Real-world comparison of predictive controllers based on internal and external description of the controlled system

Daniel Honc[0000-0001-8440-5272] and Aleš Novotný[0009-0002-9430-7101]

University of Pardubice, Faculty of Electrical Engineering and Informatics
nám. Čs. Legií 565, 530 02 Pardubice, Czech Republic
daniel.honc@upce.cz

Abstract. Predictive control is a very general method of designing an optimal controller, allowing the use of different criteria, but also a large number of different models of the controlled system. In this paper, the derivation and real application of two types of predictive controller are shown - for internal description (expressed by a discrete-time state-space model) and for external description (expressed by discrete-time transfer function). In an ideal simulation both controllers give identical results, but in real conditions different behavior will be displayed. The aim of this paper is to demonstrate the differences between the two solutions from an application point of view, to discuss the strengths and weaknesses of both types of predictive controllers and to compare their behavior in real conditions. The result is not the "best" controller but understanding their differences and facilitating the selection of the right form for a specific application.

Keywords: Predictive Controller, State-space Model, Transfer Function, DC Motor Speed Control.

1 Introduction

The feed-forward control responds with a control action to the setpoint or to disturbances. The feedback control (regulation) operates with the setpoint and the measured controlled variable or only with the control error. In the case of state-space controllers, feedback from the state variables (measured or estimated) is used, and to ensure the integrative character, the circuit is supplemented by an I controller operating with a control error. Predictive control computes the optimal future control action (including the current one) given a criterion that takes into account the future control errors at a finite horizon. The future controlled variables (controlled outputs of the system) are predicted (estimated) based on the knowledge of the controlled process dynamic model, the system state and the calculated future control actions (including the current one). From this perspective, it would be forward control. Since the model is never completely accurate and various disturbances affect the system, the concept of a sliding horizon is used. At each sampling interval, the state of the system is estimated (or measured) and the optimal control actions are calculated. Only the current control action is implemented and the whole algorithm is repeated at the next interval. Thus, information about the measured system output (measured state) is fed into the algorithm via the state estimation (or measurement) and some feedback is provided.

A discrete-time dynamic model of the controlled system is used to predict the controlled variable. In the industrial applications of predictive controllers [1,2], models in the form of finite impulse response [3] and finite step response [4,5] were initially used. Generalized Model Predictive Control is a version where a model in the form of transfer function or state-space description is used [6,7]. Often the controlled future variables are computed as the sum of two separate responses - forced and free. The forced response is related to the control actions. On the other hand, the free response depends on system state (initial conditions) and disturbance variables (system inputs not used for the control). Any type of approximation of the dynamic behavior of the controlled system (even nonlinear) can be used to calculate the free response. For the forced response, models in the form of an internal description (state-space model) and in the form of an external description (transfer function) are most commonly used. Both variants give the same results in the case of an ideal simulation (the behaviour of the model coincides with the behaviour of the controlled system, the system is not affected by disturbances, there is no measurement noise on the controlled variables). In real conditions, both variants differ, both in terms of the real application (the way the free response is calculated) and in terms of the resulting behaviour. The derivation of both types of controllers and their application to a DC motor speed control system is shown in this paper.

The paper is structured as follows. Chapter 2 presents the derivation of both versions of the predictive controller. In chapter 3, the real system and its model are described. Chapter 4 shows the control results and chapter 5 presents the conclusions and discussion.

2 Derivation of the predictive controller

2.1 Cost function

In the cost function (criterion), weighted sums of the squares of future control errors and future changes in control actions (including the current one) is considered

$$J = r\sum_{j=1}^{N_2}\big(y(k+j) - w(k+j)\big)^2 + \mathrm{q}\sum_{j=1}^{N_u}\Delta u(k+j-1)^2 =$$

$$(\mathbf{Y}-\mathbf{W})^T\mathbf{R}(\mathbf{Y}-\mathbf{W}) + \mathbf{U}^T\mathbf{Q}\mathbf{U} \qquad (1)$$

where

$$\mathbf{Y} = \begin{bmatrix} y(k+1) \\ y(k+2) \\ y(k+3) \\ \vdots \\ y(k+N_2) \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} w(k+1) \\ w(k+2) \\ w(k+3) \\ \vdots \\ w(k+N_2) \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix},$$

$$\mathbf{R} = diag(r) \text{ and } \mathbf{Q} = diag(q).$$

If the control actions are not penalized, a theoretically ideal desired value (setpoint) tracking, but practically unusable solution (very sensitive to model inaccuracies) will be obtained. If absolute control action is used, the controller would leave a permanent control error. By weighting between a term of future control errors and future control action changes (moves or speed), the rate of setpoint tracking (disturbance rejection) and, in turn, the robustness of the controller can be tuned. With a larger penalization of the control action changes, the control quality may not be as high, but the controller will be robust enough to work for a less accurate model or for a nonlinear system. Two horizons will be considered: the horizon for setpoint tracking N_2 and the control horizon N_u. Longer setpoint tracking horizon can be chosen to stabilize the system and use a shorter control horizon to reduce computational complexity - after N_u control actions, the manipulated variable remains constant (its changes are zero). To minimize the cost function, it is necessary to know the relationship between the future inputs **U**, the current state $\mathbf{x}(k)$ and the future outputs of the system **Y**. This relationship is called as a predictor.

2.2 Predictor with an internal description model

The model of a one-dimensional system (Single-Input Single-Output SISO) in the form of a discrete-time state-space model is

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathrm{u}(k)$$
$$\mathrm{y}(k) = \mathbf{C}\mathbf{x}(k) \tag{2}$$

For the system model, input the change in control action $\Delta u(k)$ - as in the criterion (1) is used. A summator with the input $\Delta u(k)$ is assigned before the system with input u(k). The model must contain information about the past control action and therefore $u(k\text{-}1)$ becomes the next state variable of the extended state vector $\mathbf{x}_\mathrm{p}(k)$. This gives us a model of a system with an extended state

$$\underbrace{\begin{bmatrix} \mathbf{x}(k+1) \\ \mathrm{u}(k) \end{bmatrix}}_{\mathbf{x}_p(k+1)} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0}_{1\times nx} & 1 \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ \mathrm{u}(k-1) \end{bmatrix}}_{\mathbf{x}_p(k)} + \underbrace{\begin{bmatrix} \mathbf{B} \\ 1 \end{bmatrix}}_{\mathbf{N}} \Delta\mathrm{u}(k)$$
$$\mathrm{y}(k) = \underbrace{[\mathbf{C} \quad 0]}_{\mathbf{O}} \underbrace{\begin{bmatrix} \mathbf{x}(k) \\ \mathrm{u}(k-1) \end{bmatrix}}_{\mathbf{x}_p(k)} \tag{3}$$

Output estimation (prediction) errors are not taken into account at the model level and in the derivation of the prediction equations, but in the estimation of the state by the observer. Output prediction in matrix form - the predictor has the form [6,7]

$$\underbrace{\begin{bmatrix} \mathrm{y}(k+1) \\ \mathrm{y}(k+2) \\ \mathrm{y}(k+3) \\ \vdots \\ \mathrm{y}(k+N_2) \end{bmatrix}}_{\mathbf{Y}} =$$

$$\underbrace{\begin{bmatrix} \mathbf{ON} & 0 & 0 & \cdots & 0 \\ \mathbf{OMN} & \mathbf{ON} & 0 & \cdots & 0 \\ \mathbf{OM}^2\mathbf{N} & \mathbf{OMN} & \mathbf{ON} & \cdots & 0 \\ & & \vdots & & \\ \mathbf{OM}^{N_2-1}\mathbf{N} & \mathbf{OM}^{N_2-2}\mathbf{N} & \mathbf{OM}^{N_2-3}\mathbf{N} & \cdots & \mathbf{OM}^{N_2-N_u}\mathbf{N} \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} \Delta\mathrm{u}(k) \\ \Delta\mathrm{u}(k+1) \\ \Delta\mathrm{u}(k+2) \\ \vdots \\ \Delta\mathrm{u}(k+N_u-1) \end{bmatrix}}_{\mathbf{U}} +$$

$$\underbrace{\begin{bmatrix} \mathbf{OM} \\ \mathbf{OM}^2 \\ \mathbf{OM}^3 \\ \vdots \\ \mathbf{OM}^{N_2} \end{bmatrix}}_{\mathbf{F}_p} \mathbf{x}_p(k) = \underbrace{\mathbf{GU}}_{\text{forced response}} + \underbrace{\mathbf{F}_p\mathbf{x}_p(k)}_{\text{free response } \mathbf{f}} \tag{4}$$

2.3 Predictor with an external description model

The model of the SISO system contains a disturbance model – see Fig. 1 (random walk of prediction errors of the output correlated by the proportion of polynomials $C/\Delta A$). The consequence of using the Δ operator is that the resulting controller will have integral character (the $1/\Delta$ transfer is a discrete-time integrator) [6,7]. The choice of the polynomial C is equivalent to the choice of the dynamics of the state observer and can thus influence the sensitivity of the controller to measurement noise.

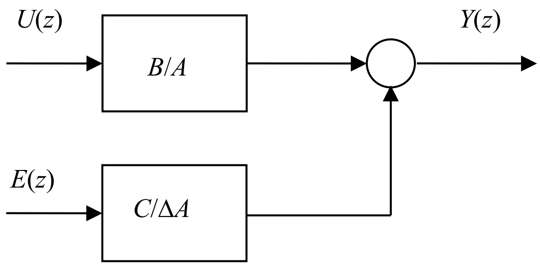

Fig. 1. Block diagram of the external model.

The model of the system in the transformed domain (discrete-time transfers functions and images of discrete-time variables) can be written as

$$Y(z) = \frac{B}{A}U(z) + \frac{C}{\Delta A}E(z) \tag{5}$$

where

$\Delta = 1 - z^{-1}$ is image (operator) of time increment (change, difference),
e(k)=y(k)-y(k|k-1) is controlled variable estimation error and
$A = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}$, $B = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}$, $C = 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots + c_{nc} z^{-nc}$ are polynomials of the model (n is the order of the model, n_c is the order of the filter polynomial C).

If the model of the system is multiplied by the product ΔA, the model is

$$\underbrace{\Delta A}_{\tilde{A}} Y(z) = B\Delta U(z) + CE(z)$$

$$\tilde{A} = \Delta A = 1 + \tilde{a}_1 z^{-1} + \tilde{a}_2 z^{-2} + \cdots + \tilde{a}_n z^{-n} + \tilde{a}_{n+1} z^{-(n+1)} \tag{6}$$

The prediction equations can be written in matrix form. Only the current and past estimation errors are taken into account. All future estimation errors are assumed to be zero - that the estimated controlled variable is equal to the measured one.

$$\underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ \tilde{a}_1 & 1 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{\mathbf{A}_p} \underbrace{\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N) \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} b_1 & 0 & \cdots & 0 \\ b_2 & b_1 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & b_1 \end{bmatrix}}_{\mathbf{B}_p} \underbrace{\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N-1) \end{bmatrix}}_{\mathbf{U}} +$$
$$\underbrace{\begin{bmatrix} -\tilde{a}_1 & -\tilde{a}_2 & \cdots & -\tilde{a}_{n+1} \\ -\tilde{a}_2 & -\tilde{a}_3 & \cdots & 0 \\ & & & \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{A}_m} \underbrace{\begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-n) \end{bmatrix}}_{\mathbf{Y}_m} + \underbrace{\begin{bmatrix} b_2 & b_3 & \cdots & b_n \\ b_3 & b_4 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{B}_m} \underbrace{\begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-n+1) \end{bmatrix}}_{\mathbf{U}_m} +$$
$$\underbrace{\begin{bmatrix} c_1 & c_2 & \cdots & c_{n_c} \\ c_2 & c_3 & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{C}_m} \underbrace{\begin{bmatrix} e(k) \\ e(k-1) \\ \vdots \\ e(k-n_c+1) \end{bmatrix}}_{\mathbf{E}_m} \tag{7}$$

Multiplying $\mathbf{A}_p^{-1}$ from the left side gives the output predictions in matrix form - the predictor [7,8]

$$\underbrace{\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N) \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\mathbf{A}_p{}^{-1}\mathbf{B}_p}_{\mathbf{G}} \underbrace{\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N-1) \end{bmatrix}}_{\mathbf{U}} + \underbrace{\mathbf{A}_p{}^{-1}\mathbf{A}_m}_{\mathbf{F}_y} \underbrace{\begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-n) \end{bmatrix}}_{\mathbf{Y}_m} +$$

$$\underbrace{\mathbf{A}_p{}^{-1}\mathbf{B}_m}_{\mathbf{F}_u} \underbrace{\begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-n+1) \end{bmatrix}}_{\mathbf{U}_m} + \underbrace{\mathbf{A}_p{}^{-1}\mathbf{C}_m}_{\mathbf{F}_e} \underbrace{\begin{bmatrix} e(k) \\ e(k-1) \\ \vdots \\ e(k-n_c+1) \end{bmatrix}}_{\mathbf{E}_m} = \mathbf{GU} + \mathbf{F}_y\mathbf{Y}_m + \mathbf{F}_u\mathbf{U}_m +$$

$$\mathbf{F}_e\mathbf{E}_m = \underbrace{\mathbf{GU}}_{\text{forced response}} + \underbrace{\underbrace{[\mathbf{F}_y \quad \mathbf{F}_u \quad \mathbf{F}_e]}_{\mathbf{F}_p} \cdot \underbrace{\begin{bmatrix} \mathbf{Y}_m \\ \mathbf{U}_m \\ \mathbf{E}_m \end{bmatrix}}_{\mathbf{x}_p}}_{\text{free response } \mathbf{f}} \tag{8}$$

The forced response is the portion of the output of the system **Y** caused by control moves in the vector **U** if the current state $\mathbf{x}_\mathrm{p}$ were zero. The free response **f** is the part of the output of system **Y** induced by the state $\mathbf{x}_\mathrm{p}$ if the control action changes in vector **U** were zero (the control actions would not change). The forced response responds to control action changes, while the free response depends on the current state of the system. If there is no change in the control actions, the system will only react with a free response at the output.

2.4 Cost function minimization

The goal is to calculate future changes of control actions (including the current one) **U** that will lead to the minimum value of the cost function *J* at each sampling interval. If the predictor $\mathbf{Y} = \mathbf{GU} + \mathbf{f}$ is substituted into the cost function (1), the result is

$$J = (\mathbf{GU} + \mathbf{f} - \mathbf{W})^T\mathbf{R}(\mathbf{GU} + \mathbf{f} - \mathbf{W}) + \mathbf{U}^T\mathbf{QU} = (\mathbf{U}^T\mathbf{G}^T + \mathbf{f}^T - \mathbf{W}^T)\mathbf{R}(\mathbf{GU} + \mathbf{f} - \mathbf{W}) + \mathbf{U}^T\mathbf{QU} = \mathbf{U}^T \underbrace{(\mathbf{G}^T\mathbf{RG} + \mathbf{Q})}_{\mathbf{H}} \mathbf{U} + \mathbf{U}^T \underbrace{\mathbf{G}^T\mathbf{R}(\mathbf{f} - \mathbf{W})}_{\mathbf{g}} + \underbrace{(\mathbf{f} - \mathbf{W})^T\mathbf{RG}}_{\mathbf{g}^\mathrm{T}} \mathbf{U} + \underbrace{(\mathbf{f} - \mathbf{W})^T\mathbf{R}(\mathbf{f} - \mathbf{W})}_{k} \tag{9}$$

When **R** is a symmetric matrix, then $\mathbf{R}^T$=**R** and the above substitution for **g** and $\mathbf{g}^T$ can be used, and since the individual terms of the criterion are scalars, it holds $\mathbf{U}^T\mathbf{g} + \mathbf{g}^T\mathbf{U} = 2\mathbf{g}^T\mathbf{U}$.

The cost function is a quadratic form (multidimensional quadratic function)

$$J = \mathbf{U}^T\mathbf{HU} + 2\mathbf{g}^T\mathbf{U} + k \tag{10}$$

Analytical solution

In the absence of constraints, the minimization problem can be solved analytically. Vector derivative of the cost function J by $\mathbf{U}$ is performed

$$\frac{dJ}{d\mathbf{U}} = 2\mathbf{HU} + 2\mathbf{g} \tag{11}$$

and set it equal to a zero vector of dimension equal to the number of control action changes N_u

$$2\mathbf{HU} + \mathbf{2g} = \mathbf{0} \tag{12}$$

Equation (12) is solved by multiplying $\mathbf{H}^{-1}$ from the left side

$$\mathbf{U} = -\mathbf{H}^{-1}\mathbf{g} = \underbrace{(\mathbf{G}^T\mathbf{RG} + \mathbf{Q})^{-1}\mathbf{G}^T\mathbf{R}}_{\mathbf{L}}(\mathbf{W} - \mathbf{f}) = \mathbf{L}(\mathbf{W} - \mathbf{f}) \tag{13}$$

In order to be a minimum, the matrix $\mathbf{H}$ must be positively semidefinite (just as the coefficient a of a quadratic function must be positive nonzero to have a minimum).

Numerical solution

In case of constraints (on control actions, states or controlled variables) it is necessary to solve the problem numerically using numerical optimization - application of quadratic programming. The result is a vector of optimal future (and current) changes of the control actions $\mathbf{U}$. Constraints in the form of inequality, equality and maximum and minimum values can be considered [9]. Constraints in the form of inequalities $\mathbf{AU} \leq \mathbf{b}$ can be used to define constraints on the control action, state and controlled variables.

2.5 Control law

This subsection presents results for the unconstrained case where the analytical solution could be used. In the case that the control is constrained, for example by a control action, the analytical solution is not valid in the sense of an optimal solution and numerical methods that respect these constraints must be used. In principle, the behaviour of the controller is very similar, but in case of constraints activation, slightly different control actions can be generated.

The optimal control action change is

$$\Delta u(k) = \mathbf{K}(\mathbf{W} - \mathbf{f}) \tag{14}$$

where $\mathbf{K}$ is the first row of the matrix $\mathbf{L}$ and the optimal control action is

$$u(k) = \Delta u(k) + u(k-1) \tag{15}$$

A predictive controller does not react to current, past and previous control errors, such as a PID controller. The control law can be expressed as a negative feedback with gain **K** responding to the difference between the future desired value **W** and the free response of the system **f**. The free response **f** is a function of the system state used for the prediction

$$\mathbf{f} = \mathbf{F}_p \mathbf{x}_p(k) \tag{16}$$

For the internal version, the state of the system used for prediction is the state of the system augmented by the past control action

$$\mathbf{x}_p(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathrm{u}(k-1) \end{bmatrix} \tag{17}$$

For the external version, the state of the system used for predictions is determined by past values of output, control action changes, and output estimation errors

$$\mathbf{x}_p = \begin{bmatrix} \mathbf{Y}_m \\ \mathbf{U}_m \\ \mathbf{E}_m \end{bmatrix} \tag{18}$$

The control law can be drawn as the block diagram in Fig. 2

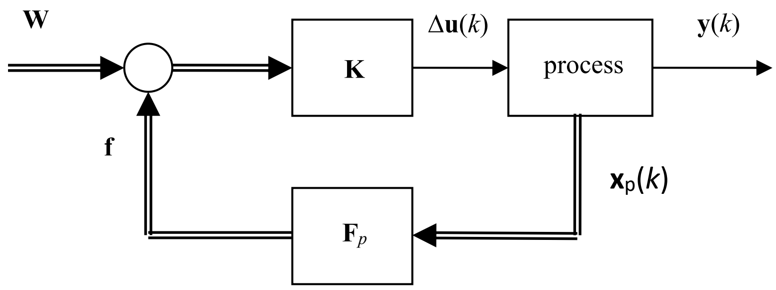

Fig. 2. Block diagram of the predictive controller.

If the system behaves in the future without changes in the control action as required, the vector of future control action changes **U** will be zero, the current control action change will also be zero, and the current control action will remain at its past value. If a deviation occurs between the desired value of **W** and the free response of the system **f**, the control action change will be proportional to this deviation. The less the variation of the control actions in the criterion is penalized, the stronger the gain **K** will come out. Although the control response will be faster and theoretically of better quality, in practice it will be more sensitive to model quality and to measurement noise, and the robustness of the resulting controller will be reduced.

The parameters of the controller must be tuned with respect to its real behavior. Sometimes the system will get closer to the theoretical behaviour and sometimes the difference between the expected and real behaviour will be larger. It is not only a question of the quality of the model, but, for example, for lower order systems the situation will be more favourable than for higher orders or for systems with time delays.

3 Controlled process

The aim will be to control the speed of the DC motor of the GUNT RT060 system [10] – see Fig. 3. The voltage of the DC motor (signal Y) is used to control its speed. The motor is coupled to a generator and a flywheel is placed on the axis to increase the dynamics. The speed (signal X) is measured by an inductive sensor. The generator (signal Z) can be used to vary the load.

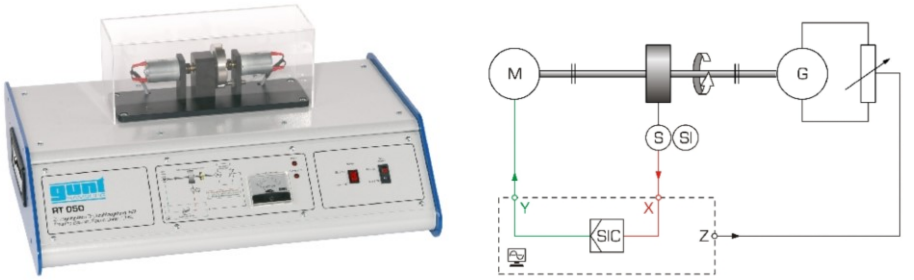

Fig. 3. RT 050 system - speed control [10].

First, it is necessary to perform an identification experiment - to excite the system and record its response. The model is found in terms of the optimal match between the measured and simulated output of the system for the same initial conditions and input. MATLAB function *lsim* is used to simulate model response and *fminsearch* to find the optimal parameters. We choose a description in the form of a second order proportional system. The identification results in the following continuous-time transfer function

$$F(s) = \frac{k}{(T_1 s+1)(T_2 s+1)} = \frac{2.04}{(1.61s+1)(1.16s+1)} \tag{19}$$

The time courses of the input variable used to excite the system, the measured and simulated output of the system are shown in Fig. 4.

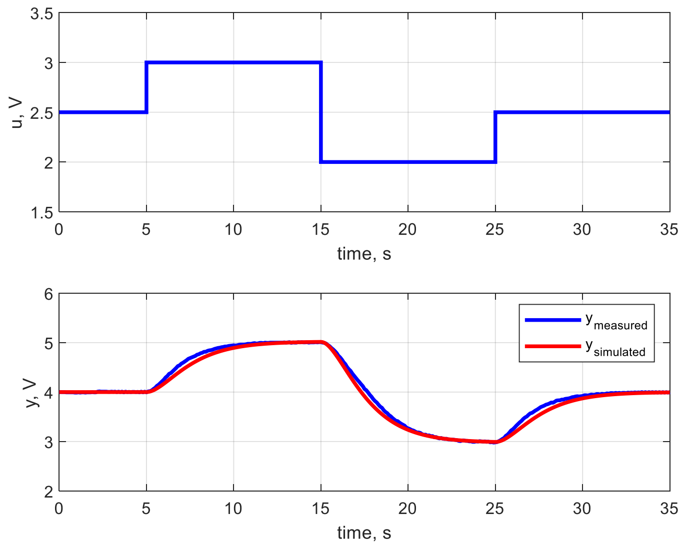

Fig. 4. Time course of variables during identification.

4 Control experiments

Both types of predictive controller are applied to the RT 050 system and the control results are compared and discussed. In both cases, the analytical forms of the controllers are used (constraints are not considered). The two versions differ in the way the free response **f** is calculated. For the internal description, the system state is augmented with the past control action. The system state is estimated by a state observer. For the external description, state vector in the form of output, input changes, and output estimation error history is used. The parameters of both controllers are given in Tab. 1.

Table 1. Controller parameters.

Parameter	Symbol	Value	Unit
Sampling period	T_s	0.1	s
Desired value monitoring horizon	N_2	30	-
Control horizon	N_u	30	-
Control deviation penalty	r	1	-
Penalisation of changes in the action variable	q	1	-

4.1 Controller with an internal description model

The setting of the state observer can be seen as tunable parameter of the controller that influences its behavior. Two control experiments were measured for different observer settings. The control response for the discrete-time observer poles [0.3 0.3] is shown in Fig. 5 and for the observer poles [0.8 0.8] is shown in Fig. 6. Variable $y_{predicted}$ is predicted output calculated from the estimated state.

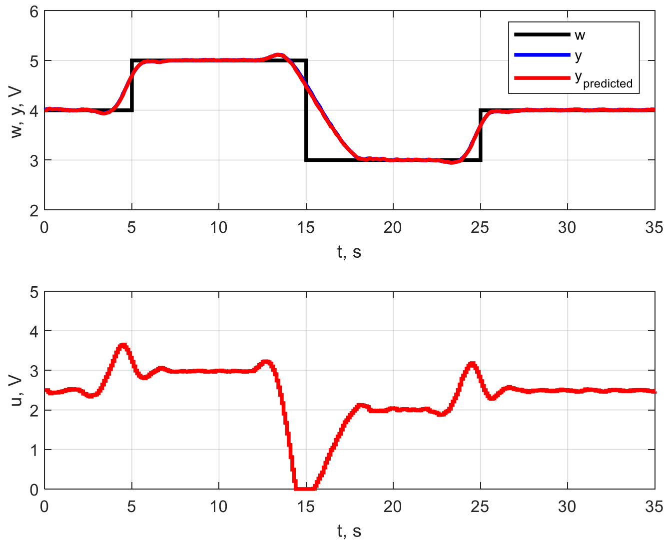

Fig. 5. Control by the state-space controller and observer poles [0.3 0.3].

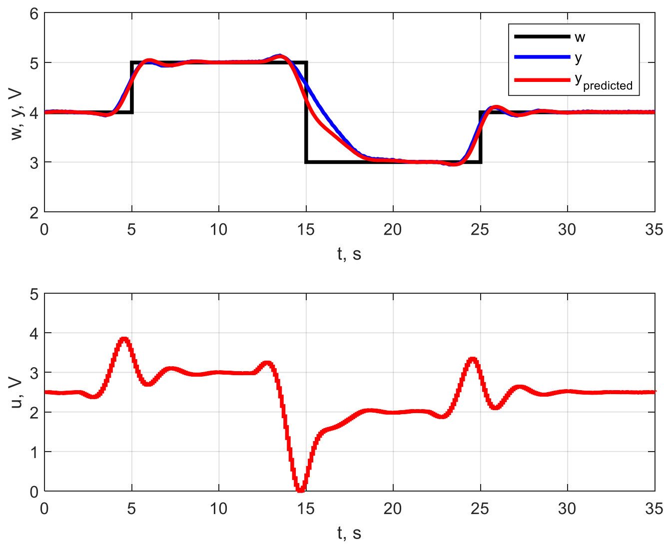

Fig. 6. Control by the state-space controller and observer poles [0.8 0.8].

4.2 Controller with an external description model

Tunable parameter is the filter polynomial *C*. Two control responses for different filter polynomial settings were measured. The control response for the polynomial *C* = 1 is shown in Fig. 7 and for the polynomial *C* = 1-0.8z-1 is shown in Fig. 8.

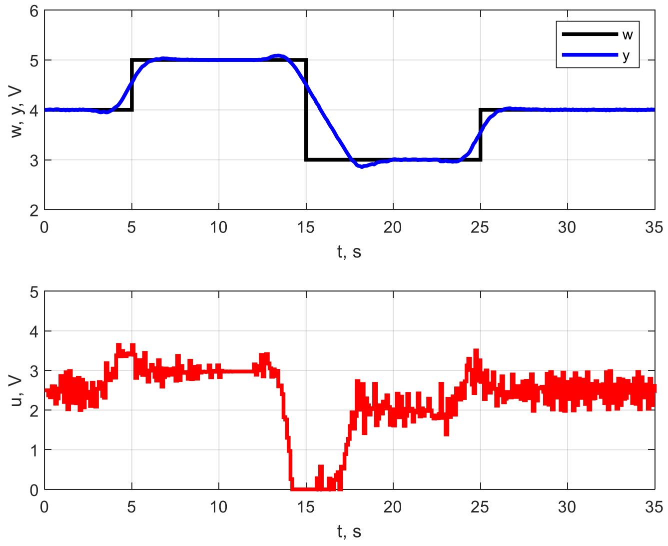

Fig. 7. Control by the transfer function controller and polynomial C=1.

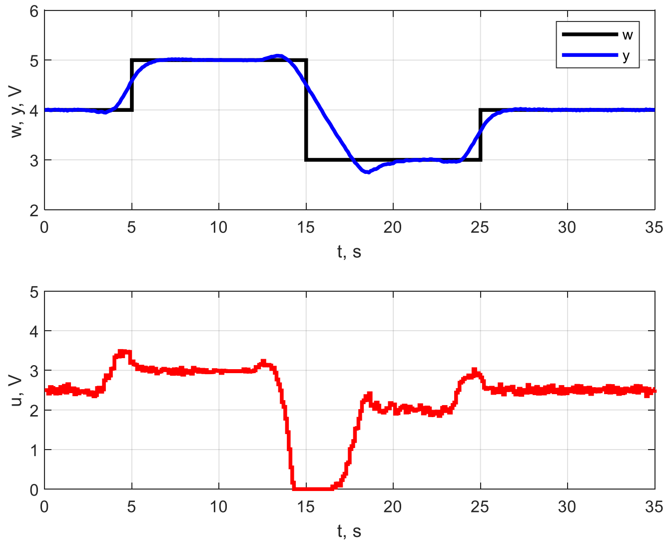

Fig. 8. Control by the transfer function controller and polynomial $C=1-0.8z^{-1}$.

5 Conclusion

The aim of the paper was to present the derivation of two types of predictive controller - one based on an internal description of the controlled system and the other using an external description. Both controllers exhibit different behaviors in real applications. While the external description requires an observer of the state, the second variant uses only the outputs, control changes and prediction errors history. The controller based on external description is integrative in nature - it does not leave a permanent control error. Conversely, an internal description-based controller can result in permanent control error when the state observer is not set properly. For observers with faster dynamics, some filtering ability of the observer is lost (the control actions are not smoothed as possible), but the permanent control error will be eliminated. The controller settings form a separate challenge of the predictive control problem. In a real application, a certain trade-off has to be chosen between the quality of the control and the robustness of the resulting solution. For lower order systems it is able to get closer to the theoretical results. For higher order systems and specially for systems with time delay and non-minimum phase, nature limits us and we have to accept settings that give a certain lower quality of control, but the controllers will be sufficiently robust and thus practically usable. A major advantage of a predictive controller based on an internal description can be the respect of the constraints on the state variables, which is not possible in principle for a controller with an external description. Especially for models obtained by the analytical procedure of applying energy and mass balances, where the state variables have physical meaning, this feature can be very interesting. The question is the accuracy of the estimation of the state variables - whether the constraints are applied with realistic accuracy. Another advantage is the seamless filtering of the state variables, which leads to a quieter course of the control action even when the controller has a relatively large gain or significant measurement noise is present. The extension of the controller for multidimensional systems is straightforward - for the predictor, all relationships remain the same, but only the matrix dimensions change. The advantage of a controller with a model in the form of an external description is the integrative nature, which automatically eliminates model mismatch and disturbance effects. However, for stronger filtering settings, the quality of the control response is reduced. The extension to multidimensional systems is not as straightforward as the internal description - one has to work with a transfer function matrix, which is not as algorithmically convenient as the state-space description. In the end, however, the choice of the model used is mainly a matter of the target application and the personal preference of the control designer.

Acknowledgment

This research was supported by grant SGS_2024 at the Faculty of Electrical Engineering and Informatics, University of Pardubice.

References

1. Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology, Control Engineering Practice, 11 (7), pp. 733 - 764, DOI: 10.1016/S0967-0661(02)00186-7 (2003).
2. García, C.E., Prett, D.M., Morari, M.: Model predictive control: Theory and practice-A survey, Automatica, 25 (3), pp. 335 - 348, DOI: 10.1016/0005-1098(89)90002-2 (1989).
3. Prasath, G., Jørgensen, J.B.: Model predictive control based on finite impulse response models Proceedings of the American Control Conference, art. no. 4586531, pp. 441 - 446, DOI: 10.1109/ACC.2008.4586531 (2008).
4. Ding, B.-C., Chen, Q., Xie, Y.-J.: Predictive control oriented identification of finite step response model, Kongzhi yu Juece/Control and Decision, 31 (11), pp. 2030 - 2036, DOI: 10.13195/j.kzyjc.2015.1307 (2016).
5. Tange, Y., Kiryu, S., Matsui, T.: Fast model predictive control based on formula manipulation and finite step prediction, IEEJ Transactions on Electronics, Information and Systems, 140 (2), pp. 219 - 226, DOI: 10.1541/ieejeiss.140.219 (2020).
6. Camacho, E.F., Bordons, C.: Model Predictive Control, Springer (2007).
7. Honc, D., Sharma, K.R., Abraham, A., Dušek, F., Pappa, N.: Teaching and Practicing Model Predictive Control, IFAC-PapersOnLine, 49 (6), pp. 34-39 (2016).
8. Rossiter, J.A.: Model-based Predictive Control – A Practical Approach, CRC Press (2003).
9. Maciejowski, J.: Predictive Control with Constraints, Prentice Hall (2002).
10. GUNT RT 050 Training system: speed control, http://www.gunt.de/index.php?option=com_gunt&task=gunt.list.category&lang=en&category_id=83&product_id=1046, last accessed 2024/01/01.