T-POLYNOMIAL IN GENERALIZED PREDICTIVE CONTROL

D. Honc, F. Dušek

Institute of Electrical Engineering and Informatics, University of Pardubice, Czech Republic

Abstract

Briefly theory is given to introduce Model Predictive Control (MPC) concept. Method called Generalized Predictive Control (GPC) published in [1], [2] and [3] is based on Controller Auto-Regressive Integrated Moving-Average (CARIMA) model. Resulting controller has integrating character. In white noise case (without data filtering) the controller is rather sensitive to the measurement noise and model uncertainties. Polynomial C in process model is usually used as a controller parameter (as a filter) to increase controller robustness and it is called as T-polynomial. Different approaches how to design this polynomial coming out from the process model and controller parameters are suggested in the literature [4]. In the paper Kalman estimator is designed and its characteristic polynomial is used as a polynomial T. Prediction abilities of the models with and without data filtering are compared.

1 Introduction

Model Based Predictive Control involves wide range of control approaches sharing following features: using of process model to predict the plant future behaviour, control action calculation by minimization of the cost function – usually taking into account future control movements and future control error, receding horizon concept – whole procedure is repeated every sample time. Control is designed in time area, the process variables and criterion may have physical meanings (open way to production optimization), it is possible to cope with dead-times, non-minimal plants and multivariable systems in a nature way, different types of constraints can be handled with, etc. All this features caused that MPC is popular between control engineers and its popularity grows increasingly in research community too.

2 Generalized Predictive Control

Only main steps how to design GPC controller are outlined in the following text. Detailed solution is shown e.g. in [4].

CARIMA model is usually considered if disturbances are non-stationary – controller has integrating character. Process model is expressed as

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

where $A(z^{-1})$, $B(z^{-1})$ and $C(z^{-1})$ are polynomials defined as

$$A\left(z^{-1}\right) = 1 + a_1 z^{-1} + \ldots + a_{na} z^{-na}$$
$$B\left(z^{-1}\right) = b_0 + b_1 z^{-1} + \ldots + b_{nb} z^{-nb}$$
$$C\left(z^{-1}\right) = 1 + c_1 z^{-1} + \ldots + c_{nc} z^{-nc}$$

The operator Δ is defined as $\Delta = 1 - z^{-1}$, *y(k)* is plant output, *u(k)* is plant input and *e(k)* is zero-mean white noise.

Finite horizon quadratic criterion is considered (cost function) with penalization of future control error and control moves. J-step ahead predictions are expressed as a sum of forced response (response obtained if initial conditions are zero) and free response (response of the process if the control signals are kept constant). If there are no constraints the optimum can be expressed analytically as a linear gain matrix which multiplies predicted errors between the future reference and the free response of the process. If there are no predicted errors there is no control move - the objective will be fulfilled with the free evolution of the process.

Instead of polynomial C so called T- polynomial is used as a fixed observer or a prefilter. Low frequency disturbances are removed by the Δ term in the prediction. Polynomial T detunes the response to unmeasurable high-frequency disturbances and prevents against excessive control actions. T is used as a design parameter that can influence robustness of the controller. Different approaches how to design T can be found in the literature but systematic approach has not been established. In our paper Kalman state estimator is designed and its characteristic polynomial is used as a T-polynomial.

3 Kalman state estimator design

The Kalman estimator is the optimal solution to the following estimation problem.

For a given discrete-time state-space plant model

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) + \mathbf{G}w(k) \\ y_m(k) &= \mathbf{C}x(k) + \mathbf{D}u(k) + \mathbf{H}w(k) + v(k) \end{aligned} \tag{2}$$

and the process and measurement noise covariance data $E\{ww^T\} = Q, \quad E\{vv^T\} = R$ to construct a state estimate that minimizes the steady-state error covariance

$$P = \lim_{k\to\infty} E\{e(k \mid k-1)e(k \mid k-1)^T\}, \quad e(k \mid k-1) = x(k) - \hat{x}(k \mid k-1) \tag{3}$$

The Kalman estimator has equation

$$\hat{\mathbf{x}}(k+1 \mid k) = \mathbf{A}\hat{\mathbf{x}}(k \mid k-1) + \mathbf{B}u(k) + \mathbf{L}(y_m(k) - \mathbf{C}\hat{\mathbf{x}}(k \mid k-1) - \mathbf{D}u(k)) \tag{4}$$

The gain matrix $\mathbf{L}$ is computed by solving a discrete Riccati equation (the size of matrix $\mathbf{L}$ is number of the states $\times$ number of the plant outputs).

4 Input-output equivalent to Kalman estimator

We can use Kalman estimator as a filter to calculate filtered (estimated) plant outputs from measured outputs and inputs.

Prediction of the plant output with Kalman estimator is

$$\hat{y}(k+1 \mid k) = \mathbf{C}[\mathbf{A}\hat{\mathbf{x}}(k \mid k-1) + \mathbf{B}u(k) + \mathbf{L}(y_m(k) - \mathbf{C}\hat{\mathbf{x}}(k \mid k-1) - \mathbf{D}u(k))] + \mathbf{D}u(k) \tag{5}$$

After Z-transformation we get discrete transfer function between plant input and measured output and filtered (estimated) plant output as

$$\hat{Y}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A} + \mathbf{L}\mathbf{C})^{-1}((\mathbf{B} - \mathbf{L}\mathbf{D})U(z) + \mathbf{L}Y_m(z)) + \mathbf{D}U(z) \tag{6}$$

It is not necessary to enumerate Equation (6). The deterministic part of the filter is identical with the process model. Equivalent CARMA process model can be used to get prediction equation

$$A(z^{-1})y_m(k) = B(z^{-1})u(k-1) + T(z^{-1})e(k), \quad e(k) = y_m(k) - \hat{y}(k \mid k-1) \tag{7}$$

Where $e(k)$ is 1-step ahead prediction error, polynomials A and B are process model polynomials and polynomial T is equal to the characteristic polynomial of Kalman estimator

$$T = \det(z\mathbf{I} - \mathbf{A} + \mathbf{L}\mathbf{C}) \tag{8}$$

The order of polynomial T is equal to the number of the state variables.

Optimal plant output prediction (we do not know future prediction error and therefore suppose $e(k+1)=0$) is

$$\begin{aligned} \hat{y}(k+1|k) &= z[1 - A(z^{-1})]y_m(k) + B(z^{-1})u(k) + z[T(z^{-1}) - 1]e(k) = \\ &= z[1 - T(z^{-1})]\hat{y}(k|k-1) + B(z^{-1})u(k) + z[T(z^{-1}) - A(z^{-1})]y_m(k) \end{aligned} \tag{9}$$

5 Simulated experiments

Following two simulated experiments are calculated to demonstrate the filter behaviour – the 1-step ahead prediction ability. Gausian white noises *w* and *v* are added to the input and output of the process. Variables *u* and y_m are used in the filter to predict $\hat{y}(k+1)$ denoted as y_p.

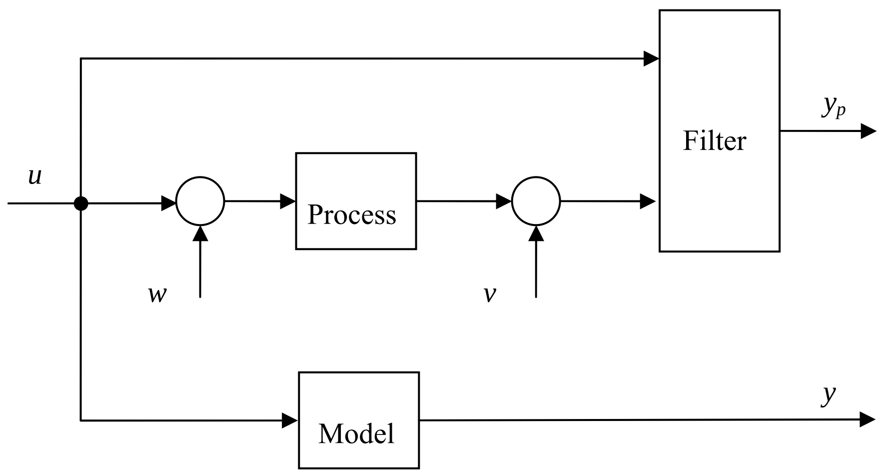
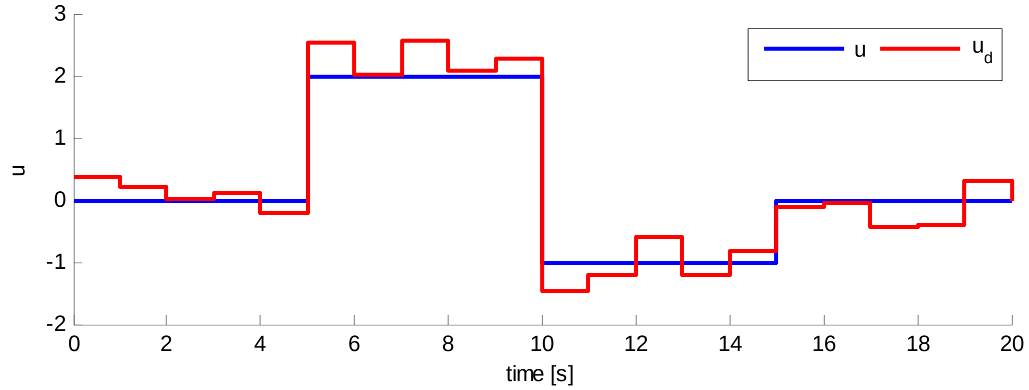


Figure 1: Variables notation scheme

Suppose *w* to be an additive Gausian white noise on the plant input (**G** = **B**) and *v* is Gausian white measurement noise.

Covariance matrices – variances of noise *w* and *v* are: Q = 0.1, R = 1

Our goal is to design a Kalman filter that estimates the output $\hat{y}(k+1)$ from given inputs *u(k)* and the noisy output measurements $y_m(k)$.



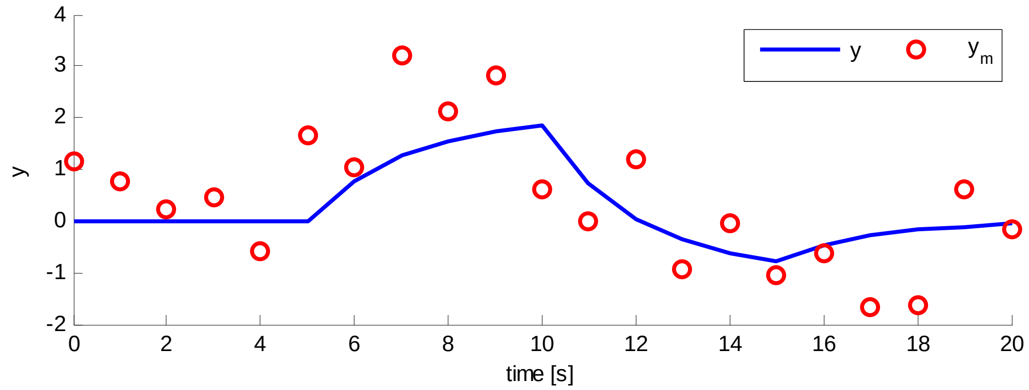


Figure 2: Model and noisy process input and output variables

Prediction with filtered variables is compared with "deterministic prediction". For white noise case - if $T(z^{-1}) = 1$ we get from Eq. (9)

$$\hat{y}(k+1) = B(z^{-1})u(k) + z[1 - A(z^{-1})]y_m(k)$$

5.1 First order example

Continuous-time process model: $G(s) = \dfrac{1}{2s+1}$

Discrete-time process model, sample time T_S = 1 s

Transfer function: $G(z^{-1}) = \dfrac{0.3935z^{-1}}{1-0.6065z^{-1}} \quad \rightarrow \quad A(z^{-1}) = 1 - 0.6065z^{-1}, \quad B(z^{-1}) = 0.3935$

State-space model:

$$\begin{aligned} x(k+1) &= 0.6065x(k) + 0.5u(k) \\ y_m(k) &= 0.7869x(k) \end{aligned} \quad \rightarrow \quad \mathbf{A} = 0.6065, \quad \mathbf{B} = 0.5, \quad \mathbf{C} = 0.7869$$

Calculated estimator gain: $L = 0.0182$

Filter from Eq. (6):

$$\hat{Y}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A} + \mathbf{LC})^{-1}(\mathbf{B}U(z) + \mathbf{L}Y_m(z)) = \frac{0.7869(0.5U(z) + 0.0182Y_m(z))}{z - 0.6065 + 0.0182 \cdot 0.7869}$$

$$\hat{y}(k+1) = 0.5922\hat{y}(k) + 0.3935u(k) + 0.0143y_m(k)$$

Characteristic polynomial of Kalman estimator:

$$T(z^{-1}) = z^{-1}\det(z\mathbf{I} - \mathbf{A} + \mathbf{LC}) = z^{-1}(z - 0.6065 + 0.0182 \cdot 0.7869) = 1 - 0.5922z^{-1}$$

Filter from Eq. (9):

$$\hat{y}(k+1) = z\left[1 - A(z^{-1})\right]y_m(k) + B(z^{-1})u(k) + z\left[T(z^{-1}) - 1\right](y_m(k) - \hat{y}(k)) =$$
$$= 0.6065y_m(k) + 0.3935u(k) - 0.5922(y_m(k) - \hat{y}(k)) = 0.5922\hat{y}(k) + 0.3935u(k) + 0.0143y_m(k)$$

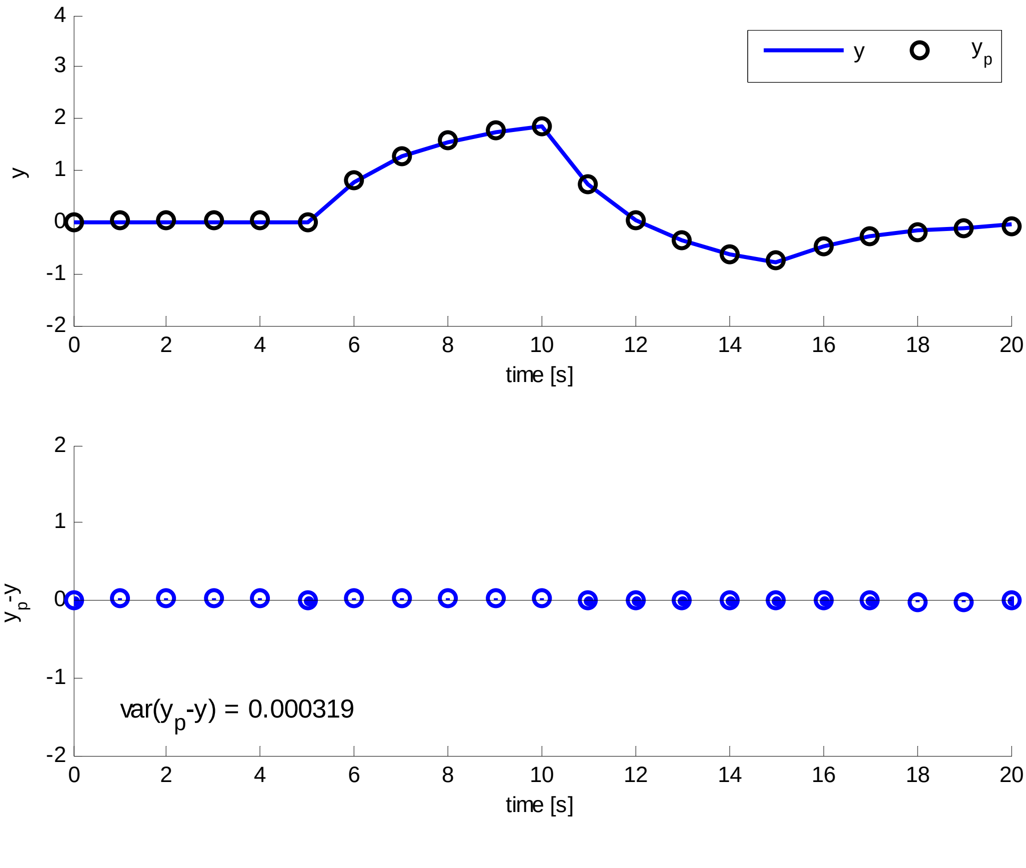


Figure 3: Plant output predictions – with filtered variables

Predictions without data filtering (from Eq. (9) if $T(z^{-1}) = 1$):

$$\hat{y}(k+1) = B(z^{-1})u(k) + z[1 - A(z^{-1})]y_m(k) = 0.3935u(k) + 0.6065y_m(k)$$

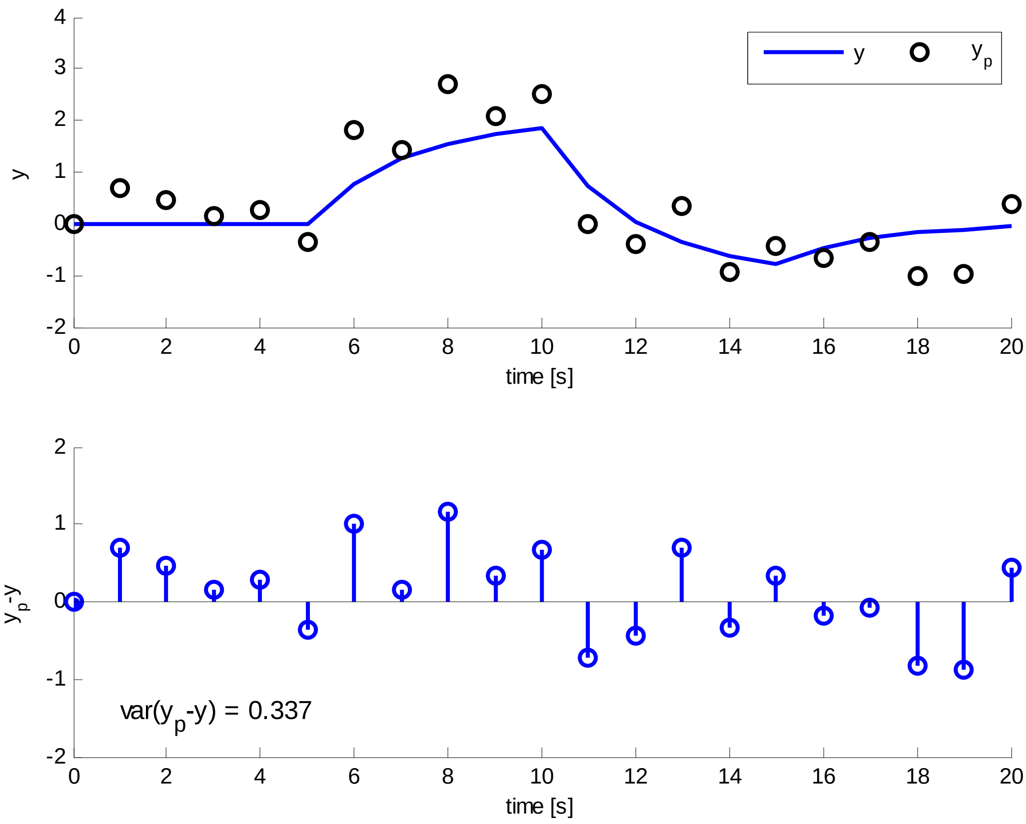


Figure 4: Plant output predictions – without filtering

5.2 Second order example

Continuous-time process model: $G(s) = \dfrac{1}{(s+1)^2}$

Discrete-time process model, sample time $T_S = 1$ s:

Transfer function:

$$G(z^{-1}) = \frac{0.2642z^{-1} + 0.1353z^{-2}}{1 - 0.7358z^{-1} + 0.1353z^{-2}} \quad \rightarrow \quad \begin{aligned} A &= 1 - 0.7358z^{-1} + 0.1353z^{-2} \\ B &= 0.2642 + 0.1353z^{-1} \end{aligned}$$

State-space:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.7358 & -0.2707 \\ 0.5 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k)$$

$$y_m(k) = \begin{bmatrix} 0.2642 & 0.2707 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad \rightarrow \quad \mathbf{A} = \begin{bmatrix} 0.7358 & -0.2707 \\ 0.5 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0.2642 & 0.2707 \end{bmatrix}$$

The noises are identical as in example 5.1.

Calculated estimator gain: $\mathbf{L}=\begin{bmatrix}0.0369\\0.0298\end{bmatrix}$

Filter from Eq. (6):

$$\hat{Y}(z)=\mathbf{C}(z\mathbf{I}-\mathbf{A}+\mathbf{LC})^{-1}(\mathbf{B}U(z)+\mathbf{L}Y_m(z))=$$

$$=[0.2642\quad 0.2707]\left(\begin{bmatrix}z & 0\\0 & z\end{bmatrix}-\begin{bmatrix}0.7358 & -0.2707\\0.5 & 0\end{bmatrix}+\begin{bmatrix}0.0369\\0.0298\end{bmatrix}[0.2642\quad 0.2707]\right)^{-1}\left(\begin{bmatrix}1\\0\end{bmatrix}U(z)+\begin{bmatrix}0.0369\\0.0298\end{bmatrix}Y_m(z)\right)=$$

$$=[0.2642\quad 0.2707]\begin{bmatrix}z-0.7261 & 0.2807\\-0.4921 & z+0.0081\end{bmatrix}^{-1}\left(\begin{bmatrix}1\\0\end{bmatrix}U(z)+\begin{bmatrix}0.0369\\0.0298\end{bmatrix}Y_m(z)\right)=$$

$$=[0.2642\quad 0.2707]\frac{\begin{bmatrix}z+0.0081 & -0.2807\\0.4921 & z-0.7261\end{bmatrix}}{z^2-0.7180z+0.1323}\left(\begin{bmatrix}1\\0\end{bmatrix}U(z)+\begin{bmatrix}0.0369\\0.0298\end{bmatrix}Y_m(z)\right)=$$

$$=\frac{[0.2642z+0.1353\quad 0.2707z-0.2707]}{z^2-0.7180z+0.1323}\left(\begin{bmatrix}1\\0\end{bmatrix}U(z)+\begin{bmatrix}0.0369\\0.0298\end{bmatrix}Y_m(z)\right)=$$

$$=\frac{(0.2642z+0.1353)U(z)+[(0.2642z+0.1353)0.0369+(0.2707z-0.2707)0.0298]Y_m(z)}{z^2-0.7180z+0.1323}=$$

$$=\frac{0.2642zU(z)+0.1353U(z)+0.0178zY_m(z)-0.0031Y_m(z)}{z^2-0.7180z+0.1323}$$

$$\hat{y}(k+1)=0.7180\hat{y}(k)-0.1323\hat{y}(k-1)+0.2642u(k)+0.1353u(k-1)+0.0178y_m(k)-0.0031y_m(k-1)$$

Characteristic polynomial of Kalman estimator:

$$T(z^{-1})=z^{-2}\det(z\mathbf{I}-\mathbf{A}+\mathbf{LC})=$$

$$=z^{-2}\det\left(\begin{bmatrix}z & 0\\0 & z\end{bmatrix}-\begin{bmatrix}0.7358 & -0.2707\\0.5 & 0\end{bmatrix}+\begin{bmatrix}0.0369\\0.0298\end{bmatrix}[0.2642\quad 0.2707]\right)=$$

$$=z^{-2}\det\left(\begin{bmatrix}z-0.7261 & 0.2807\\-0.4921 & z+0.0081\end{bmatrix}\right)$$

$$=1-0.7180z^{-1}+0.1323z^{-2}$$

Filter from Eq. (9):

$$\hat{y}(k+1)=z\left[1-A(z^{-1})\right]y_m(k)+B(z^{-1})u(k)+z\left[T(z^{-1})-I\right](y_m(k)-\hat{y}(k))=$$

$$=0.7358y_m(k)-0.1353y_m(k-1)+0.2642u(k)+0.1353u(k-1)+(-0.7180+0.1323z^{-1})(y_m(k)-\hat{y}(k))=$$

$$=0.7180\hat{y}(k)-0.1323\hat{y}(k-1)+0.2642u(k)+0.1353u(k-1)+0.0178y_m(k)-0.0031y_m(k-1)$$

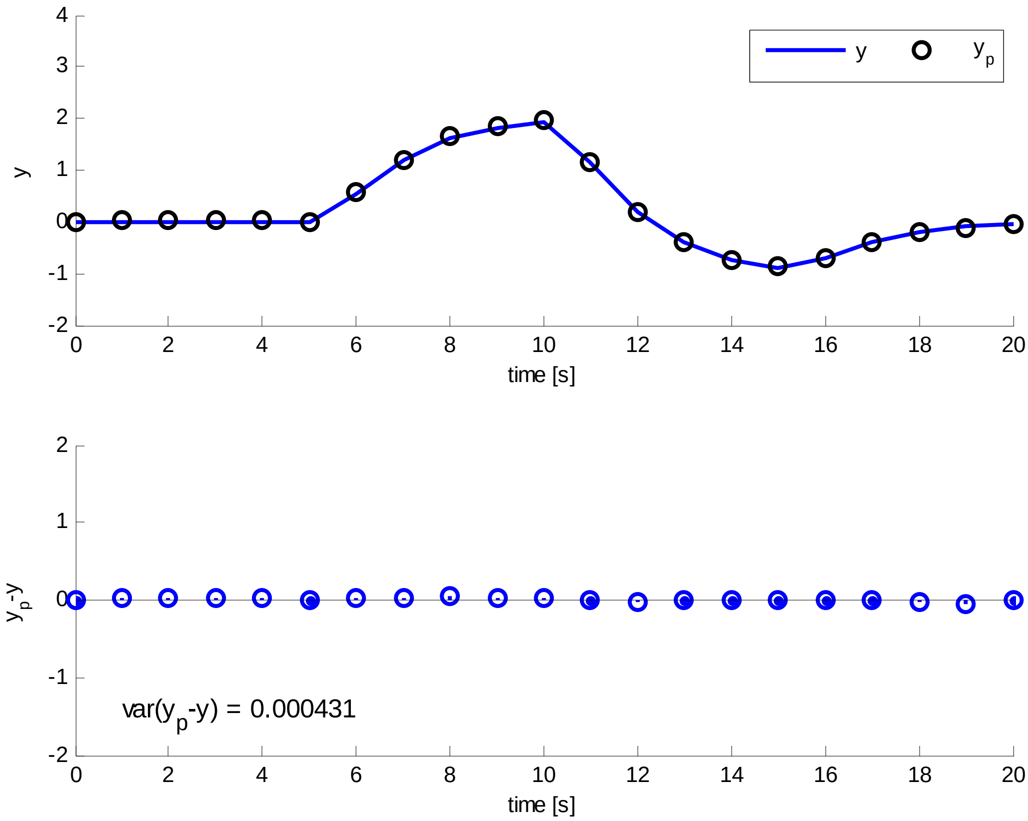


Figure 5: Plant output predictions – with filtered variables

Predictions without data filtering (from Eq. (9) if $T(z^{-1}) = 1$):

$$\hat{y}(k+1) = B(z^{-1})u(k) + z\left[1 - A(z^{-1})\right]y_m(k) = 0.2642u(k) + 0.1353u(k-1) + 0.7358y_m(k) - 0.1353y_m(k-1)$$

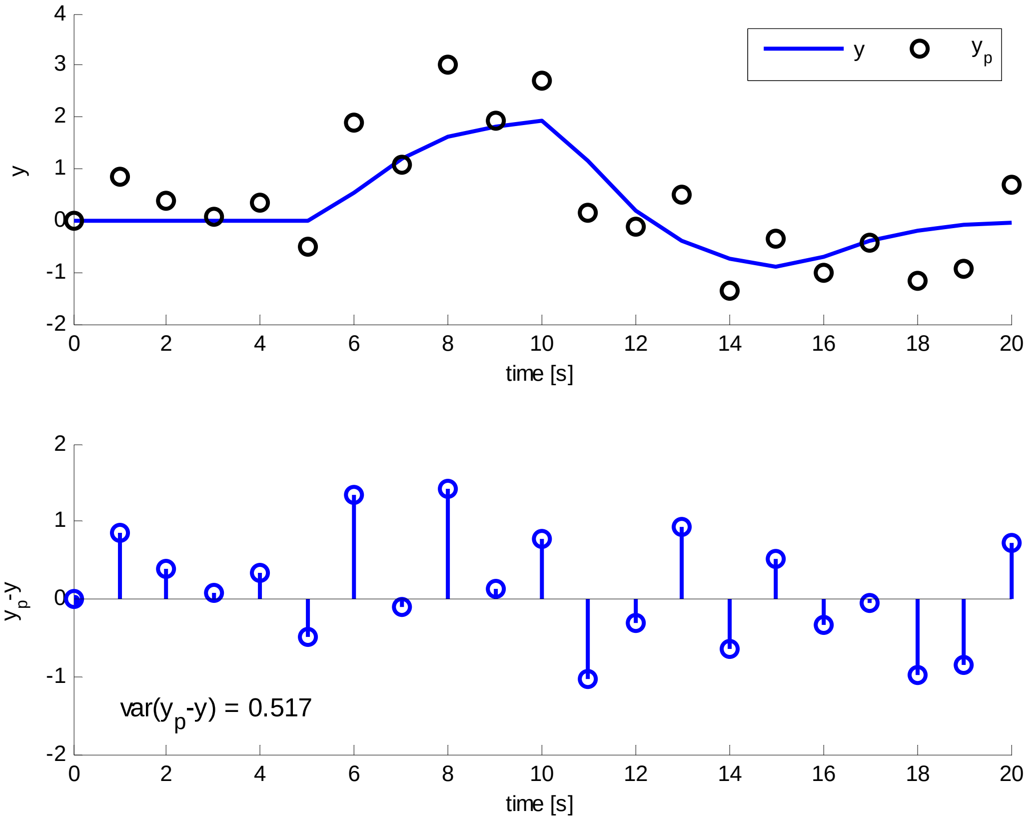


Figure 6: Plant output predictions – without filtering

6 Solution in MATLAB

Filter and characteristic polynomial equations are derived analytically in chapter 5. Code how to use MATLAB tools is given in the following script. Filter transfer function is obtained with Control Toolbox functions. Characteristic polynomial of Kalman estimator and filter symbolic transfer function are calculated with Symbolic Toolbox.

```
% Transfer function
mod_tf_c = tf(1,[2 1]);
mod_tf_c = tf(1,conv([1 1],[1 1]));
% Sample time
Ts=1;
% Discrete-time transfer function
mod_tf_d = c2d(mod_tf_c,Ts);
% Discrete-time state-space model
mod_ss_d = ss(mod_tf_d);
% Matrix data A,B,C,D for the state-space model
[A,B,C,D]=ssdata(mod_ss_d);
% Sizes of the system (number of states, inputs and outputs)
[nx,nu]=size(B); [ny,nx]=size(C);
% Stochastic discrete-time state-space model
%       x[n+1] = Ax[n] + Bu[n] + Gw[n]             {State equation}
%       y[n]   = Cx[n] + Du[n] + Hw[n] + v[n]      {Measurements}
% with known inputs u, process noise w, measurement noise v
G=B; %G=ones(nx,1);
H=zeros(ny,1);
mod_ss_stoch=ss(A,[B G],C,[D H],Ts);
% Noise covariances
% E{ww'} = QN,     E{vv'} = RN
QN=0.1; RN=1;
```

```
% Discrete Kalman estimator
% x[n+1|n] = Ax[n|n-1] + Bu[n] + L(y[n] - Cx[n|n-1] - Du[n])
[KEST,L,P] = kalman(mod_ss_stoch,QN,RN);

% Solution by Control Toolbox
% Input 1 ... u
% Input 2 ... ym
% State-space representation
est_ss = ss(A-L*C,[B-L*D L],C,D,Ts);
% Transfer function representation
est_tf = tf(est_ss);
set(est_tf,'InputName',{'u','ym'},'OutputName','yp');

% Characteristic polynomial of Kalman estimator
charpol = poly(A-L*C);      % poly(eig(A-L*C));

% Solution by Symbolic Toolbox
z=sym('z');
est_x=inv(z*eye(nx)-A+L*C);
% Symbolic transfer function between u and yp
est_u=C*est_x*(B-L*D);
[num_est_u,den_est_u]=numden(est_u);
pol_den_est_u=sym2poly(den_est_u);
% Normalized numerator and denominator
num_est_un=vpa(num_est_u/pol_den_est_u(1),4);
den_est_un=vpa(den_est_u/pol_den_est_u(1),4);

% Symbolic transfer function between ym and yp
est_y=C*est_x*L;
[num_est_y,den_est_y]=numden(est_y);
pol_den_est_y=sym2poly(den_est_y);
% Normalized numerator and denominator
num_est_yn=vpa(num_est_y/pol_den_est_y(1),4);
den_est_yn=vpa(den_est_y/pol_den_est_y(1),4);

% Symbolic characteristic polynomial of Kalman estimator
charpol=vpa(det(z*eye(nx)-A+L*C),4);
% Polynomial coefficient vector
pol_charpol=sym2poly(charpol);
```

7 Conclusion

Importance of data filtering for practical applications is well known. Filter design for predictive controller is discussed in the paper. *C* polynomial of the process model is treated as a prefilter. This polynomial is called *T*-polynomial in GPC controller terminology and in our case it is designed as a Kalman filter characteristic polynomial. Two examples are given to demonstrate the prediction ability of the model with and without data filtering. The benefit of the data filtering for plant output prediction is clearly seen from the simulated experiments. Disadvantage is that the order of the filter is equal to the number of the state variables – for higher order systems or multidimensional system approximation filter with lower order should be sufficient.

Acknowledgments

The research has been supported in the project of Czech Republic MSM 0021627505. This supports is very gratefully acknowledged.

References

[1] D.W. CLARKE, C. MOHTADI, P.S. TUFFS. Generalized Predictive Control – Part I. The Basic Algorithm. *Automatica*, vol. 23, no. 2, p. 137-148, 1987.

[2] D.W. CLARKE, C. MOHTADI, P.S. TUFFS. Generalized Predictive Control – Part II. Extensions and Interpretations. *Automatica*, vol. 23, no. 2, p. 149-160, 1987.

[3] D.W. CLARKE, C. MOHTADI. Properties of Generalized Predictive Control. *Automatica*, vol. 25, no. 6, p. 859-875, 1989.

[4] E.F. CAMACHO, C. BORDONS. *Model Predictive Control*. London: Springer-Verlag London Limited, Great Britain, 280 pp., 2002.

Ing. Daniel Honc, Ph.D.
Institute of Electrical Engineering and Informatics, University of Pardubice, Czech Republic
nám. Čs. legií 565, 532 10 Pardubice
tel: +420 466 037 107 and e-mail: daniel.honc@upce.cz

Doc. Ing. František Dušek, CSc.
Institute of Electrical Engineering and Informatics, University of Pardubice, Czech Republic
nám. Čs. legií 565, 532 10 Pardubice
tel: +420 466 037 125 and e-mail: frantisek.dusek@upce.cz