INCREMENTAL STATE-SPACE PREDICTIVE CONTROLLER WITH STATE KALMAN ESTIMATION

K. Belda, J. Böhm

Department of Adaptive Systems
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 182 08 Praha 8 - Libeň
fax : + 420 – 266052068 and e-mails: belda@utia.cas.cz, bohm@utia.cas.cz

Abstract: The paper deals with the incremental modification of generalised state-space Predictive algorithm with state-space Kalman filter estimation. The main points of the derivation of this combination in square-root forms are outlined. The incremental modification can solve the question of steady-state errors. As a benchmark model for simulative tests, simple linear single-input single-output (SISO) system of second order was considered; furthermore, as an example of multi-input multi-output (MIMO) systems, one selected model of redundant parallel robots was used.

Keywords: State-space Predictive algorithm, Steady-state error, Kalman estimation.

1 INTRODUCTION

Predictive controller based on model-based generalized Predictive algorithm (Ordys *et al.* 1993) can offer more powerful control actions than standard PID based controllers and therefore it gains significant and widespread application in industrial process control. Its basic formulation can be adapted, without difficult modifications, directly for multi-input multi-output (MIMO) systems.

Conventional use of PID based controllers for MIMO systems represents taking the systems as a set of single-input single-output units (setSISO) with independent (decentralized) control in their appropriate loops (Sciavicco *et al.* 1996). Internal relations in the controlled MIMO system are assumed as outside disturbances. The use of PID based controllers may generally provide control, but it need not achieve good results, mainly in case of dynamically nonuniform systems. Moreover, it is not applicable for under or over actuated systems, where the knowledge of some model, representing system decoupling, is necessary.

On the other hand, Predictive control design, which is based on some model representation (model-based approach), can assume just by model most of all relations in the controlled system. That applies not only for adequately actuated systems, but also for under and over actuated systems. Through the model, it can design more suitable control actions that closely correspond to actual requirements (desired values). Against independent (decentralized) PID controllers, it represents global, so-called centralized control design (Belda *et al.* 2003).

In spite of incontestable advantages of predictive control design, it can cause, in general point of view, occurrence of steady-state errors (Belda *et al.* 2004). It is happened not only when in the criterion of predictive design, absolute values of control actions are penalized, but also e.g. when unmeasured disturbances or real passive resistances – insensitivities occur. This paper deals with one possible solution based on incremental modification of state-space Predictive algorithm supported with state-space Kalman filter estimation.

In the paper, the main points of derivation of this combination in square-root forms are outlined. As a benchmark model for simulative tests, simple SISO system of second order was considered; furthermore, as an example of MIMO systems, one selected model of redundant parallel robots was used.

The paper is organized as follows. In section two, the construction and modification of state-space model for incremental algorithm is shown. The third section deals partly with the equations of prediction and partly with predictive control design derived in square-root form. The next section, section four, concerns with the question of achievability of new state vector following from incremental model modification. For solving this question, the observer based on Kalman filter is used (Anderson *et al.* 1979). Two observer structures are considered. Design of observer gain is formulated also in square-root form. In concluding sections, sections five and six, several illustrative examples are demonstrated and the use of presented approach is summarized and compared.

2 MATHEMATICAL MODEL OF CONTROLLED SYSTEM AND ITS MODIFICATIONS

Mathematical model, used in control design, is important prior information, which can adjust and optimize control actions for real controlled system. Let us proceed from ordinary used mathematical model – ordinary differential equation generally of n^{th} order

$$y^{(n)} + a_{n-1} y^{(n-1)} + \cdots + a_0 y = b_0 u + \cdots + b_{n-1} u^{(n-1)} \quad (1)$$

The model of the system is represented either by single equation (1) in case of single-input single-output (SISO) systems or set of equations for multi-input multi-output (MIMO) systems. Generally, these cases can be written in state-space form (2)

$$\begin{aligned} \dot{\mathbf{X}}(t) &= \mathbf{A}_C \mathbf{X}(t) + \mathbf{B}_C \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}_C \mathbf{X}(t) \end{aligned} \quad (2)$$

Due to digital realization in practice, the models are discretized to a form

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{A}\,\mathbf{X}(k) + \mathbf{B}\,\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\,\mathbf{X}(k) \end{aligned} \quad (3)$$

The discrete state-space model (3) is suitable form for design based on Predictive control algorithms. In order to constitute (build in) incremental character to predictive algorithms generally both for static and astatic systems, one of possibilities is the following simple modification of state-space model (3)

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{A}\,\mathbf{X}(k) + \mathbf{B}\,\mathbf{u}(k) \\ \mathbf{u}(k+1) &= \mathbf{u}(k) + \Delta\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\,\mathbf{X}(k) \end{aligned} \quad (4)$$

$$\underbrace{\begin{bmatrix} \mathbf{X}(k+1) \\ \mathbf{u}(k+1) \end{bmatrix}}_{\mathbf{X}g(k+1)} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}}_{\mathbf{A}g} \underbrace{\begin{bmatrix} \mathbf{X}(k) \\ \mathbf{u}(k) \end{bmatrix}}_{\mathbf{X}g(k)} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}}_{\mathbf{B}g} \underbrace{\Delta\mathbf{u}(k)}_{\Delta\mathbf{u}(k)}$$

$$\mathbf{y}(k) = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix}}_{\mathbf{C}g} \underbrace{\begin{bmatrix} \mathbf{X}(k) \\ \mathbf{u}(k) \end{bmatrix}}_{\mathbf{X}g(k)} \quad (5)$$

State-space model (5) after condensing has the same form as notation (3)

$$\begin{aligned} \mathbf{X}g(k+1) &= \mathbf{A}g\,\mathbf{X}g(k) + \mathbf{B}g\Delta\,\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}g\,\mathbf{X}g(k) \end{aligned} \quad (6)$$

This modification can ensure consecutive change of control actions in spite of cases of static systems without possibility to measure ideal system state, but only state (output) with immeasurable disturbances. In case of astatic systems, this modification can solve model inaccuracies from reality, as well physical insensitivities (passive resistance, mechanical backlashes etc.). Finally, it removes loss effect at penalization of control action in design criterion.

3 PREDICTIVE DESIGN OF CONTROL

Generalized predictive control belongs with linear quadratic control to multi-step approach (Ordys *et al.* 1993, Philips *et al.* 1995). It combines both feed-forward part and feed-back part. The feed-forward part is represented by prediction via mathematical model describing a controlled system. This part forms the dominant part of control actions. The feed-back, closed from measured outputs, compensates some inaccuracies of the model and certain bounded disturbances.

The design consists in local minimization of the criterion expressed by quadratic cost function. In it, the predictions, given by equations of prediction, are involved. The following subsections outline this approach of model-based design of control.

3.1 Equation of prediction

The prediction is fundamental part of the design. It can define the character of the algorithm (Rossiter 2003, Belda *et al.* 2004). May the following algorithm types are considered

- absolute algorithm
- incremental algorithm.

Absolute algorithm generates directly appropriate values of the actions, i.e. their full (absolute) values. The algorithm arises from the model (3).

On the other hand, incremental algorithm generates only increments of the control actions, which are counted for absolute values applied to the system.

The prediction for both algorithms leads to the repetitive insertion of state-space formula (3) or (6). (Note: For simplicity, in further text, the symbol $(\cdot)$ will represent simultaneously both absolute and incremental state model matrices, and symbol $(.)$ will mark also simultaneously either absolute values of control action or its increment respectively.)

$$\begin{aligned} \widehat{\mathbf{X}}_{(\cdot)(k+1)} &= \mathbf{A}_{(\cdot)}\,\mathbf{X}_{(\cdot)(k)} + \mathbf{B}_{(\cdot)(.)}\mathbf{u}_{(k)} \\ \widehat{y}_{(k+1)} &= \mathbf{C}_{(\cdot)}\mathbf{A}_{(\cdot)}\,\mathbf{X}_{(\cdot)(k)} + \mathbf{C}_{(\cdot)}\mathbf{B}_{(\cdot)(.)}\mathbf{u}_{(k)} \\ &\ \ \vdots \qquad\qquad \vdots \qquad\qquad \ddots \\ \widehat{\mathbf{X}}_{(\cdot)(k+N)} &= \mathbf{A}_{(\cdot)}^{N}\,\mathbf{X}_{(\cdot)(k)} + \cdots + \mathbf{B}_{(\cdot)(.)}\mathbf{u}_{(k+N-1)} \\ \widehat{y}_{(k+N)} &= \mathbf{C}_{(\cdot)}\mathbf{A}_{(\cdot)}^{N}\,\mathbf{X}_{(\cdot)(k)} + \cdots + \mathbf{C}_{(\cdot)}\mathbf{B}_{(\cdot)(.)}\mathbf{u}_{(k+N-1)} \end{aligned} \quad (7)$$

In condensed notation, equations of prediction are

$$\hat{\mathbf{y}} = \mathbf{f} + \mathbf{G}\ {}_{(.)}\mathbf{u}$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{C}_{(\cdot)}\mathbf{A}_{(\cdot)} \\ \vdots \\ \mathbf{C}_{(\cdot)}\mathbf{A}_{(\cdot)}^{N} \end{bmatrix} \mathbf{X}_{(\cdot)(k)},\ \mathbf{G} = \begin{bmatrix} \mathbf{C}_{(\cdot)}\mathbf{B}_{(\cdot)} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{(\cdot)}\mathbf{A}_{(\cdot)}^{N-1}\mathbf{B}_{(\cdot)} & \cdots & \mathbf{C}_{(\cdot)}\mathbf{B}_{(\cdot)} \end{bmatrix} \quad (8)$$

3.2 Minimization of the design criterion

Design criterion – quadratic cost function is defined for certain interval of predictions (several steps to future). It includes the part of control error, in which the model of system is covered (insertion of equations of prediction (8)) and part of control actions, where the input energy (control actions) is weighted. This part redistributes control errors to individual steps of predictions and provides coupling within interval of predictions. Usual form of the criterion for predictive design is written as follows

$$J_k = \sum_{j=No+1}^{N} \left\{ \left(\mathbf{y}_{(k+j)} - \mathbf{w}_{(k+j)} \right)^T \mathbf{Q}_y \left(\mathbf{y}_{(k+j)} - \mathbf{w}_{(k+j)} \right) \right\} +$$

$$+ \sum_{j=1}^{Nu} \left\{ {}_{(.)}\mathbf{u}_{(k+j-1)}{}^T \mathbf{Q}_u\ {}_{(.)}\mathbf{u}_{(k+j-1)} \right\} \quad (9)$$

The criterion is expressed in step k. N is a horizon of prediction, No is a horizon of initial insensitivity and Nu is a control horizon. $\mathbf{Q}_y$ and $\mathbf{Q}_u$ are output and input penalizations and $\mathbf{y}_{(k+j)}$ and ${}_{(.)}\mathbf{u}_{(k+j-1)}$ are output and input (absolute or incremental) values.

The control actions are obtained by minimization of described criterion (9), which can be simply rewritten to the following matrix product

$$J_k = [(\hat{\mathbf{y}} - \mathbf{w})^T, {}_{(.)}\mathbf{u}^T] \begin{bmatrix} \mathbf{Q}\mathbf{y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}\mathbf{u} \end{bmatrix} \begin{bmatrix} \mathbf{Q}\mathbf{y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}\mathbf{u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ {}_{(.)}\mathbf{u} \end{bmatrix} =$$

$$= \qquad \mathbf{J}^T \qquad \times \qquad \mathbf{J} \quad (10)$$

where $\hat{\mathbf{y}}$ is a vector substituted by the equation (8) (time step k+1, · · ·, k+N), $\mathbf{w}$ is a vector of desired values, corresponding to vector $\hat{\mathbf{y}}$ and ${}_{(.)}\mathbf{u}$ is a vector of designed future inputs, again in discrete time instants for the whole horizon (k, · · ·, N - 1).

The product (10), as it is indicated, can be decomposed in so-called square roots of the criterion. From mathematical point of view, the minimization of the square root gives straightforward direction for practical use.

If the square root of the criterion on the right side is selected and expression of prediction (8) is inserted in this square root, then the new criterion is given

$$\mathbf{J} = \begin{bmatrix} \mathbf{Q}\mathbf{y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}\mathbf{u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ {}_{(.)}\mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}\mathbf{y}\ \mathbf{G} \\ \mathbf{Q}\mathbf{u} \end{bmatrix} {}_{(.)}\mathbf{u} - \begin{bmatrix} \mathbf{Q}\mathbf{y}\ (\mathbf{w} - \mathbf{f}) \\ \mathbf{0} \end{bmatrix} \quad (11)$$

$\mathbf{J}$ is a column vector and its Euclidean norm equals a cost of the square root of the criterion (9).

The objective is to search for such ${}_{(.)}\mathbf{u}$, which minimizes the square root (11); which means, that ${}_{(.)}\mathbf{u}$ minimizes the norm |**J**|, and thus as well the criterion (9). In case of square root (11), the minimization leads to a system of algebraic equations with more rows than columns – over-determined system

$$\begin{bmatrix} \mathbf{Q}\mathbf{y}\ \mathbf{G} \\ \mathbf{Q}\mathbf{u} \end{bmatrix} {}_{(.)}\mathbf{u} - \begin{bmatrix} \mathbf{Q}\mathbf{y}\ (\mathbf{w} - \mathbf{f}) \\ \mathbf{0} \end{bmatrix} = \mathbf{0}$$

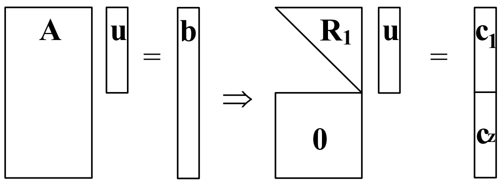
$$\mathbf{A} \qquad \mathbf{u} - \qquad \mathbf{b} \qquad = \mathbf{0} \quad (12)$$

For optimization of the criterion, the orthogonal triangular decomposition (Golub *et al.* 1989; Lawson *et al.* 1974) is used. It reduces excess rows of matrix **A** [(2·N·i)×(N·i)] and elements of vector **b** [2·N·i] (i is a number of inputs of controlled system) into upper triangular matrix **R** and vector **c** according to the following scheme:

$$\mathbf{A}\mathbf{u} = \mathbf{b} \qquad /\mathbf{Q}^T$$

$$\mathbf{Q}^T\mathbf{A}\mathbf{u} = \mathbf{Q}^T\mathbf{b}$$

$$\mathbf{R}\mathbf{u} = \mathbf{c} \quad (13)$$



(14)

Vector $\mathbf{c}_z$ is a lost vector, whose Euclidean norm $|\mathbf{c}_z|$ is equal value of square root $\sqrt{J}$ (i.e. $J = \mathbf{c}_z{}^T\mathbf{c}_z$). To obtain unknown control actions **u**, only upper part of the system (29) is need

$$\mathbf{R}_1\,\mathbf{u} = \mathbf{c}_1$$

$$\mathbf{u} = (\mathbf{R}_1)^T\,\mathbf{c}_1 \quad (15)$$

Since a matrix $\mathbf{R}_1$ is upper triangle, then the control ${}_{(.)}\mathbf{u}$ is given directly by back-run procedure.

The ratio of elements of penalizations **Qu** and **Qy** (usually selected as **Qy** = *diag*(1) and **Qu** = *diag*(λ), $\lambda \in \langle 0, 1 \rangle$) determines magnitude of the redistributed loss in considered horizon of the prediction N.

The horizons Nu and No are selected in view of algorithm. In absolute algorithm, control horizon Nu is usually equal horizon of prediction N; lower value at least by one is required for avoiding block zeros elements in the computation. Initial insensitivity horizon No has similar reason for selection. For absolute algorithm is usually equaled zero; in incremental case should be at least equaled one. It causes, that differences at the beginning of the horizon N are not considered.

Finally, let us note, how to construct real control actions, when the incremental algorithm is selected. After computing vector **u** of control actions for whole horizon, only first control ${}_{(.)}\mathbf{u}(k)$ is used (it is also common for absolute algorithm).

Then, in case of incremental algorithm, where is true $(.)\mathbf{u}(k) \equiv \Delta\mathbf{u}(k)$, the next control actions are given according to second line of equation (4) i.e. in original notation

$$\mathbf{u}(k+1) = \mathbf{u}(k) + \Delta\mathbf{u}(k) \quad (16)$$

4 STATE-SPACE ESTIMATION

In case of the use of Predictive control in state-space formulation, it is necessary to solve the question of achievability of state vector. This question is more significant at incremental algorithm, which requires knowledge of extended state system $\mathbf{X}g$. Just at incremental algorithm, the original state (in spite of its possible availability) is not usable. When the state (extended by values of control actions) will be used, the incremental character is suppressed.

Usual solution of state estimation is state-state observer based on Kalman filtering (Franklin *et al.* 1990). In this section, the Kalman filtering – computation of observer gain will be demonstrated in square root form – optimal way of estimation for real-time use.

Consider a linear, discrete multi-input multi-output system defined by equations (3) or (6) respectively. The Kalman filter is designed such that the estimate is 'the best' in the sense that the expected value of the sum of squares of the error in the estimate is minimal – the best estimate given by the minimum variance estimate, i.e. the conditional expectation $E[\mathbf{X}(k+1\,|\,k)]$ respectively (Billings 1980). The following formulas represent the minimum variance estimate criterion (Anderson *et al.* 1979)

$$E\,(\,\|\mathbf{X}-\hat{\mathbf{x}}\|^2 \,|\, \mathbf{Y}=\mathbf{y}) \le E\,(\,\|\mathbf{X}-\mathbf{z}\|^2 \,|\, \mathbf{Y}=\mathbf{y})$$
$$\hat{\mathbf{x}} = E\,(\mathbf{X}|\mathbf{Y}=\mathbf{y}) = \hat{\mathbf{X}}(\mathbf{y}), \quad \hat{\mathbf{X}} = E\,(\mathbf{X}|\mathbf{Y}), \quad \mathbf{z} = \mathbf{Z}(\mathbf{y})$$

$$\Rightarrow \text{searching for} \qquad \hat{\mathbf{X}} = E\,(\mathbf{X}|\mathbf{Y}) \quad (17)$$

where $\hat{\mathbf{X}}$ is a vector function of random vector variables, $\hat{\mathbf{x}}$ are values of random vector variables, $\mathbf{Z}$ is different arbitrary function determined values of random variable $\mathbf{x} = \mathbf{z}$.

Especially for normal distributed vector (considered here) of random vectors $\mathbf{y}$ and $\mathbf{x}$ with mean values $\overline{\mathbf{y}}$ and $\overline{\mathbf{x}}$, conjugate covariance

$$\Sigma = \begin{bmatrix} \sum yy & \sum yx \\ \sum xy & \sum xx \end{bmatrix} \quad (18)$$

with knowledge of $\mathbf{y}$ (measurement) and conditional probability density

$$p_{\mathbf{X}|\mathbf{Y}}(x,y) = \frac{p_{\mathbf{XY}}(x,y)}{p_{\mathbf{Y}}(y)} = \frac{1}{2\pi^{\frac{N+1}{2}}} \frac{e^{-\frac{1}{2}[x'-\bar{x}' \;\; y'-\bar{y}']\sum^{-1}\begin{bmatrix} x-\bar{x} \\ y-\bar{y} \end{bmatrix}}}{\left|\sum\right|^{1/2}} \frac{2\pi^{1/2}\left|\sum yy\right|^{1/2}}{e^{-\frac{1}{2}(y'-\bar{y}')\sum yy^{-1}(y-\bar{y})}} \quad (19)$$

it can be written (Anderson *et al.* 1979)

$$\hat{\mathbf{X}}(\mathbf{Y}) = E(\mathbf{X}|\mathbf{Y}) = \overline{\overline{\mathbf{X}}} = \overline{\mathbf{X}} + \sum xy \sum yy^{-1}(\mathbf{Y} - \overline{\mathbf{y}}) \quad (20)$$
$$\sum(\mathbf{X}|\mathbf{Y}) = \sum xx - \sum xy \sum yy^{-1} \sum yx \qquad |\mathbf{Y} = \mathbf{y} \quad (21)$$

In previous expressions (21) and (22), the marginal covariances are expressed

$$\begin{aligned} \sum xx &= E((\mathbf{X}_{(k)} - \overline{\mathbf{X}}_{(k)})(\mathbf{X}_{(k)} - \overline{\mathbf{X}}_{(k)})^T) = \sum{}_{k,k-1} \\ \sum yy &= E((\mathbf{y}_{(k)} - \overline{\mathbf{y}}_{(k)})(\mathbf{y}_{(k)} - \overline{\mathbf{y}}_{(k)})^T) = \mathbf{R} + \mathbf{C}\sum{}_{k,k-1}\mathbf{C}^T \\ \sum xy &= E((\mathbf{X}_{(k)} - \overline{\mathbf{X}}_{(k)})(\mathbf{y}_{(k)} - \overline{\mathbf{y}}_{(k)})^T) = \sum{}_{k,k-1}\mathbf{C}^T \\ \sum yx &= (\sum xy)^T = \mathbf{C}^T \sum{}_{k,k-1} \end{aligned} \quad (22)$$

Then conjugate covariance (18) is written that way

$$\Sigma = \begin{bmatrix} \sum yy & \sum yx \\ \sum xy & \sum xx \end{bmatrix} = \begin{bmatrix} \mathbf{R}v + \mathbf{C}\sum_{k,k-1}\mathbf{C}^T & \mathbf{C}\sum_{k,k-1} \\ \sum_{k,k-1}\mathbf{C}^T & \sum_{k,k-1} \end{bmatrix} \quad (23)$$

where $\mathbf{R}v$ is mutual covariance $E(\mathbf{v}\mathbf{v}^T)$. Equation (23) represents only measurement update of the conjugate covariance - current correction.

To obtain time update – prediction of the covariance is necessary to arise from predicted covariance

$$cov\begin{bmatrix} \mathbf{y} \\ \mathbf{X} \end{bmatrix} = E\left(\begin{bmatrix} \mathbf{y}_{k+1} \\ \mathbf{X}_{k+1} \end{bmatrix} \begin{bmatrix} \mathbf{y}^T_{k+1^T} & \mathbf{X}^T_{k+1^T} \end{bmatrix}\right) = \Sigma_{k+1,k} = \cdots$$

$$\cdots = \begin{bmatrix} \overline{\mathbf{R}} + \mathbf{C}(\mathbf{A}\Sigma_{xx}\mathbf{A}^T + \mathbf{G}\overline{\mathbf{Q}}\mathbf{G}^T)\mathbf{C}^T & \mathbf{C}(\mathbf{A}\Sigma_{xx}\mathbf{A}^T + \mathbf{G}\overline{\mathbf{Q}}\mathbf{G}^T) \\ (\mathbf{A}\Sigma_{xx}\mathbf{A}^T + \mathbf{G}\overline{\mathbf{Q}}\mathbf{G}^T)\mathbf{C}^T & \mathbf{A}_D\Sigma_{xx}\mathbf{A}_D{}^T + \mathbf{G}_D\overline{\mathbf{Q}}\mathbf{G}_D{}^T \end{bmatrix} \quad (24)$$

The expression (24) represents both measurement and time update together – so-called combined update. The expression can be decomposed to product of square roots, which are suitable for real-time realization

$$\begin{bmatrix} \mathbf{R}\mathbf{R}^T + \mathbf{C}(\mathbf{A}\mathbf{S}\mathbf{S}^T\mathbf{A}^T + \mathbf{G}\mathbf{Q}\mathbf{Q}^T\mathbf{G}^T)\mathbf{C}^T & \mathbf{C}(\mathbf{A}\mathbf{S}\mathbf{S}^T\mathbf{A}^T + \mathbf{G}\mathbf{Q}\mathbf{Q}^T\mathbf{G}^T) \\ (\mathbf{A}\mathbf{S}\mathbf{S}^T\mathbf{A}^T + \mathbf{G}\mathbf{Q}\mathbf{Q}^T\mathbf{G}^T)\mathbf{C}^T & \mathbf{A}\mathbf{S}\mathbf{S}^T\mathbf{A}^T + \mathbf{G}\mathbf{Q}\mathbf{Q}^T\mathbf{G}^T \end{bmatrix} =$$

$$= \begin{bmatrix} \mathbf{R} & \mathbf{CAS} & \mathbf{CGQ} \\ \mathbf{0} & \mathbf{AS} & \mathbf{GQ} \end{bmatrix} \begin{bmatrix} \mathbf{R}^T & \mathbf{0} \\ \mathbf{S}^T\mathbf{A}^T\mathbf{C}^T & \mathbf{S}^T\mathbf{A}^T \\ \mathbf{Q}^T\mathbf{G}^T\mathbf{C}^T & \mathbf{Q}^T\mathbf{G}^T \end{bmatrix} = \sqrt{\Sigma_{k+1}}(\sqrt{\Sigma_{k+1}})^T \quad (25)$$

where $\mathbf{R}$, $\mathbf{Q}$ and $\mathbf{S}$ are square roots of mutual covariances $E(\mathbf{v}\mathbf{v}^T)$, $E(\mathbf{w}\mathbf{w}^T)$ and $\Sigma xx = \mathbf{S}\mathbf{S}^T$.

If the steady-state value matrix is recursively searched using orthogonal-triangular decomposition (equation (26))

$$(\sqrt{\Sigma_{k+1}})^T = \begin{bmatrix} \mathbf{R}^T & \mathbf{0} \\ \mathbf{S}_k{}^T\mathbf{A}^T\mathbf{C}^T & \mathbf{S}_k{}^T\mathbf{A}^T \\ \mathbf{Q}^T\mathbf{G}^T\mathbf{C}^T & \mathbf{Q}^T\mathbf{G}^T \end{bmatrix} = \begin{bmatrix} (inv)^T & (\mathbf{k})^T \\ \mathbf{0} & \mathbf{S}_{k+1}{}^T \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad (26)$$

then Kalman filter gain can be calculated from this steady-state solution

$$\mathbf{kfg} = ((inv)^{-1}\,\mathbf{k}\,)^T \quad (27)$$

For single-input single-output system, the Kalman gain is a gain vector and for multi-input multi-output system, it is a gain matrix.

In equation (26) on the right side, the future values of square root of $\Sigma xx = \mathbf{S}\mathbf{S}^T$ are appeared. Steady-state solution arises in situation, when $\mathbf{S}_{k+1} = \mathbf{S}_k$.

After this recursive procedure (repetitive converging search for steady-state solution), the equations of estimation (equations of state observer) can be written. They can be formulated in several ways. Firstly, consider single equation of estimation (28) for $\hat{\mathbf{x}}_{(\cdot)k+1,k}$

$$\hat{\mathbf{x}}_{(\cdot)_{k+1,k}} = \mathbf{A}_{(\cdot)}\hat{\mathbf{x}}_{(\cdot)_{k,k-1}} + \mathbf{A}_{(\cdot)}\,\mathbf{kfg}\,(\mathbf{y}_k - \mathbf{C}_{(\cdot)}\hat{\mathbf{x}}_{(\cdot)_{k,k-1}}) + \mathbf{B}_{(\cdot)(.)}\mathbf{u}(k) \quad (28)$$

It follows directly from state-space models (3, 6). For control design, only the delayed estimation (estimation from previous time step) can be used. Thus, the prediction has not current correction, which can further improve the estimation. (Note: It is used also for single-input single-output cases, where the observer gain is selected according to Ackermann formula (Philips *et al.* 1995); The estimation can be possibly used for computation of state-space model matrices, when the controlled system is described by model with time-varying parameters).

The second way, which takes into account current measurement, let us say, way of the 'corrector-predictor' type (Billings 1980), is represented by equation (29) for $\hat{\mathbf{x}}_{(\cdot)k,k}$

$$\hat{\mathbf{x}}_{(\cdot)_{k,k}} = \hat{\mathbf{x}}_{(\cdot)_{k,k-1}} + \mathbf{kfg}\,(\mathbf{y}_k - \mathbf{C}_{(\cdot)}\hat{\mathbf{x}}_{(\cdot)_{k,k-1}}) \quad (29)$$

which represents update according to topical measurement, i.e. a 'corrector'. The corrected state $\hat{\mathbf{x}}_{(\cdot)_{k,k}}$ is then used both for control design and possibly for the systems with time-varying parameters. For next step, another equation is necessary. It should provide estimation – time update and it is given by state-space model (3, 6).

$$\hat{\mathbf{x}}_{(\cdot)_{k+1,k}} = \mathbf{A}_{(\cdot)}\hat{\mathbf{x}}_{(\cdot)_{k,k}} + \mathbf{B}_{(\cdot)(.)}\mathbf{u}(k) \quad (30)$$

The both forms of the observer based on Kalman filter designed gain involved in control circuit with incremental predictive controller are shown in Figures 1 and 2.

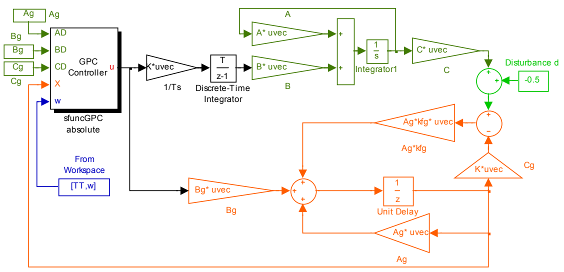


Figure 1. Control circuit with incremental predictive controller and single structured observer.

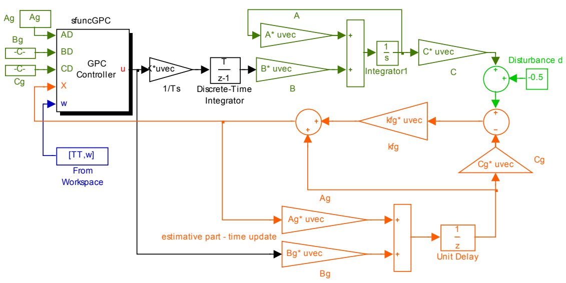


Figure 2. Control circuit with incremental predictive controller and two-part structured observer.

5 ILLUSTRATIVE EXAMPLES

To illustrate presented theory of predictive design and state-space observation based on Kalman filtering, the simple system (31) was selected.

$$y''(t) + 2y'(t) + y(t) \;=\; u(t) \quad (31)$$

It represents a stable system with double root $s = -1$. For comparison, the system (31) was controlled by both algorithms, absolute and incremental. The results are shown in Figures 3 and 4. (Note: In GPC pages (Belda *et al.* 2004) are published generalized presented algorithms. They can serve for next simulative tests and demonstrations).

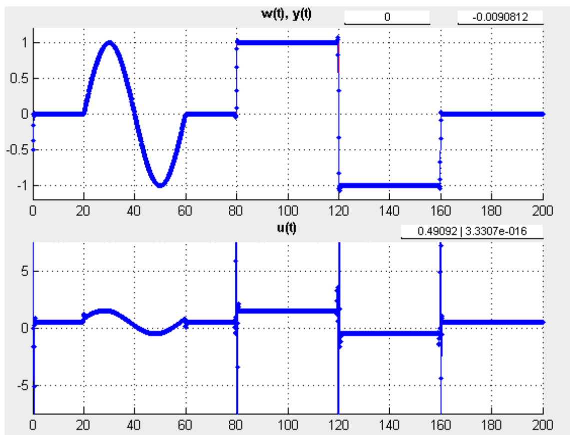


Figure 3. Time histories of control process – absolute predictive algorithm (setting of parameters: $N = 10$, $No = 0$, $Nu = N$, $\lambda = 0.01$).

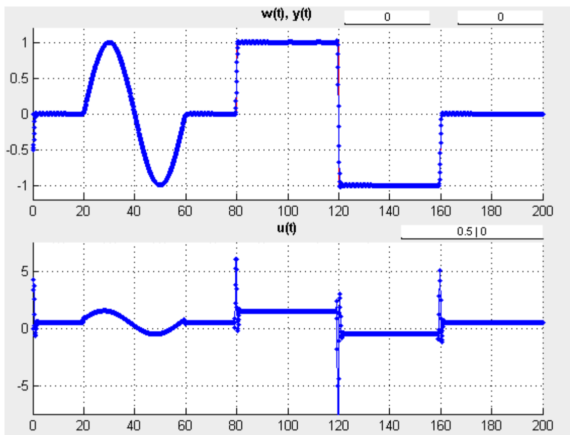


Figure 4. Time histories of control process – incremental predictive algorithm (setting parameters: $N = 10$, $No = 1$, $Nu = N - 1$, $\lambda = 0.01$).

In Figures 3 and 4, the absolute algorithm stopped with nonzero error (response on immeasurable disturbance; in example selected as d = - 0.5) and also nonzero constant value control action. The incremental algorithm removed the steady-state error and stopped on nonzero constant control action equaled the magnitude of immeasurable disturbance of system output.

The following figure shows the use of incremental algorithm for multi-input multi-output system (e.g. robotic structure), which has time-varying model parameters. In case of known mathematical appropriate model of the robot (Belda *et al.* 2001), the estimated state can be used also for its parameter computation.

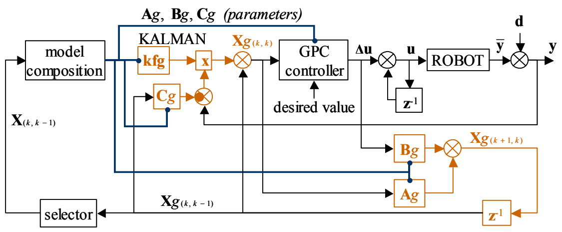


Figure 5. Control circuit with incremental predictive controller and two-part structured observer applied to a system (e.g. robot), which has time-varying parameters.

(Note: In case of time-varying parameters, the actual corrected state cannot be used for model composition, almost the rule of causality should be fulfilled. The procedure can be partly recursively repeated i.e. [: computation of the model » computation of Kalman gain » correction (measurement update) :], and then continue with control design, prediction of the state for next time step (time update)).

6 CONCLUSION

This paper was focused on application of model-based approach represented by discrete generalized predictive control.

On the basis of shown results in section 5 (only representative selection, which can be extended by (Belda *et at.* 2004), the initial questions, how to solve steady-state error and how to provide estimation of extended state vector following from incremental model modification were answered.

Moreover, it can be noted, that the design of observer gain via Kalman filtering in square-root form has several suitable properties:

Square-root Kalman estimation

- keeps physical meaning
- can overcame low regularity
- enables iterative process to spread in time
- can be stopped at certain accuracy.

These properties are desirable in real-time control of high-dynamical systems, where the computation should be achievable during short time period.

In comparison, usually presented approach via searching for steady solution eigenvectors of Hamiltonian matrixes can has problem with local low regularity or it can not be achieved during sampling period due to necessary to perform whole algorithm at every time without possibility to start from some previous posterior value or to stop the procedure at certain reached accuracy.

ACKNOWLEDGMENTS

The authors appreciate the kind support by the Grant Agency of the Czech Republic, grant (101/03/0602, 2003/05), “Redundant drives and measurement for hybrid machine tools” and grant (102/05/0271, 2005/07), “Models of Predictive Control, Algorithms and Implementation”.

7 REFERENCES

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

Belda, K., Böhm, J. and Valášek, M. (2003). State-Space Generalized Predictive Control for Redundant Parallel Robots. *Mechanics Based Design of Structures and Machines*, **Vol. 31**, No. 3, pp. 413-432.

Belda, K., Böhm, J., Píša, P., Valášek, M. (© 2004) GPC pages: http://www.utia.cas.cz/AS/belda/.

Böhm, J. Belda, K. and Valášek, M. (2001). Study of Control of Planar Redundant Parallel Robot. Procedings of the IASTED int. conference MIC, Innsbruck, pp. 694-699.

Billings, S., A. (1980) Introduction to Kalman Filters. *Int. IEE Colloquium on “Kalman Filters and Their Applications”*, pp. 1-7.

Franklin, G. F., Powell, J. D., and Workman, M.L. (1990). Digital Control of Dynamic Systems, Addison-Wesley Public Company, Reading.

Golub, H. G. and Van, Ch. F. L. (1989). *Matrix Computations*, The Johns Hopkins Univ. Press.

Lawson, Ch. and Hanson, R. (1974). *Solving Least Square Problems*, Prentice-Hall, Inc., New York.

Maciejowski, J.M. (2001) *Predictive Control with Constrains*. Prentice Hall, New York.

Ordys, A. and Clarke, D. (1993) A State - Space Description for GPC Controllers. *Int. J. Systems SCI.*, **Vol. 24**, No. 9, pp. 1727 - 1744.

Phillips, L., and Nagle, T. (1995). *Digital Control Systems Analysis and Design*, Prentice Hall, New Jersey.

Rossiter, J. A. (2003). *Model-Based Predictive Control – A Practical Approach*. CRC Press, London.

Sciavicco, L. and Siciliano, B. (1996) *Modeling and Control of Robot Manipulator.* The Mc-Graw Hill Companies, Inc., New York.