtered. The filtered values are used only for computation of systems output predictions and consequently for computation of the control law. The displayed inputs and outputs are real unfiltered values.

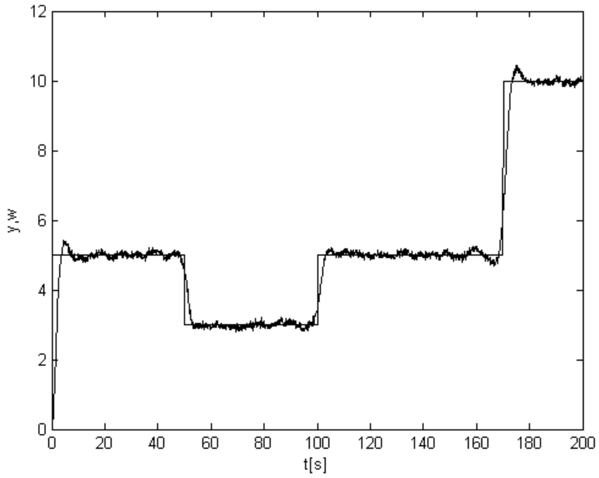

Fig. 1 Controlled variable sampled by 0,1*s* – case without filtering of variables

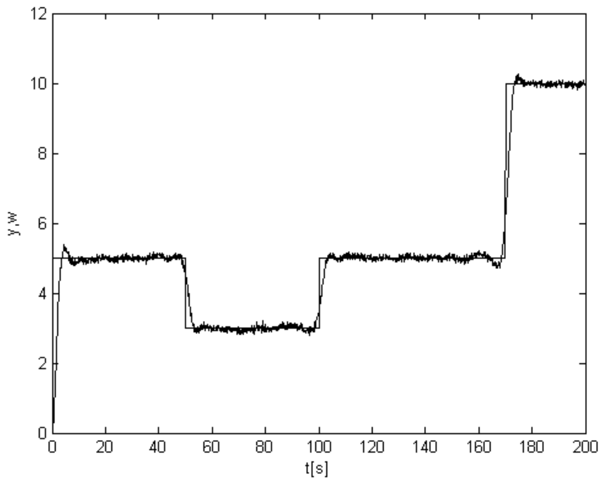

Fig. 2 Controlled variable sampled by 0,1*s* – case with filtering of variables

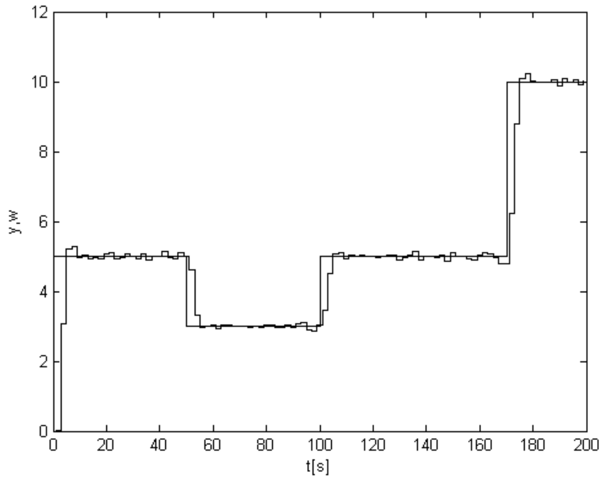

Fig 3 Controlled variable sampled by 2*s* – case without filtering of variables

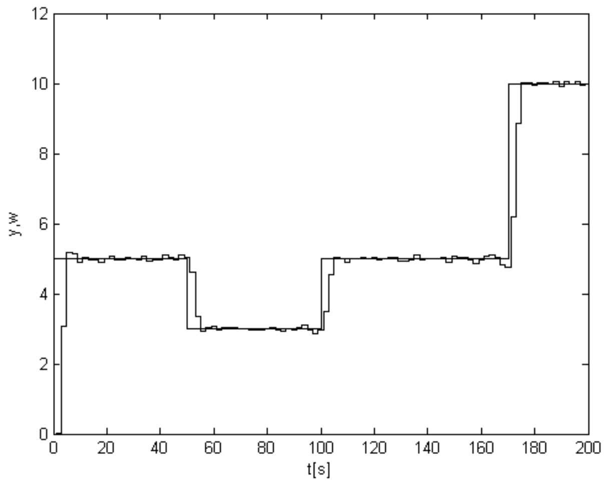

Fig 4 Controlled variable sampled by 2*s* – case with filtering of variables

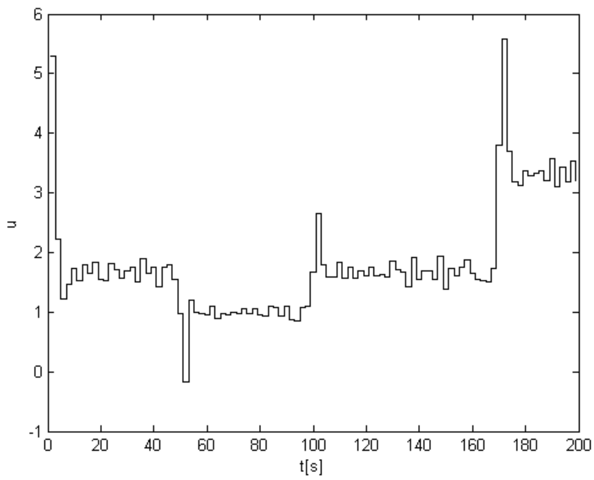

Fig 5 Manipulated variable – case without filtering of variables

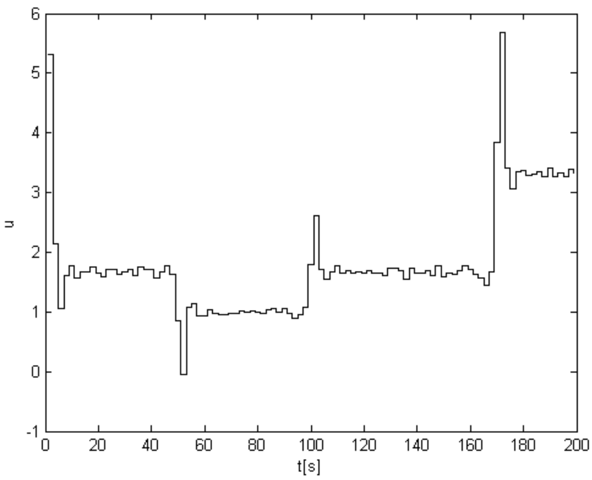

Fig 6 Manipulated variable – case with filtering of variables

6.2 MIMO Control

The MIMO simulation controlled system was chosen as follows

$$\boldsymbol{G}(s)=\begin{bmatrix}\dfrac{3}{5s^2+6s+1} & \dfrac{2}{2s^2+4s+1}\\ \dfrac{7}{3s^2+10s+1} & \dfrac{5}{2s^2+8s+1}\end{bmatrix} \tag{91}$$

A corresponding discrete model in the form given by equations (8), (9) and (24) was obtained by recursive identification. Control in the initial adaptation phase then has worse quality. The filter C was chosen as (90).

The best value of the sampling period was found as T_0 = 0,5 *s*. The controlled variable was affected by a noise with zero mean value and constant covariance. Simulation sampling of noise was 0,1 *s*.

In figures 1-6 are simulation results. Figures 1,3 and 5 show time responses of the control without the filtering of the variables introduced in section 4. Figures 2, 4 and 6 show time responses of the control with the filtering of variables described in section 5.

In figures 1 and 2 there is the response of the controlled variable taken by 0,1 *s*. It means with the same sampling period as the simulation noise. In figures 3 and 4 there is the controlled variable taken by 2 *s*. It means with the same sampling period which is used for the control. In figures 5 and 6 is the manipulated variable.

The length of the prediction horizon was set to N = 5. The length of the control horizon was also set to N_u = 5. The coefficient λ was taken as equal to 0,1.

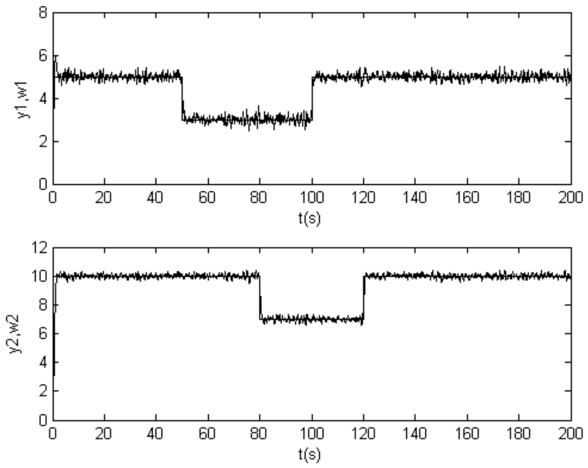

Fig. 7 Controlled variable sampled by 0,1*s* – case without filtering of variables

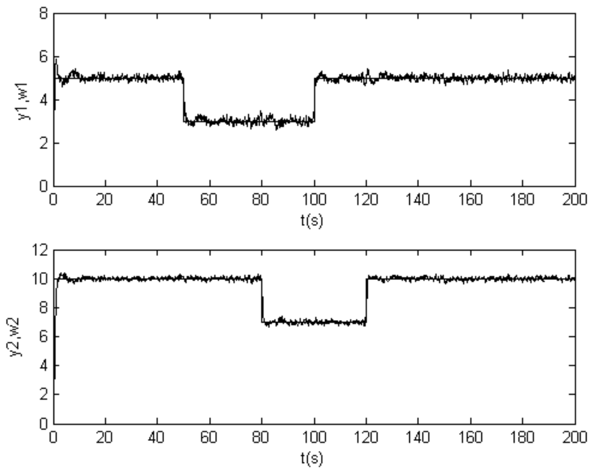

Fig. 8 Controlled variables sampled by 0,1*s* – case with filtering of variables

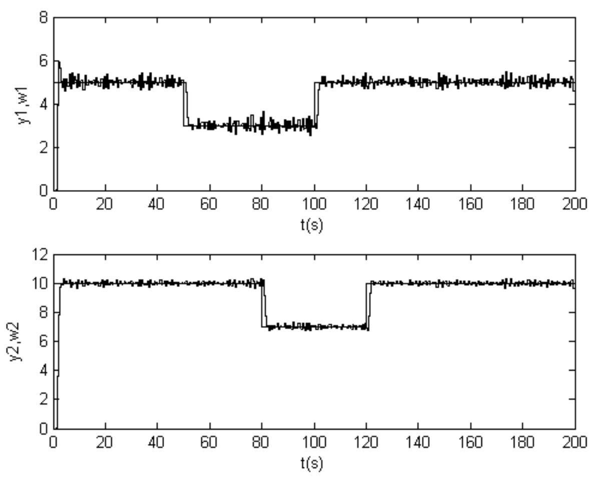

Fig. 9 Controlled variables sampled by 0,5*s* – case without filtering of variables

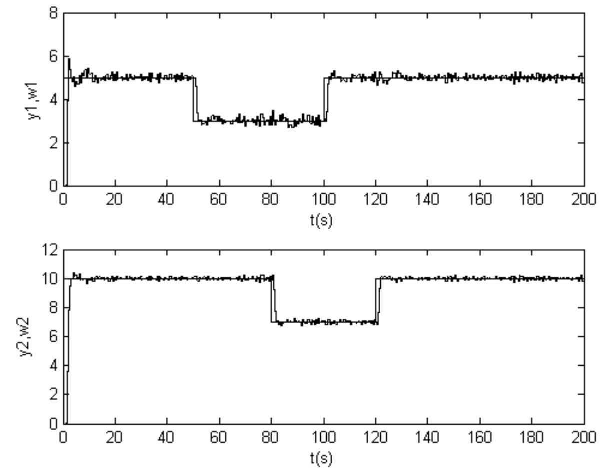

Fig. 10 Controlled variables sampled by 0,5*s* – case with filtering of variables

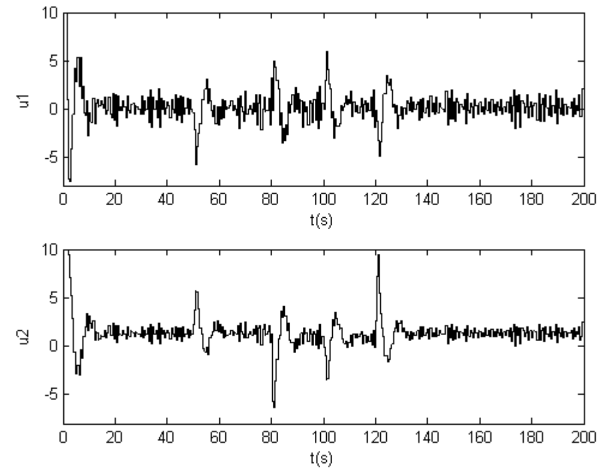

Fig. 11 Manipulated variables – case without filtering of variables

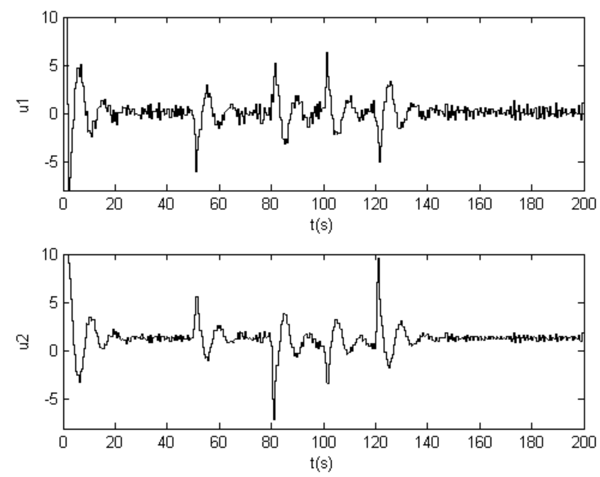

Fig. 12 Manipulated variables – case with filtering of variables

7 Conclusions

Specific self-contained prediction equations for the input-output models in the form of transfer function in case of SISO system and matrix fraction in case of MIMO system were derived for the case with filtering of the input and output variables. Simulations, where the filtered variables are used for computation of the control law and the manipulated variable, were performed. In the simulation results are displayed real unfiltered variables. By simulation control of a range of systems were compared control results of cases with and without the C-filter. The C-filter is a tuning parameter for which setting we do not have available any exact methodology. The filter was designed by try it and see approach as a low pass filter. Obviously better results were achieved in case with the C-filter particularly regarding rate of oscillations of the input and output variables. It is obvious that the variables are more settled in case with the C-filter. The filter reduces sensitivity of the closed loop system to high frequency noise. Cost for this improvement is a relatively difficult setting of the C-filter as a parameter.

References:

[1] E. F. Camacho, C. Bordons, *Model Predictive Control*, Springer-Verlag, London, 2004.

[2] M. Morari, J. H. Lee, Model predictive control: past, present and future. *Computers and Chemical Engineering*, 23, 1999, 667-682.

[3] J. Mikleš & M. Fikar, *Process Modelling, Optimisation and Control.* (Berlin: Springer-Verlag, 2008).

[4] D. W. Clarke, C. Mohtadi, P. S. Tuffs, Generalized predictive control, part I: the basic algorithm. *Automatica*, 23, 1987, 137-148.

[5] D. W. Clarke, C. Mohtadi, P. S. Tuffs, Generalized predictive control, part II: extensions and interpretations. *Automatica*, 23, 1987, 149-160.

[6] P.R. Krishnawamy, et al., Reference System Decoupling for Multivariable Control. *Ind. Eng. Chem. Res*., 30, 1991, 662-670.

[7] W. L. Luyben, Simple Method for Tuning SISO Controllers in Multivariable Systems, *Ing. Eng. Chem. Process Des. Dev.*, Vol. 25, 1986, 654 – 660.

[8] S. J. Quin & T. A. Bandgwell, An overview of industrial model predictive control technology. *Proceedings of the Chemical Process Control – V.* Vol. 93 of *AIChE Symposium Series.* CACHE & AIChE. Tahoe City, CA, USA, 1996, 232-256.

[9] S. J. Quin & T. A. Bandgwell, An overview of nonlinear model predictive control applications. *Nonlinear Model Predictive Control* (F. Allgöwer & A. Zheng, Ed.), (Basel – Boston – Berlin: Birkhäuser Verlag, 2000), 369-392.

[10] S. J. Quin & T. A. Bandgwell, A survey of industrial model predictive control technology. *Control Engineering Practice*, *11*(7), 2003, 733-764.

[11] F. G. Shinskey, *Process Control System*, 4th ed.; McGraw-Hill. New York, 1996.

[12] M. Kubalčík, M., V. Bobál, Adaptive Control of Coupled Drives Apparatus Based on Polynomial Theory. In Proc. IMechE Vol. 220 Part I: *J. Systems and Control Engineering*, 220(I7), 2006, 641-654.

[13] T. W. Yoon, D.W. Clarke, Observer design in receding horizon predictive control, *International Journal of Control*, 61, 1995, 171-191.

[14] J. A. Rossiter, *Model Based Predictive Control: a Practical Approach* (CRC Press, 2003).