Various Utilization of Predictive Control in Parallel Machine Tools

K. Belda

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, e-mail: belda@utia.cas.cz

I. Introduction

Nowadays, the future development of industrial robots – machine tools – requires change and improvement in their control. It means replacement of traditional control (e.g. NC systems, PID/PSD structures), which cannot fully utilize mechanical properties of the machines. These general approaches provide control of the tool drives as separate units only, but not solve the control from view of the whole machine system. This contribution introduces various utilization and possibilities (not only control tasks) of model-based predictive control. It is a representative of up-to-date way that can be use in parallel machines – new developed industrial tools.

II. Problem formulation

The basic control tasks arise partly from the structure partly from requirements of the users (expected properties etc.). In a branch of parallel machines especially over-actuated, the issue is to provide optimal cooperation of all drives interrelated through the movable platform (gripper, chuck). For accomplishment of the issue, some supporting tasks can influence this process. The planning trajectories of movement, calibration, and backlash problems belong among such tasks.

III. Modeling

The model-based approaches use some model as prior information (feed-forward). It enables to predict future behavior of robotic system. Such a way, the input energy can be optimized partly in view of distribution of moving masses, partly by future requirements.

In case of mechanical systems (parallel robots), the initial model is given in a form of system of nonlinear differential equations (1) based on Lagrange's formulation [2]

$$\ddot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \dot{\mathbf{y}}) + \mathbf{g}(\mathbf{y})\mathbf{u} \tag{1}$$

that can be reshaped in state-space formulation

$$\begin{aligned} \dot{\mathbf{X}} &= \mathbf{f}(\mathbf{X}) + \mathbf{g}(\mathbf{X})\mathbf{u}, \quad \mathbf{X} = [\mathbf{y}, \dot{\mathbf{y}}]^T \\ \mathbf{y} &= \mathbf{h}\,\mathbf{X} \end{aligned} \tag{2}$$

Eq. (1) or (2) can be directly used for one-step ahead strategies.

For multi step cases, the nonlinear model should be firstly linearized and then discretized as follows [3]

$$\begin{aligned} \dot{\mathbf{X}} &= \mathbf{A}(\mathbf{X})\,\mathbf{X} + \mathbf{g}(\mathbf{X})\,\mathbf{u} \\ \mathbf{y} &= \quad \mathbf{h} \quad \mathbf{X} \end{aligned} \tag{3}$$

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

The model (4) is fundamental form for control design based on generalized predictive algorithms.

IV. Predictive control

The predictive control [1] is a multi-step control based on N – step prediction of outputs inserted to quadratic cost function (5)

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

$$+ \sum_{j=1}^{Nu} \left\{ u(k+j-1)^T Q_u \; u(k+j-1) \right\} \tag{5}$$

which is minimized. The criterion has several types of horizon (N, Nu, No) enabling both to follow desired trajectories and to plan optimal trajectories in view of the cost function.

1. Predictions of future outputs

The most usual, absolute predictive algorithm arises directly from state-space model (4)

$$\begin{array}{lclclclclcl} \hat{\mathbf{X}}\,(k+1) & = & \mathbf{A} & \mathbf{X}(k) & + & & \mathbf{Bu}(k) & & & & \\ \hat{\mathbf{y}}\,(k+1) & = & \mathbf{C}\,\mathbf{A} & \mathbf{X}(k) & + & \mathbf{C} & \mathbf{Bu}(k) & & & & \\ \hat{\mathbf{X}}\,(k+2) & = & & \mathbf{X}(k) & + & \mathbf{A} & \mathbf{Bu}(k) & & & & \\ \hat{\mathbf{y}}\,(k+2) & = & \mathbf{C}\,\mathbf{A}^2 & \mathbf{X}(k) & + & \mathbf{C}\,\mathbf{A} & \mathbf{Bu}(k) & & & & \\ \vdots & & & \vdots & & & \vdots & \ddots & & & \\ \hat{\mathbf{X}}(k+N) & = & \mathbf{A}^N & \mathbf{X}(k) & + & \mathbf{A}^{N-1} & \mathbf{Bu}(k) & + & \cdots & + & \mathbf{Bu}(k+N-1) \\ \hat{\mathbf{y}}\,(k+N) & = & \mathbf{C}\,\mathbf{A}^N & \mathbf{X}(k) & + & \mathbf{C}\,\mathbf{A}^{N-1} & \mathbf{Bu}(k) & + & \cdots & + & \mathbf{C}\,\mathbf{Bu}(k+N-1) \end{array} \tag{6}$$

written in matrix notation

$$\hat{\mathbf{y}} = \mathbf{f} + \mathbf{G}\mathbf{u} \,\Bigg|_{\substack{\hat{\mathbf{y}} = [\hat{\mathbf{y}}(k+1),\, \hat{\mathbf{y}}(k+2),\, \cdots,\, \hat{\mathbf{y}}(k+N)]^T \\ \mathbf{u} = [\mathbf{u}(k),\, \mathbf{u}(k+1),\, \cdots,\, \mathbf{u}(k+N-1)]^T}} ,$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^N \end{bmatrix} \mathbf{X}(k), \quad \mathbf{G} = \begin{bmatrix} \mathbf{CB} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix} \tag{7}$$

Another, unconventional algorithm is based on increments with simulative substitution. It considers modified version (4):

$$
\begin{aligned}
\mathbf{X}(k+1) &= \underbrace{\mathbf{A}(\mathbf{X}(k))\,\mathbf{X}(k)+\mathbf{B}(\mathbf{X}(k))\,\mathbf{u}(k)_{sim}} + \mathbf{B}\,\Delta\mathbf{u}(k) \\
\mathbf{X}(k+1) &= \qquad\qquad \mathbf{X}(k+1)_{sim} \qquad\qquad + \mathbf{B}\,\Delta\mathbf{u}(k), \quad \mathbf{u}(k)=\mathbf{u}(k)_{sim}+\Delta\mathbf{u}(k) \\
\mathbf{y}(k) &= \mathbf{C}\;\mathbf{X}(k)
\end{aligned} \tag{8}
$$

where $\mathbf{X}(k+1)_{sim}$, $\mathbf{u}(k)_{sim}$ (etc. $\mathbf{X}(k+i)_{sim}$, $\mathbf{u}(k+i-1)_{sim}$; $i = 1,\cdots, N$) is a state and control action given from pre-simulation in considered horizon of some comparable discrete control. The derivation is similar to previous case – i.e. suitable repetitive substitution for predicted future states and outputs respectively.

$$
\begin{bmatrix} \hat{\mathbf{y}}(k+1) \\ \hat{\mathbf{y}}(k+2) \\ \vdots \\ \hat{\mathbf{y}}(k+N) \end{bmatrix} = \begin{bmatrix} \mathbf{C}\;\mathbf{X}(k+1)_{sim} \\ \mathbf{C}\;\mathbf{X}(k+2)_{sim} \\ \\ \mathbf{C}\;\mathbf{X}(k+N)_{sim} \end{bmatrix} +
$$

$$
+\begin{bmatrix} \mathbf{CB}(\mathbf{X}(k)) & & \cdots & \mathbf{0} \\ \mathbf{C}(\mathbf{A}(\mathbf{X}(k+1)_{sim})+\mathbf{I})\mathbf{B}(\mathbf{X}(k)) & \mathbf{CB}(\mathbf{X}(k+1)_{sim}) & & \vdots \\ \vdots & & \ddots & \\ \mathbf{C}(\mathbf{A}(\mathbf{X}(k+N-1)_{sim})+\cdots+\mathbf{A}(\mathbf{X}(k+1)_{sim})+\mathbf{I})\mathbf{B}(\mathbf{X}(k)) & & \cdots & \mathbf{CB}(\mathbf{X}(k+N-1)_{sim}) \end{bmatrix}
$$

$$
\times \begin{bmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \\ \Delta\mathbf{u}(k+N-1) \end{bmatrix} \tag{9}
$$

and corresponding matrix notation is expressed as

$$
\begin{aligned}
\hat{\mathbf{y}} &= \hat{\mathbf{y}}_{sim} + \\
&+ \mathbf{G} \\
&\times \Delta\mathbf{u}
\end{aligned}
\qquad
\left|
\begin{aligned}
\hat{\mathbf{y}} &= [\hat{\mathbf{y}}(k+1), \hat{\mathbf{y}}(k+2), \cdots, \hat{\mathbf{y}}(k+N)]^T \\
\Delta\mathbf{u} &= [\Delta\mathbf{u}(k), \Delta\mathbf{u}(k+1), \cdots, \Delta\mathbf{u}(k+N-1)]^T
\end{aligned}
\right. \tag{10}
$$

All mentioned predictive algorithms of $\hat{\mathbf{y}}$ can be generalized via one expression

$$
\hat{\mathbf{y}} = \mathbf{f} + \mathbf{G}\mathbf{u} \tag{11}
$$

in which the vector f and matrix **G** are changed according to type of prediction.

For next derivation - control design, the differences in (11) are not significant and they do not change the derivation. The implication of described prediction consists in the use. The former, absolute form is the simplest one. And later, the more difficult, incremental algorithm with simulative substitution uses prior predicted states from numerical simulation of comparable local discrete control (i.e. control + numerical pre-simulation within considered horizon *N*).

Thus, the new prediction (11) within horizon N is not based only on one model given by $\mathbf{X}(k)$, but it is based on a set of models changing with each step of prediction $(1, \cdots, N)$. The prediction (11) respects by such way the nonlinearities of controlled system.

2. Computation of control actions

The control actions are obtained by minimization of quadratic criterion (5). It can be simply rewritten to the following matrix product

$$J_k = [(\hat{\mathbf{y}}-\mathbf{w})^T, \mathbf{u}^T] \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix} \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}}-\mathbf{w} \\ \mathbf{u} \end{bmatrix} = \mathbf{J}^T \mathbf{J} \tag{12}$$

where $\hat{\mathbf{y}}$ is a vector composed according to (11) (time step $k+1, \cdots, k+N$), $\mathbf{w}$ is a vector of desired values, corresponding to vector $\hat{\mathbf{y}}$ and u is a vector of designed future inputs, again in discrete time instants for the whole horizon $(k, \cdots, N-1)$.

The product (12) is more suitable form that can be decomposed in two parts so-called square roots of the criterion. From mathematical point of view the minimization of square root is more suitable [7]. Let us select square root on the right side (without transpositions) and use the expression for prediction (11)

$$\mathbf{J} = \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}}-\mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{Q_y G} \\ \mathbf{Q_u} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{Q_y}(\mathbf{w}-\mathbf{f}) \\ \mathbf{0} \end{bmatrix} \tag{13}$$

$\mathbf{J}$ is a column vector, of which Euclidean norm is a cost of the square root of the criterion.

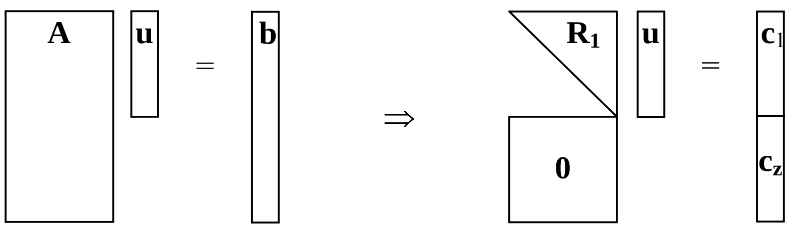
We search for such $\mathbf{u}$, which minimizes the square root (13) i.e. the control $\mathbf{u}$ should minimize the norm |J| of criterion. It is fulfilled, if the right side of expression (13) (system with more rows than columns, over-determined system) is annulled.

$$\mathbf{J} = \begin{bmatrix} \mathbf{Q_y G} \\ \mathbf{Q_u} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{Q_y}(\mathbf{w}-\mathbf{f}) \\ \mathbf{0} \end{bmatrix} \overset{!}{=} \mathbf{0}$$

$$\mathbf{A} \quad \mathbf{u} - \quad \mathbf{b} \quad = \mathbf{0} \tag{14}$$

In computation, the orthogonal triangular decomposition [4] is used. It reduces excess rows of matrix $\mathbf{A}$ $[(2 \cdot N \cdot i) \times (N \cdot i)]$ and elements of vector $\mathbf{b}$ $[2 \cdot N \cdot i]$ (i is a number of DOF) into upper triangular matrix $\mathbf{R}$ and vector $\mathbf{c}$ according to the following scheme:

$$\begin{aligned} \mathbf{A u} &= \mathbf{b} \qquad / \mathbf{Q}^T \quad \text{(multiplication by orthogonal matrix)} \\ \mathbf{Q}^T \mathbf{A u} &= \mathbf{Q}^T \mathbf{b} \\ \mathbf{R u} &= \mathbf{c} \end{aligned} \tag{15}$$



(16)

Vector $\mathbf{c_z}$ is a lost vector, whose Euclidean norm $|\mathbf{c_z}|$ is equal to value of square root $\sqrt{J}$ (i.e. $J = \mathbf{c_z}^T\mathbf{c_z}$). To obtain unknown control **u**, we need only upper part of the system (16)

$$\begin{aligned} \mathbf{R}_1 \mathbf{u} &= \mathbf{c}_1 \\ \mathbf{u} &= (\mathbf{R}_1)^T \mathbf{c}_1 \end{aligned} \tag{17}$$

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

3. Planning of quadratically-optimal trajectories

As one interesting possibility, the predictive control offers, due to its several horizons (*N*, *No*, *Nu*), planning of trajectories by recording of outputs from simulation. The task is defined as follows: let us have two points – start and end, and at the same time, a path (trajectory) is not conditioned, only end-point must be achieved.

In such case, we can use predictive control with specific setting of the output horizons *N* and *No*. If we set, that the horizon $N = Nmax$ and $No = N - k$, where *k* is order of the controlled system, then the quadratic criterion will consider only last *k* differences among predicted end-point and its reference value. Thus, the matrix **G** and corresponding differences (**w** - **f**) in the criterion (12) or (13), are reduced only on their last *k* rows and elements respectively

$$\left[\begin{array}{cc:c} \multicolumn{2}{c:}{\overline{\mathbf{G}}} & \mathbf{w}-\mathbf{f} \\ \hdashline \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \cdot & & \\ \mathbf{1} & & \vdots \\ & \cdot & \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \end{array}\right] = \left[\begin{array}{ccccc:c} \mathbf{CA}^{N-k}\mathbf{B} & \cdots & \mathbf{CB} & \cdots & \mathbf{0} & (\mathbf{w}-\mathbf{f})_{N-k} \\ \vdots & & & \ddots & \vdots & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & \cdots & \mathbf{CAB} & \mathbf{CB} & (\mathbf{w}-\mathbf{f})_{N} \\ \hdashline \mathbf{1} & \cdots & & \cdots & \mathbf{0} & \mathbf{0} \\ & \cdot & & & & \\ \vdots & & \mathbf{1} & & \vdots & \vdots \\ & & & \cdot & & \\ \mathbf{0} & \cdots & & \cdots & \mathbf{1} & \mathbf{0} \end{array}\right] \tag{18}$$

Lower unit matrix in (18) corresponds to dimension of input penalization.

Such form, specifically last k rows of matrix **G** and corresponding differences (**w** - **f**), causes quadratic distribution of energy to individual inputs (control actions) within whole horizon *Nmax*.

If indicated procedure would be applied, then the control process has no information, in which step should stop. Difference of horizons *N* and *No* is still the same. Information on stopping the control process is given by horizon *No*.

Described sequence represents specific dead-bead control spread within time. Thus, it is not necessary to achieve end point during minimal number of steps (= order of system) in control process, but on the other hand (from reasons of feasibility by drives) it is better to distribute the input energy uniformly without rapid turns in some wider horizon. Its length should arise from technological requirements.

The sequence can be used only once under condition, that the system is linear and horizon *No* is a little bit lower than horizon *N*; i.e. value *No* gives the length of horizon, on which the system should stop after previous $(Nmax - No)$ steps.

In case of nonlinear systems, the sequence has to be repeated with progressively shortened horizon N

$$N := Nmax, Nmax - 1, \cdots, Nmin + 1, Nmin \qquad (19)$$

where the value $Nmin$ is suitable selected, not exceed number c. 20 ($k < Nmin < 20$). The higher numbers do not improve the process. The repetition provides the changes of model during planning of the trajectory according to real state of the controlled system i.e. it respects nonlinearity by changing of models in compliance with real positions and velocities of the robot.

V. Concluding notes

Presented predictive algorithms allow and solve:

- redundancy of drives, where the actions are not uniquely defined
- possible mutual drive fighting
- utilization of model in nonlinear form
- quadratically-optimal trajectory planning.

Figure 1: Example of parallel structure (laboratory model)

Acknowledgement

This research has been supported by IGS CTU (040 6413, 2004) “Implementation of control in redundant parallel robotic structures” and GA ČR (101/03/0620, 2003-2005) “Redundant drives and measurement for hybrid machine tools”.

References

[1] A. Ordys and D. Clarke, “A state-Space Description for GPC Controllers”, *Int. J. Systems*, 24(9): 1727-1744, 1993.

[2] V. Stejskal and M. Valášek, *Kinematics and Dynamics of Machinery,* New York: Marcel Dekker, Inc., 1996, 494 pp.

[3] K. Belda, J. Böhm and M. Valášek, “State-Space Generalized Predictive Control for Redundant Parallel Robots”, *Mechanics Based Design of Structures and Machines*, 31(3): 413-432, 2003.

[4] K. Belda, *Control of Redundant Parallel Structures of Robotic Systems*, Ph.D. thesis, Czech Technical University in Prague, September 2002.