# ENHANCING ARX-MODEL BASED MPC BY KALMAN FILTER AND SMOOTHER

**Lubomír Baramov[*] and Vladimír Havlena[*, **]**

[*]*Honeywell Prague Laboratory, Pod vodárenskou věží 4, Prague, Czech Republic*

[**]*Faculty of Electrical Engineering, Czech Technical University, Technická 2, Prague, Czech Republic*

Abstract: An approach to enhancing a model-based predictive controller by Kalman filter is proposed. The controller uses an ARX process model and the structure of the controller is assumed fixed; some of its internal variables – past values of controlled variables (output history) are accessible and can be modified to achieve better performance in disturbance attenuation and noise rejection. We present an algorithm of updating the output history using Kalman filter to achieve predictions equivalent to those of the state-space model, thus overcoming the limitations of the ARX predictor. Interesting relations of this algorithm to Kalman interval smoother are given. 

Keywords: Kalman filter, smoothing, predictive control, disturbance rejection.

## 1. INTRODUCTION

Model-based predictive control (MPC) is a concept, which has made a significant impact on advanced process control. It can be briefly described as follows: A sequence of future control actions is computed to optimise the future process behaviour over a finite time-horizon subject to various constraints. Further, only the first action of the sequence is implemented, and the optimisation is repeated on the horizon shifted one step forward. Particular implementations of MPC differ in the optimisation problem formulation as well as in the model used for predicting the future process behaviour. Prediction models in MPC can be either input-output or state-space based. The latter are more flexible in capturing the real process structure, noises and uncertainties; they are favoured in theoretical research (Bemporad and Morari, 1999). The input-output models are used in most real-world applications due to a simpler representation; there is also some inertia in the industry and state-space models have not been fully accepted yet. We shall consider MPC of (Havlena and Findejs, 2005); it uses the ARX model that is very economical from the point of view of the on-line computational effort and data storage. However, it assumes a noise model that is not always realistic – e.g., if there is a significant sensor noise, the predictions exhibit large variations, which propagate to control actions. In that case, the output error model, or a state-space model and Kalman filter (*KF*) would be appropriate. However, major changes of the MPC engine are costly; KF is thus used as an incremental improvement for enhancing the current controller, in particular, for estimating unknown inputs. We propose an algorithm, which uses KF also to modify stored output values used by ARX so that the predictions are equal to those of a state-space model. A recursive formula for updating the output history is found, similar to Kalman interval smoother, *KIS* (Anderson and Moore, 1979). A new convergence result is obtained for KIS.

## 2. THE BACKGROUND

### *2.1 ARX and state-space models*

MISO (Multi-Input-Single-Output) ARX models are used for predicting the system output as follows:

$$\hat{y}_m(k|k-1) = -\sum_{i=1}^{s_m} a_{mi} y_m(k-i) + \sum_{r=1}^{l}\sum_{i=1}^{s_m} b_{mri} u_r(k-i-n_{dr}) + e_m(k) \tag{1}$$

where $y_m$ is a (measured) output variable and $\hat{y}_m$ its prediction, $m=1,...,p$; $u_r$ is the input (at this point we do not distinguish between the manipulated and the disturbance variables), $e_m$ is Gaussian white noise and $n_{dr}$ is time-delay for the $r^{\text{th}}$ input. Further let

$$\left[\frac{Y_p^{(m)}(k)}{\hat{Y}_f^{(m)}(k)}\right] = \begin{bmatrix} y_m(k-s_m) \\ \vdots \\ y_m(k-1) \\ \hline \hat{y}_m(k|k-1) \\ \vdots \\ \hat{y}_m(k+N|k-1) \end{bmatrix}, \left[\frac{U_p^{(r)}(k)}{U_f^{(r)}(k)}\right] = \begin{bmatrix} u_r(k-n_{dr}-s_m) \\ \vdots \\ u_r(k-1) \\ \hline u_r(k) \\ \vdots \\ u_r(k+N-1) \end{bmatrix} \tag{2}$$

The formula for expected future outputs is given by

$$\hat{Y}_f^{(m)}(k) = \tilde{A}_f^{-1}\left(-\tilde{A}_p^{(m)} Y_p^{(m)}(k) + \sum_r \left(\tilde{B}_p^{(m,r)} U_p^{(r)}(k) + \tilde{B}_f^{(m,r)} U_f^{(r)}(k)\right)\right) \tag{3}$$

where the parameter matrices are given by

$$\left[\tilde{A}_p^{(m)} \mid \tilde{A}_f^{(m)}\right] = \left[\begin{array}{ccc|cccc} a_{s_m}^m & \cdots & a_1^m & 1 & & & \\ & \ddots & \vdots & \vdots & \ddots & & \\ & & a_{s_m}^m & a_{s_m-1}^m & \cdots & 1 & \\ & & & a_{s_m}^m & \cdots & a_1^m & 1 \\ & & & & \ddots & \ddots & \ddots \end{array}\right] \tag{4}$$

$$\left[\tilde{B}_p^{(m,r)} \mid \tilde{B}_f^{(m,r)}\right] = \left[\begin{array}{ccc|ccc} b_{s_m}^{m,r} & \cdots & b_1^{m,r} & & & \\ & \ddots & & \ddots & & \\ & & b_{s_m}^{m,r} & \cdots & b_1^{m,r} & \\ & & & \ddots & & \ddots \\ & & & b_{s_m}^{m,r} & \cdots & b_1^{m,r} \\ & & & & \ddots & \ddots \end{array}\right] \tag{5}$$

The dimensions are compatible with (2). The future inputs are either subject to optimisation (manipulated variables), or predicted using an appropriate model, e.g., $u_r(k+i) = u_r(k)$, $i=1,...,N$ (disturbances).

The process can be represented in the state-space as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k) \\ y(k) &= Cx(k) + n(k) \end{aligned} \tag{6}$$

where $x \in R^n$ is the state, $u \in R^l$ is the input (assumed known) and $y \in R^p$ is the output. Vector sequences $v(k)$ and $n(k)$ are zero-mean Gaussian white noises (referred to as process and measurement or sensor noises, respectively), independent of each other, their variances being $E\left[vv^T\right] = Q$ and $E\left[nn^T\right] = R$. The expected future outputs of the plant are

$$\hat{Y}_f(k) = Mx(k) + NU_f(k), \tag{7}$$

$$M = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^N \end{bmatrix}, N = \begin{bmatrix} 0 & \cdots & 0 \\ CB & & 0 \\ \vdots & \ddots & \\ CA^{N-1}B & \ldots & CB \end{bmatrix}, U_f(k) = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix} \tag{8}$$

State-space model (6) is Multi-Input-Multi-Output (MIMO), whereas ARX in (1) is a collection of MISO models. The state-space model is inherently more flexible; it can reflect the physical structure of the plant and better absorb unmeasured noises, disturbances and uncertainties. Hence, it usually produces better predictions than ARX.

Prediction formula (7) is not directly usable, as the process state is not directly measurable. For this, KF is used for obtaining state estimates. KF can also be used for estimating an unknown disturbance. That situation cannot be handled by the ARX predictor alone especially, if the process is unstable. More on unknown input estimation and connections to the ARX predictor is in Section 3.2.

### *2.2 Kalman filter: an overview*

The concept of KF is described in detail in many monographs; see e.g. (Anderson and Moore, 1979), (Grewall and Andrews, 2001). Assume the plant model as in (6). Let us denote the set of data known up to time $t$ as $\mathcal{D}^t = \{u_0, \ldots, u_t, y_0, \ldots, y_t\}$. KF is associated with the problem of finding conditional probability of the state $x(k)$ if $\mathcal{D}^t$ is known; for $k > t$ this problem is called *prediction*, for $k = t$ it is *filtering* and for $k < t$ *smoothing*. For the above model, all probabilities are Gaussian; denote

$$p(x(k)|\mathcal{D}^t) = N\left(\hat{x}(k|t), P(k|t)\right) \tag{9}$$

KF gives formulas for conditional means $\hat{x}(k|t)$ and co-variances $P(k|t)$.

*Prediction and filtering.* It is assumed that we know the initial state probability $p(x(0)) = p(x(0)|\mathcal{D}^0)$. The Bayes formula yields recursive formulas for $p(x(k+1)|\mathcal{D}^k)$ (the prediction step) and $p(x(k+1)|\mathcal{D}^{k+1})$ (the filtering step) for $k = 0,1,...$. We have

$$\begin{aligned} \hat{x}(k+1|k) &= A\hat{x}(k|k) + Bu(k) \\ P(k+1|k) &= AP(k|k)A^T + Q \end{aligned} \tag{10}$$

Next, for the filtering step we shall need the Kalman gain, which is obtained as

$$K(k+1) = P(k+1|k)C^T\left(CP(k+1|k)C^T + R\right)^{-1} \tag{11}$$

The filtered state and its covariance matrix are

$$\begin{aligned} \hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + K(k+1)e(k+1|k) \\ P(k+1|k+1) &= P(k+1|k) - \\ & K(k+1)\left(CP(k+1|k)C^T + R\right)K(k+1)^T \end{aligned} \tag{12}$$

where $e(k|k-1) = y(k) - C\hat{x}(k|k-1)$.

*Smoothing.* The starting point is the knowledge of $p(x(k)|\mathcal{D}^k)$. The smoothened past values are obtained by running the following formulas recursively back in time:

$$\begin{aligned} \hat{x}(t|k) &= \hat{x}(k|k) + F(t)\left(\hat{x}(t+1|k) - \hat{x}(t+1|t)\right) \\ P(t|k) &= P(k|k) - F(t)\left(P(t+1|t) - P(t+1|t)\right)F(k)^T \end{aligned} \tag{13}$$

for $t = k-1, k-2, ...$ and $F(t) = P(t|t)A^T P(t+1|t)^{-1}$. In our case, we are particularly interested in smoothing an interval of the past outputs $\hat{Y}(k) = \left[\hat{y}(k-s\,|\,k)^T \quad \ldots \quad \hat{y}(k-1\,|\,k)^T \quad \hat{y}(k\,|\,k)^T\right]^T$, where $\hat{y}(t\,|\,k) := C\hat{x}(t\,|\,k)$. A recursive formula can be obtained from (10)—(13) of the form

$$\hat{Y}(k\,|\,k) = \left[\frac{\left[0 \mid I_{s\cdot p}\right]\hat{Y}(k-1\,|\,k-1)}{\hat{y}(k\,|\,k)}\right] + \left[\frac{\Gamma(k-1)}{0}\right]x(k) \quad (14)$$

$$\Gamma(k) = \left[\frac{\left[0 \mid I_{(s-1)\cdot p}\right]\Gamma(k-1)\Phi(k)}{CP(k\,|\,k)}\right]$$

where

$$\Phi(k) = A^T\left(I - C^T K(k)^T\right) \quad (15)$$
$$x(k) = A^T C^T\left(CP(k\,|\,k-1)C^T - R\right)^{-1} e(k\,|\,k-1)$$

This recursive algorithm is called *Kalman Interval Smoother, KIS*; it can be immediately extended to non-stationary state-space models. Smoothened past outputs can be substituted to (2) and (3) to modify the output history. This leads to reducing sensitivity of these predictions to sensor noise in most cases, but the improvement is not guaranteed; a smooth history does not necessary mean smooth predictions. The history update algorithm, which guarantees predictions equal to those generated directly from the KF state is given in the next section.

## 3. MAIN RESULTS

### *3.1 Recursive formula for output history update*

Here we shall derive an algorithm for updating the past output history so that the ARX model predicts the same future values as the state-space one. It is assumed that these models are equivalent in terms of the input-to-output relation, i.e., their predictions are equal in the noiseless case. Hence, the results presented here are aimed at correcting the ARX-model predictions due to its inadequate noise model. It is assumed that the ARX model is minimal, i.e. there are no pole-zero cancellations in the MISO transfer functions.

The history update can be made on the MISO basis. For this, let $M^{(m)}$ denote the observability matrix of (8) for the *m*-th output. The following lemma then expresses the equivalence.

**Lemma 1.** *Let the ARX model and the state-space one be input-output equivalent. Then, the future predictions are equivalent for any future input if and only if the following relation holds:*

$$Y_p^{(m)}(k) = \tilde{A}_p^{(m)}(1:s_m, 1:s_m)^{-1}\Bigg(\sum_r \tilde{B}_p^{(m,r)}(1:s_m, 1:s_m)\times \quad (16)$$
$$\times U_p^{(r)}(k) - \tilde{A}_f^{(m)}(1:s_m, 1:N+1)M^{(m)}x(k)\Bigg)$$

This lemma can be proven by applying a canonical form and using (3) and (7). It can also be found that if all states are observable from *m*-th output and $A$ has no eigenvalue at 0, then there is a one-to-one mapping between the state and the output history.

Formula (16) is quite simple, especially if we consider the convenient structure of matrices given by (4) and (5). However, it can be computationally more demanding than that for KIS. In this section we are going to obtain a recursive update formula similar to (14) for the output history satisfying (16), where $x(k)$ is replaced by $\hat{x}(k\,|\,k)$. First, we shall assume that matrix $A$ has no eigenvalue at 0, i.e. the system has no pure delays ($n_{dr} = 0$ for all r). This assumption will be relaxed later. Then it can be shown that the past outputs in (16) can be equivalently obtained by running (6) from state $x(k)$ back in time, i.e.,

$$Y_p^{(m)}(k) = F^{(m)}x(k) - \sum_r G^{(m,r)}U_p^{(r)}(k) \quad (17)$$
$$=: F^{(m)}x(k) - Z^{(m)}(k)$$

$$F^{(m)} = \begin{bmatrix} C^{(m)}A^{-n} \\ \vdots \\ C^{(m)}A^{-1} \end{bmatrix}, G^{(m,r)} = \begin{bmatrix} C^{(m)}A^{-1}B^{(r)} & \ldots & C^{(m)}A^{-n}B^{(r)} \\ & \ddots & \vdots \\ & & C^{(m)}A^{-1}B^{(r)} \end{bmatrix} \quad (18)$$

Further, if we exploit the structure of matrix $G^{(m,r)}$, we obtain the following recursive formula

$$Z^{(m)}(k) = \left[\frac{[0 \quad I]Z^{(m)}(k-1)}{0}\right] + \sum_r G^{(m,r)}(:,1)u_r(k-1) \quad (19)$$

Next, using (10)—(12) yields the following identity:

$$F^{(m)}\hat{x}(k\,|\,k) = F^{(m)}\Big(A\hat{x}(k-1\,|\,k-1) + \sum_r B^{(r)}u_r(k-1) +$$
$$K(k)e(k\,|\,k-1)\Big) = \left[\frac{[0 \quad I]F^{(m)}}{C^{(m)}}\right]\hat{x}(k-1\,|\,k-1) + \quad (20)$$
$$F^{(m)}K(k)e(k\,|\,k-1) + \sum_r G^{(m,r)}(:,1)u_r(k-1)$$

Then we arrive to a recursive formula for the history update of the form

$$\tilde{Y}^{(m)}(k\,|\,k) = \left[\frac{[0 \quad I]\tilde{Y}^{(m)}(k-1\,|\,k-1)}{\hat{y}_m(k\,|\,k)}\right] + \left[\frac{F^{(m)}}{0}\right]K(k)e(k\,|\,k-1) \quad (21)$$

where $\tilde{Y}^{(m)}(k\,|\,k) = \left[Y_p^{(m)}(k)^T \mid y_m(k\,|\,k)\right]^T$. This formula holds only for systems without time-delay; in what follows we shall find, under a certain assumption on the process noise model, an update formula that does not have a singularity for singular $A$. In particular, let $\{A_e\}$ be a sequence of matrices with no eigenvalues at the origin so that $\lim_{e\to 0} A_e = A$. Then, the assumption on the process noise covariance matrix is

$$W_j^{(m)} = \lim_{e\to\infty} C^{(m)}A_e^{-j}Q < \infty, \quad \forall m, j \quad (22)$$

A discussion on this assumption is included later in this sub-section. Now, let us denote

$$\bar{\Phi}(k) = \left(I - C^T K(k)^T\right)A^T \quad (23)$$
$$\bar{x}(k) = C^T\left(CP(k\,|\,k-1)C^T + R\right)^{-1}e(k\,|\,k-1)$$

Then we have

$$\left\{C^{(m)}A_e^{-1}K(k)e(k\,|\,k-1)\right\} = \left\{C^{(m)}A_e^{-1}P(k\,|\,k-1)\bar{x}(k)\right\}$$
$$= \left\{C^{(m)}A_e^{-1}\left[AP(k-1\,|\,k-2)\bar{\Phi}(k-1) + Q\right]\bar{x}(k)\right\} \quad (24)$$
$$\to \left[CP(k-1\,|\,k-2)\bar{\Phi}(k-1) + W_1^{(m)}\right]\bar{x}(k)$$

Repeating the process *i*-times yields

$$\left\{C^{(m)}A_e^{-i}K(k)e(k\mid k-1)\right\}\to\left[CP(k-i\mid k-i-1)\bar{\Phi}(k-i)\,\mathbf{L}\right. \quad (25)$$
$$\left.\times\bar{\Phi}(k-1)+\sum_{j=1}^{i-1}W_j^{(m)}\bar{\Phi}(k-j)\times\mathbf{K}\times\bar{\Phi}(k-1)+W_i^{(m)}\right]\bar{x}(k)$$

Let us denote

$$\bar{\Gamma}^{(m)}(k)=\begin{bmatrix}C^{(m)}P(k-s+1\mid k-s)\bar{\Phi}(k-s+1)\,\mathbf{L}\,\bar{\Phi}(k)+\\ +\sum_{j=1}^{s-1}W_j^{(m)}\bar{\Phi}(k-j+1)\,\mathbf{L}\,\bar{\Phi}(k)+W_s^{(m)}\\ \vdots\\ C^{(m)}P(k\mid k-1)\bar{\Phi}(k)+W_1^{(m)}\end{bmatrix} \quad (26)$$

Then, the history update formula can be written as

$$\tilde{Y}^{(m)}(k\mid k)=\begin{bmatrix}[0\quad I]\tilde{Y}^{(m)}(k-1\mid k-1)\\ \hat{y}_m(k\mid k)\end{bmatrix}+\begin{bmatrix}\bar{\Gamma}^{(m)}(k-1)\\ 0\end{bmatrix}\bar{x}(k) \quad (27)$$
$$\bar{\Gamma}^{(m)}(k)=\begin{bmatrix}[0\quad I]\bar{\Gamma}^{(m)}(k-1)\\ C^{(m)}P(k\mid k-1)\end{bmatrix}\bar{\Phi}(k)+\begin{bmatrix}W_s^{(m)}\\ \vdots\\ W_1^{(m)}\end{bmatrix}$$

This algorithm is similar to that for KIS in (14). Indeed, we can derive an explicit formula relating these update terms as

$$\bar{\Gamma}^{(m)}(k)\bar{x}(k)=\Gamma^{(m)}(k)x(k)+\Omega^{(m)}(k)\bar{x}(k) \quad (28)$$

where

$$\Omega^{(m)}(k)=\begin{bmatrix}\sum_{j=1}^{s-1}W_j^{(m)}\bar{\Phi}(k-j+1)\,\mathbf{L}\,\bar{\Phi}(k)+W_s^{(m)}\\ \vdots\\ W_1^{(m)}\bar{\Phi}(k)+W_2^{(m)}\\ W_1^{(m)}\end{bmatrix} \quad (29)$$

Now we shall make a comment on the structural restriction imposed by assumption (22). Assume that the basis of the state-space is chosen so that

$$A=\begin{bmatrix}A_{11} & A_{12}\\ 0 & A_{22}\end{bmatrix},\ A_{22}^{n_D}=0 \quad (30)$$

where the $n_D\times n_D$ sub-matrix $A_{22}$ has all eigenvalues at the origin and $A_{11}$ is invertible. Then, a covariance matrix satisfying the assumption takes the form

$$Q=\begin{bmatrix}Q_{11} & 0\\ 0 & 0\end{bmatrix} \quad (31)$$

On the other hand, if the zero eigenvalues are observable, this also characterizes all process noise covariance matrices, for which the proposed algorithm works. It means that if we decompose the process dynamics to a cascade of systems with finite and infinite impulse responses, only the latter can be disturbed by process noise. This assumption is acceptable in many cases.

### *3.2 Simultaneous history update of the output and estimated disturbance.*

KF can, besides filtering and smoothing the states, estimate unknown disturbances of *a priori* known dynamics. In that case, the process model is augmented by the disturbance model as follows

$$\begin{bmatrix}x(k+1)\\ x_d(k+1)\end{bmatrix}=\begin{bmatrix}A & B_1C_d\\ 0 & A_d\end{bmatrix}\begin{bmatrix}x(k)\\ x_d(k)\end{bmatrix}+\begin{bmatrix}B_2\\ 0\end{bmatrix}u_k(k)+\begin{bmatrix}v(k)\\ v_d(k)\end{bmatrix} \quad (32)$$
$$\begin{bmatrix}y(k)\\ u_u(k)\end{bmatrix}=\begin{bmatrix}C & D_2C_d\\ 0 & C_d\end{bmatrix}\begin{bmatrix}x(k)\\ x_d(k)\end{bmatrix}+\begin{bmatrix}n(k)\\ 0\end{bmatrix}$$

Variables $u_k$ and $u_u$ denote the known input (manipulated variable and/or measured disturbance) and the unknown one (disturbance), respectively. The disturbance dynamics can be, e.g., a discrete-time representation of an integrator for constant disturbances (it also works well for slowly varying inputs), a cascade of two integrators for ramps or a harmonic oscillator. The unknown disturbance, which is an input for the predictor, is an output of KF; hence, it can be smoothened, or its past values can be updated using algorithm (27). However, typical disturbance models have eigenvalues on the unit circle or close to it; hence, the past values can be computed as $\tilde{u}_u(k-i\mid k)=C_dA_d^{-i}\hat{x}_d(k\mid k)$ without any conditioning problem.

If we consider both the output and the estimated disturbance past values as free parameters to obtain Kalman filter-equivalent predictions from the ARX model, we have an under-determined problem whose degrees of freedom can be exploited. For convenience, assume that inputs $u_i(k)$ are known for $i=1,..,j$ and the rest is unknown. Let us denote

$$H_Y=\begin{bmatrix}-\tilde{A}_p^{(1)}(1:s_1,:) & & \\ & \mathbf{O} & \\ & & -\tilde{A}_p^{(p)}(1:s_p,:)\end{bmatrix}, \quad (33)$$
$$H_U=\begin{bmatrix}\tilde{B}_p^{(1,1)}(1:s_1,:) & \mathbf{L} & \tilde{B}_p^{(1,j)}(1:s_1,:)\\ \vdots & & \vdots\\ \tilde{B}_p^{(p,1)}(1:s_p,:) & \mathbf{L} & \tilde{B}_p^{(p,j)}(1:s_p,:)\end{bmatrix}$$
$$\Theta_Y(k)=\begin{bmatrix}Y_p^{(1)}(k\mid k)\\ \vdots\\ Y_p^{(p)}(k\mid k)\end{bmatrix},\Theta_U(k)=\begin{bmatrix}U_p^{(1)}(k\mid k)\\ \vdots\\ U_p^{(j)}(k\mid k)\end{bmatrix} \quad (34)$$
$$T(k)=\begin{bmatrix}\tilde{A}_f^{(1)}(1:s_1,:)\bar{M}\cdot\hat{x}_a(k\mid k)-\sum_{r=j+1}^{l}\tilde{B}_p^{(1,r)}(1:s_1,:)U_p^{(r)}(k)\\ \vdots\\ \tilde{A}_f^{(p)}(1:s_p,:)\bar{M}\cdot\hat{x}_a(k\mid k)-\sum_{r=j+1}^{l}\tilde{B}_p^{(p,r)}(1:s_p,:)U_p^{(r)}(k)\end{bmatrix} \quad (35)$$

where $\hat{x}_a$ is the state of the augmented system and matrix $\bar{M}$ is of the prediction formula (8) (with properly re-arranged rows). The histories $\Theta_Y$ and $\Theta_U$ have to satisfy the following equation

$$[H_Y\quad H_U]\begin{bmatrix}\Theta_Y(k)\\ \Theta_U(k)\end{bmatrix}:=H\Theta(k)=T(k) \quad (36)$$

Now, let us write the recursive update algorithm:

$$\Theta(k+1)=A_\Theta\Theta(k)+B_\Theta\begin{bmatrix}\hat{y}(k\mid k)\\ \hat{d}(k\mid k)\end{bmatrix}+f(k) \quad (37)$$
$$A_\Theta=\operatorname{diag}\left(A_{\Theta 1},.....,A_{\Theta p+j}\right),\quad B_\Theta=\operatorname{diag}\left(B_{\Theta 1},.....,B_{\Theta p+j}\right),$$

where

$$[A_{\Theta i}\mid B_{\Theta i}]=\left[\begin{array}{cc|c}0 & I_{si} & 0\\ 0 & 0 & 1\end{array}\right] \quad (38)$$

Let us denote a particular solution, which was obtained by the algorithm of the previous section as $\Theta^*(k)$. The corresponding update term is

$$f^*(k)=\left[\bar{\Gamma}^{(1)T}(k)\quad \mathbf{L}\quad \bar{\Gamma}^{(p+j)T}(k)\right]^T\bar{x}(k) \quad (39)$$

Any history set satisfying (36) then satisfies the identity $H\Theta(k) = H\Theta^*(k)$; for the update terms there also holds $Hf(k) = Hf^*(k)$. Among those update terms which satisfy this identity, the minimum-norm one is of interest; it is computed as

$$f_{\min}(k) = H^{\dagger} H f^*(k) \tag{40}$$

where $H^{\dagger}$ denotes the Moore-Penrose pseudo-inverse. The reason why the minimum-norm solution is to be considered is this: if the process model has eigenvalues close to the origin, update-term $f^*(k)$ tends to be of high norm; in particular, past values of $\hat{y}_s(k-i\,|\,k)$ may grow sharply in its absolute value with growing *i*. This is because of the matrices $W_i$ defined in (22) are involved in the computation whose norm may grow with *i*. Although $\Theta(k)$ is an internal variable of the controller and any particular choice satisfying (36) should not affect the control performance, numerical difficulties are possible in computing future predictions for high-norm histories.

Finally, we shall present an interesting relation to KIS whose update term can be written as

$$f_s(k) = \left[\Gamma^{(1)T}(k) \quad \cdots \quad \Gamma^{(p+j)T}(k)\right]^T x(k) \tag{41}$$

The history vector of KIS does not generally satisfy (36); however, we present a convergence result, which says that for a sequence of process noise covariance matrices, (36) is satisfied *asymptotically*. It is stated formally in the following lemma.

**Lemma 2.** *Assume the state-space form of the plant augmented by the disturbance model given by (32). Let the process noise covariance matrix be partitioned consistently and given as* $\operatorname{diag}\left(Q \quad Q_d\right)$. *In that case, if* $\|Q\| \to 0$, *then* $Hf_s(k) \to Hf^*(k)$ *uniformly for all k.*

The proof is omitted for space considerations. It is based on relation (28). The result is of practical relevance: KF designed for fast disturbance tracking assumes that the noise affecting the plant states has much smaller covariance than that driving the disturbance model. In that case, substituting the smoothened past outputs and disturbances to the ARX prediction formula yields predictions close to those computed directly from KF.

## 4. AN EXAMPLE

To illustrate the above results, we shall consider an example of the master pressure controller in a combined heat/power plant. The controlled variable is the steam pressure in the header. The manipulated variable is the total fuel supply to boilers feeding steam to the header. Disturbance is the total steam flow from this header (to turbines, and/or to other processes). This flow is not known exactly and therefore it is estimated by KF. Continuous-time transfer functions of the process are given by

$$G_U(s) = \frac{(0.02025s + 0.0025)\,\mathrm{e}^{-15s}}{26570s^6 + 18344s^5 + 4939.7s^4 + 646.7s^3 + 41s^2 + s}$$

$$G_D(s) = \frac{-40.91s^3 - 4.011s^2 - 0.1678s - 0.002148}{26570s^6 + 18344s^5 + 4939.7s^4 + 646.7s^3 + 41s^2 + s}$$

Notice that the process model is integrating and hence disturbance estimation is necessary to obtain realistic predictions. Moreover, the response to manipulated variable is much slower than that of the disturbance, which further increases the need for fast disturbance estimation. The ARX model is obtained by converting the above transfer function to its discrete-time counterpart. Sampling period is 5 seconds. The state-space realization was taken as

$$A = \left[\begin{array}{cccc|ccc} -a_1 & 1 & & & 0 & 0 & b_{11} \\ -a_2 & & \mathbf{O} & & & \vdots & b_{12} \\ \vdots & & & 1 & \vdots & \vdots & \vdots \\ -a_6 & 0 & & 0 & 0 & 0 & b_{16} \\ \hline 0 & \cdots & \cdots & 0 & 0 & 0 & 0 \\ & & \vdots & & 1 & 0 & 0 \\ 0 & \ddots & \ddots & 0 & 0 & 1 & 0 \end{array}\right], B = \left[\begin{array}{c|c} 0 & b_{21} \\ 0 & b_{22} \\ \vdots & \vdots \\ 0 & b_{26} \\ \hline 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{array}\right] \tag{42}$$

$$C = \left[\begin{array}{cccc|ccc} 1 & 0 & \ddots & 0 & 0 & 0 & 0 \end{array}\right], D = \left[\begin{array}{c|c} 0 & 0 \end{array}\right]$$

Disturbance model is assumed as $x_f(k+1) = x_f(k)$, $d(k) = x_f(k)$ and is appended to the above plant model as in (32). The process noise is, for the augmented system, assumed to have its covariance matrix given by $Q = diag\left(q_s I_6 \quad 0_{3\times 3} \quad q_f\right)$, where the zero block corresponds to transport delays in (42). This matrix satisfies the structural condition (22). The particular choice of parameters is $q_s = 1\text{e-}6$ and $q_f = 4$. The output noise variance is $R = 1\text{e-}2$. The noise parameters were not known and used rather as tuning knobs to trade-off between output-noise filtering and disturbance rejection.

The connection of KF and the ARX predictor was tested in the closed loop with a predictive controller that solves the following quadratic program:

$$\min_{\substack{\{u(k+i)\},\, i\in I_Y \\ i\in I_U}} \sum \left(r(k+i) - \hat{y}(k+i\,|\,k)\right)^2 Q_y + + \sum_{i\in I_U} \Delta u(k+i)^2 R_u \tag{43}$$

Here, $r$ is reference to be followed, $\hat{y}(k+i\,|\,k)$ is the estimate of future outputs generated by the ARX model (1); the disturbance is assumed constant over the future horizon. Reference tracking is penalized at specific instants in the future given by the set $I_Y$; further, the future inputs are restricted to $\Delta u(k+i) = 0$ if $i \notin I_U$. There are further constraints to (43) which are not considered here; for the full formulation see (Havlena and Findejs, 2005). The specific parameter choice is as follows: $Q_y = 1\text{e}5$, $R_u = 1\text{e}3$, $I_Y = 8,9,\ldots,50$ and $I_U = 0,2,4,\ldots,40$.

We compare results for three algorithms: first, no history update, i.e., $f(k) \equiv 0$ in (37). Second, the algorithm introduced in Section 3.1, i.e., the history update as in (39). Finally, KIS given by (41).

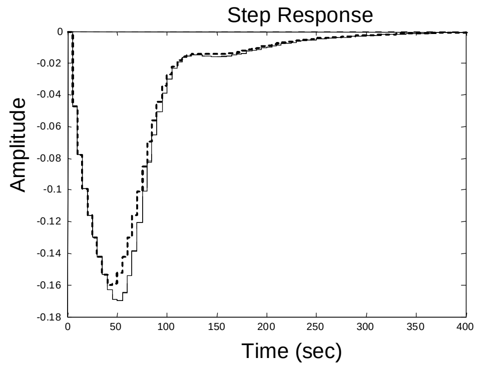


Figure 1 Output step response to disturbance: no history update (dashed), history update (solid)

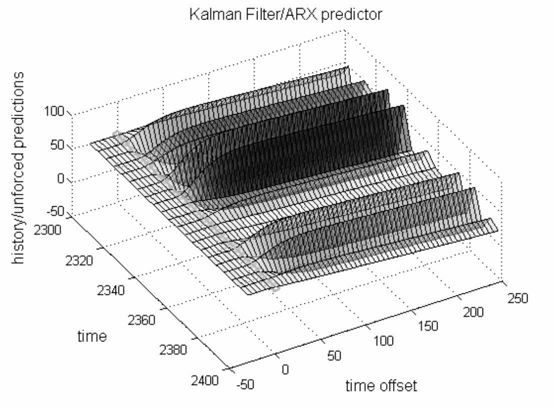


Figure 2 Output history/predictions, no history update

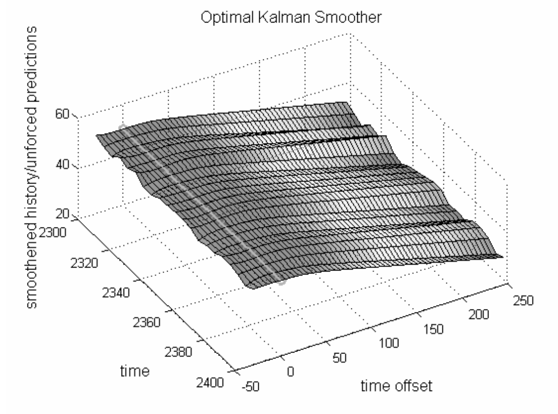


Figure 3 Output history/prediction with KIS

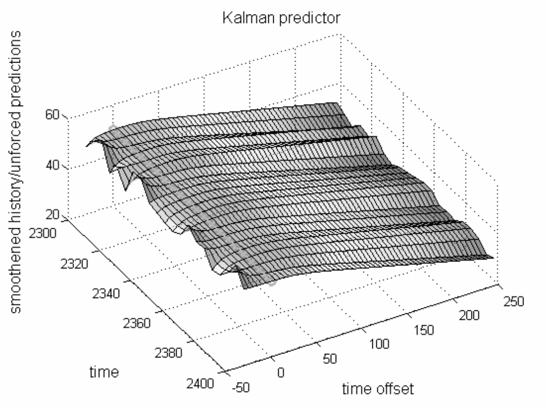


Figure 4 Output history update for Kalman-equivalent predictions

Figure 1 compares responses to the disturbance step. Both algorithms using history updates produce essentially identical responses; for the noise model used in KF, asymptotic properties due to Lemma 2 are fully shown up. Compared to the algorithm without history update, the disturbance attenuation is slower, but negligibly.

An interesting picture we can get, if we plot all predicted and smoothened values of $\hat{y}(k+i\,|\,k)$ for $i=-s,..,0,...,N$ in a 3D plot, where one independent variable is the current time, i.e. $kT_s$, and the other is the time offset $iT_s$. The predictions are assumed unforced, i.e., future manipulated variables being zero. The disturbance is set to a constant (hence, the predicted unforced trajectory is ramped). The output measurement is subject to Gaussian white noise. Figure 2 plots the history and unforced predictions without history update. Measurement noise is hugely amplified at the end-of-horizon predictions; these variations are mapped to the optimal manipulated variable. Note that the ARX predictor still receives the plant outputs pre-filtered by KF, not raw measurements. Figure 3 and Figure 4 show the same variables for the two history update algorithms. It can be seen that the predictions are fairly smooth, and identical in both cases. However, the corrected histories are very different: they are smooth for KIS and of growing magnitude for decreasing time offset in the case of algorithm (27).

## 5. CONCLUSIONS

This paper addresses the problem of modifying data of the ARX predictor to obtain future predictions equal to those of Kalman filter. A recursive algorithm for updating the data was found, under a certain assumption on the noise model. A relation to Kalman internal smoother was found.

## ACKNOWLEDGEMENT

This work was partially supported by the Grant 102/05/2075 of the Czech Science Foundation.

## REFERENCES

Anderson, B.O.D. and Moore, J.B. (1979). *Optimal Filtering*. Prentice Hall, Englewood Cliffs.

Bemporad, A. and Morari, M. (1999). Robust Model Predictive Control: A Survey. In: *Robustness in Identification and Control*, Garulli, A. Tesi, A. Vicino, A. (Eds), Springer, pp. 207-226.

Grewall, M.P. and Andrews, A.P. (2001). *Kalman Filtering. Theory and Practice Using MATLAB*, Wiley, New York.

Havlena, V. and Findejs, J. (2005). Application of model predictive control. *Control Engineering Practice*, Vol. 13, pp. 671-680.