PREDICTIVE CONTROL FOR MODERN INDUSTRIAL ROBOTS
Algorithms and their applications

Květoslav Belda, Josef Böhm
Department of Adaptive Systems
Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 182 08 Prague 8 – Libeň, Czech Republic

Pavel Píša
Department of Control Engineering
Faculty of Electrical Engineering, Czech Technical University in Prague
Karlovo nám. 13, 121 35 Prague 2, Czech Republic

Keywords: Predictive control, Redundant manipulators, Robotics, Steady-state error problem.

Abstract: Industrial robots comprise substantial parts of machine tools and manipulators in production lines. Their present development stagnates in their control. Traditional approaches, e.g. NC (numerical control) systems combined with PID/PSD structures, provide control of the tool drives as separate units only, but not solve the control from view of the whole machine system. On the other hand, in control theory, there are a lot of approaches, in which the information on tool dynamics and kinematic relations can be involved. The main contribution of this paper is to introduce various utilization and modifications (not only control tasks) of one such approach – model-base predictive control. The control is being developed for modern industrial robots based on parallel configurations. The modifications of predictive algorithm are substantiated by real laboratory experiments. The paper concerns with basic control design and its possibilities to remove positional steady-state error. Quadraticaly-optimal trajectory planning is outlined in it.

1 INTRODUCTION

Industrial robots comprise substantial parts of machine tools and manipulators in production lines. Their present development stagnates in their control, which should ensure not only high accuracy, but also economical and safe operation.

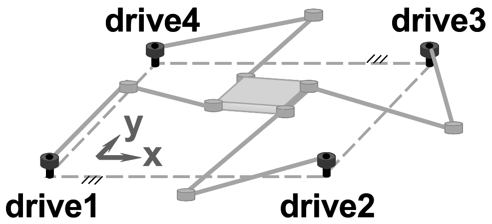
Traditional approaches, e.g. Numerical Control systems (NC systems) combined with PID/PSD structures, provide control of the robots only from view of their drives considered as separate units. It means that whole robotic system is taken into account as a set of drives and their relations. These relations are given by mechanical constrains arising from real robot structure – real mechanism. In those approaches, the relations are considered only as disturbances acting to individual drives. This concept yields no possibilities for further increase of operational accuracy and load capacity.

Modern control approaches can involve the most of properties of whole robotic system through its mathematical model. It is obtained by virtue of mathematical-physical analysis or some numerical identification method. Such control approaches are generally called model-based approaches. They can design, just by use of the mathematical model, corresponding and energetically reasonable control actions.

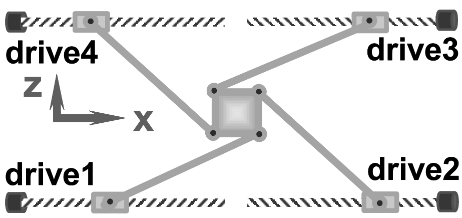
One of modern model-based approaches is multi-step Predictive control (Ordys and Clarke 1993). It is applicable in new developed industrial robots (e.g. Neugebauer, ed., 2002), which are based on redundantly actuated parallel structures. They represent multi-input multi-output systems and their redundant actuation solving the problem of workspace singularities (Belda et al. 2003) deter mines different number of their inputs and outputs.

In use of parallel robots, model-based control can be fully utilized, for more complicacy of the robot structure. Control based on model can better distribute the input energy in the structure. Example of two parallel structures is shown in Figure 1.

Belda K., Böhm J. and Píša P. (2005).
PREDICTIVE CONTROL FOR MODERN INDUSTRIAL ROBOTS - Algorithms and their applications.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics - Robotics and Automation*, pages 3-10
DOI: 10.5220/0001155800030010



'Crosshead'



'Sliding Star'

Figure 1: Planar parallel robots: horizontal and vertical

Although the parallelism is appeared in robotics in the sixties - Stewart platform in 1965 (Tsai, 1999), their wider development started in the nineties (Neugebauer, ed., 2002). The parallel robots are promising way, how to significantly improve accuracy, speed and stiffness of machine tools. They can be simply understood as movable truss constructions or as movable work platforms supported by a set of parallel arms (Tsai, 1999).

The main contribution of this paper is to demonstrate various utilization and modifications (not only control tasks) of predictive control; if it can achieve acceptable dynamic control errors and solve steady-state error problem. The theoretical results implemented in algorithms are substantiated by real laboratory experiments with redundant parallel structure 'Sliding Star' (Figure 1, 3).

2 MODEL-BASED APPROACH

The model-based approaches use the model as prior information (feed-forward). It enables to predict future behavior of a controlled system. Considering future requirements and behavior, the input energy can be optimized (Ordys and Clarke, 1993).

The models can be expressed in different forms. In case of multi-input multi-output structures (MIMO systems), as robots are, the model is useful expressed by state-space formulation. State-space model more clearly expresses the relations among inputs and outputs and their coupling.

2.1 Composition of the Robot Model

Generally, the robot is a multibody system. Its model is represented by pure equations of motion. They are composed mostly from Lagrange's equations (e.g. Stejskal and Valášek, 1996). Then the mathematical model described the real system is given by a set of differential equations (1)

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

where input vector $\mathbf{u}$ can represent only forces, caused by torques on drives in case of horizontal configuration

$$\mathbf{u} = \mathbf{F}_{\mathcal{T}} \tag{2}$$

or these forces enlarged by gravitational forces in case of vertical configuration

$$\mathbf{u} = -\mathbf{F}_{\mathcal{G}} + \mathbf{F}_{\mathcal{T}} \tag{3}$$

The equations (2) and (3) serve for final determination of real control actions – force effects required from drives. Furthermore, this arrangement provides the equality $\mathbf{f}(\mathbf{y}, \dot{\mathbf{y}}) = \mathbf{0}$ for arbitrary $\mathbf{y}$ from range of definition and zero time derivatives $\dot{\mathbf{y}} = \mathbf{0}$ in spite of presence of gravitational forces, which are added to inputs (equation (3)).

The function $\mathbf{f}(\mathbf{y}, \dot{\mathbf{y}})$ produces that the equation (1) is nonlinear for state $\mathbf{X} = [\mathbf{y}, \dot{\mathbf{y}}]^T = [\mathbf{x}_1, \mathbf{x}_2]^T$. One of ways to cope with nonlinearity is to use some kind of linearization. In the following subsection, one linearizing technique is introduced.

2.2 Exact Linearization

The subsection 2.2 deals with the linearization based on differences. Against standard linearization using partial derivatives, the resultant form is usable not only in the working point, but also in its wider neighbourhood.

The nonlinearity in the equation (1) can be linearized as follows (Valášek and Steinbauer, 1999)

$$\mathbf{f}(\mathbf{y}, \dot{\mathbf{y}}) = \mathbf{a}_1(\mathbf{y}, \dot{\mathbf{y}})\,\mathbf{y} + \mathbf{a}_2(\mathbf{y}, \dot{\mathbf{y}})\,\dot{\mathbf{y}} \tag{4}$$

and transformed in state-space form

$$\mathbf{f}(\mathbf{X}) = \begin{bmatrix} \dot{\mathbf{y}} \\ \mathbf{f}(\mathbf{y}, \dot{\mathbf{y}}) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{a}_1 & \mathbf{a}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \tag{5}$$

Provided that the nonlinear function $\mathbf{f}(\mathbf{y}, \dot{\mathbf{y}})$ and point $\mathbf{X} = [\mathbf{y}, \dot{\mathbf{y}}]^T = [\mathbf{x}_1, \mathbf{x}_2]^T$ are given; $\mathbf{X}$ belongs to the range of definition of the function; zero elements in $\mathbf{X}$ are substituted by suitable nonzero number $\kappa \to 0$ to prevent zero division. Furthermore, two types of state variables are assumed: generally outputs $\mathbf{x}_1 = \mathbf{y}$ and their time derivatives $\mathbf{x}_2 = \dot{\mathbf{y}}$; i.e.

$$\mathbf{X} = [\mathbf{y}, \dot{\mathbf{y}}]^T = [\mathbf{x}_1, \mathbf{x}_2]^T = [[x_{11}, x_{12}, \cdots], [x_{21}, x_{22}, \cdots]]^T \tag{6}$$

Finally, the assumption from previous section has to be also fulfilled for arbitrary $\mathbf{y}$ and zero $\dot{\mathbf{y}}$

$$\mathbf{f}(\mathbf{y},\dot{\mathbf{y}}) = \mathbf{f}(\mathbf{X}) = \mathbf{0} \Big|_{\mathbf{X}=\mathbf{X}_r=[\mathbf{x}_{r1} \text{ arbitrary},\ \mathbf{x}_{r2}=\mathbf{0}]} \quad (7)$$

Then the decomposition indicated by equation (4) can be reached. Its algorithm starts from second state variables $\mathbf{x}_2$ (i.e. according to the equation (6): order is

$$\{x_{21}, x_{22}, \cdots, x_{11}, x_{12}, \cdots\} \text{ i.e. } \{21, 22, \cdots, 11, 12, \cdots\}) \quad (8)$$

The order is given by amount of the information included in the function $\mathbf{f}(\mathbf{X})$ and assumption (7). The indicated order of selection will considerably simplify decomposition, as it will be shown later.

In view of previous assumptions, the exact linearization-decomposition is expressed as follows:

$$\mathbf{f}(\mathbf{X}) = \frac{\Delta\mathbf{f}(\circ)}{.\Delta x_{11}}\Delta x_{11} + \frac{\Delta\mathbf{f}(\circ)}{.\Delta x_{12}}\Delta x_{12} + \cdots +$$
$$+ \frac{\Delta\mathbf{f}(\circ)}{.(x_{21}-0)}(x_{21}-0) + \frac{\Delta\mathbf{f}(\circ)}{.(x_{22}-0)}(x_{22}-0) + \cdots \quad (9)$$

(Note: The dots before variables in denominators mark division 'element by element'; division of all elements of differences by scalar Δxij.)

In detail, the equation (9) is written that way

$$\mathbf{f}(\mathbf{X}) =$$
$$= \frac{\mathbf{f}([x_{11}, x_{12}, x_{13}, 0,0,0]^T) - \mathbf{f}([x_{r11}, x_{12}, x_{13}, 0,0,0]^T)}{.(x_{11}-x_{r11})}(x_{11}-x_{r11}) + \cdots$$
$$+ \frac{\mathbf{f}([x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}]^T) - \mathbf{f}([x_{11}, x_{12}, x_{13}, 0, x_{22}, x_{23}]^T)}{.x_{21}} x_{21} + \cdots \quad (10)$$

The individual fractions of the equation (10) are columns of the coefficients of the matrices $\mathbf{a}_1(\mathbf{y},\dot{\mathbf{y}})$ and $\mathbf{a}_2(\mathbf{y},\dot{\mathbf{y}})$ with following internal structures

$$\mathbf{a}_1(\mathbf{y},\dot{\mathbf{y}}) = \mathbf{0}, \quad \mathbf{a}_2 = \begin{bmatrix} \mathbf{a}2_{11} & \mathbf{a}2_{12} & \cdots \\ \mathbf{a}2_{21} & \mathbf{a}2_{22} & \\ \vdots & & \ddots \end{bmatrix} \quad (11)$$

The first column group (matrix $\mathbf{a}_1(\mathbf{y},\dot{\mathbf{y}})$) contains only zeros due to differences being also zeros - the vector function equals zeros for zero time derivatives; see equation (6) – e.g. numerator of the first column of $\mathbf{a}_1(\mathbf{y},\dot{\mathbf{y}})$ is

$$\mathbf{f}([x_{11}, x_{12}, x_{13}, 0,0,0]^T) - \mathbf{f}([x_{r11}, x_{12}, x_{13}, 0,0,0]^T) = \mathbf{0} \quad (12)$$

2.3 Discrete State-Space Formulation

Let the linear (or linearized) differential equation or the system of differential equations with separate the highest derivation on left side are assumed

$$\ddot{\mathbf{y}} = \mathbf{f}(\mathbf{y},\dot{\mathbf{y}}) = \mathbf{a_1}(\mathbf{y},\dot{\mathbf{y}})\,\mathbf{y} + \mathbf{a_2}(\mathbf{y},\dot{\mathbf{y}})\,\dot{\mathbf{y}} + \mathbf{g}(\mathbf{y})\mathbf{u} \quad (13)$$

Then, its continuous state-space form is written as

$$\dot{\mathbf{X}} = \mathbf{A}_C\,\mathbf{X} + \mathbf{B}_C\,\mathbf{u} \quad \Bigg| \quad \mathbf{X} = \begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix};\ \mathbf{A}_C = \begin{bmatrix}\mathbf{0} & \mathbf{1}\\ \mathbf{a}_1 & \mathbf{a}_2\end{bmatrix};\ \mathbf{B}_C = \begin{bmatrix}\mathbf{0}\\ \mathbf{g}(\mathbf{x}_1)\end{bmatrix}$$
$$\mathbf{y} = \mathbf{C}_C\,\mathbf{X} \quad \Big| \quad \mathbf{C}_C = [\mathbf{c}_1\ \ \mathbf{c}_2] \quad (14)$$

For real-time control the system (14) has to be discretized, because continuous realization is not feasible. The reason is that the model of the robot needs certain time for its own composition and real control systems are usually realized discretely.

The discretization of the model has to be realized also in finite time. Therefore, the conventional discretization technique (Šulc, 1999)

$$\mathbf{X}(k+1) = e^{\mathbf{A}_C\delta}\mathbf{X}(k) + \int_{k\delta}^{k\delta+\delta} e^{\mathbf{A}_C(k\delta+\delta-\tau)}\mathbf{B}_C\,d\tau\,\mathbf{u}(k) \quad (15)$$

is provided by finite expansion of exponential function $e^{(\cdot)}$.

Finally, then the discrete state-space model is written as follows

$$\mathbf{X}(k+1) = \mathbf{A}\,\mathbf{X}(k) + \mathbf{B}\,\mathbf{u}(k)$$
$$\mathbf{y}(k) = \mathbf{C}\,\mathbf{X}(k) \quad (16)$$

This model (16) is initial form for further explanation of design of predictive algorithm.

3 PREDICTIVE CONTROL

Generalized Predictive Control is a multi-step control (Ordys and Clarke, 1993) as well as a similar approach – Linear Quadratic Control (Phillips and Nagle, 1995). It offers more powerful control actions than standard PID controllers and therefore it gains significant and widespread application in industrial process control. Its basic formulation can be adapted, without difficult modifications, for multi-input multi-output (MIMO) systems.

The control is based on local optimization of quadratic cost function (quadratic criterion)

$$J_k = \sum_{j=No+1}^{N}\Big\{\big(\mathbf{y}_{(k+j)} - \mathbf{w}_{(k+j)}\big)^T \mathbf{Q}_\mathbf{y}\,\big(\mathbf{y}_{(k+j)} - \mathbf{w}_{(k+j)}\big)\Big\} +$$
$$+ \sum_{j=1}^{Nu}\Big\{\mathbf{u}_{(k+j-1)}{}^T\,\mathbf{Q}_\mathbf{u}\,\mathbf{u}_{(k+j-1)}\Big\} \quad (17)$$

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

The predictive control combines together both feed-forward part and feed-back part. The former, feed-forward part is represented by prediction via mathematical model of the controlled system (parallel robot). It forms the dominant part of control actions. The latter, feed-back, closed from measured outputs, compensates some model inaccuracies and certain bounded disturbances.

In spite of mentioned incontestable advantages of predictive control, it can cause, in general point of view, occurrence of steady-state errors. It is happened not only when penalizations in quadratic cost function are nonzero but also e.g. when unmeasured disturbances occur.

It can be solved by modification of generalized predictive algorithm, which will be explained thereinafter.

3.1 Equations of Prediction

The prediction is fundamental part of the design. It defines the character of the algorithm. Generally, let us consider two types of algorithms:

- absolute algorithm (standard)
- incremental algorithm (modified standard)

Absolute algorithm generates directly values of the control actions, their full (absolute) values. The algorithm arises from the model (14) or (16) without any changes. On the other hand, incremental algorithm generates only increments of the control actions. To obtain incremental/integrative character, the integrator has to be added to the model of the system. The following lines show this addition

$$\ddot{\mathbf{y}} = \mathbf{f}(\mathbf{y},\dot{\mathbf{y}}) = \mathbf{a_1}(\mathbf{y},\dot{\mathbf{y}})\,\mathbf{y} + \mathbf{a_2}(\mathbf{y},\dot{\mathbf{y}})\,\dot{\mathbf{y}} + \mathbf{g}(\mathbf{y})\mathbf{u} \qquad (18)$$

$$\mathbf{u} = \int \mathbf{du}\; dt \quad \rightarrow \quad \mathbf{u}'(t) = \mathbf{du} = \tilde{\mathbf{u}} \qquad (19)$$

after insertion of equation (19) to (18), it is obtained

$$\mathbf{y}''' = (\mathbf{f}(\mathbf{y},\dot{\mathbf{y}}))' = \mathbf{a_1}\,\mathbf{y}' + \mathbf{a_2}\,\mathbf{y}'' + \mathbf{g}\,\tilde{\mathbf{u}} \qquad (20)$$

Then continuous state-space formulation is

$$\begin{bmatrix}\dot{\mathbf{x}}_1\\ \dot{\mathbf{x}}_2\\ \dot{\mathbf{x}}_3\end{bmatrix} = \begin{bmatrix}[\mathbf{0}] & [\mathbf{1} & \mathbf{0}]\\ \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix} & \begin{bmatrix}\mathbf{0} & \mathbf{1}\\ \mathbf{a}_1 & \mathbf{a}_2\end{bmatrix}\end{bmatrix}\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \mathbf{x}_3\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \begin{bmatrix}\mathbf{0}\\ \mathbf{g}\end{bmatrix}\end{bmatrix}\tilde{\mathbf{u}} \qquad (21)$$

$$\dot{\tilde{\mathbf{X}}} = \tilde{\mathbf{A}}_C\,\tilde{\mathbf{X}} + \tilde{\mathbf{B}}_C\,\tilde{\mathbf{u}} \;\Bigg|\; \tilde{\mathbf{X}} = \begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \mathbf{x}_3\end{bmatrix};\ \tilde{\mathbf{A}}_C = \begin{bmatrix}\mathbf{0} & [\mathbf{1} \;\; \mathbf{0}]\\ \mathbf{0} & \mathbf{A}_C\end{bmatrix};\ \tilde{\mathbf{B}}_C = \begin{bmatrix}\mathbf{0}\\ \mathbf{B}_C\end{bmatrix}$$

$$\mathbf{y} = \tilde{\mathbf{C}}_C\,\tilde{\mathbf{X}} \;\Big|\; \tilde{\mathbf{C}}_C = [\mathbf{C}_C \;\; \mathbf{0}];\ \mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]^T = [\mathbf{y}, \mathbf{y}', \mathbf{y}'']^T \qquad (22)$$

Extended state $\tilde{\mathbf{X}}$ is computed by state observer (Anderson and Moore, 1979). This modified model (22) has the same form as state-space model (14). If it is discretized according to (15), then obtained model is the same as (16). That form is generally correct and it will be used in the following text also for the modification (21) and (22).

Using considered discrete state-space form (16), the equations of prediction are usually expressed as

$$\begin{aligned}
\hat{\mathbf{X}}_{(k+1)} &= \mathbf{A}\;\mathbf{X}_{(k)} + \mathbf{B}\mathbf{u}_{(k)}\\
\hat{\mathbf{y}}_{(k+1)} &= \mathbf{C}\mathbf{A}\;\mathbf{X}_{(k)} + \mathbf{C}\,\mathbf{B}\mathbf{u}_{(k)}\\
&\;\;\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad \ddots\\
\hat{\mathbf{X}}_{(k+N)} &= \mathbf{A}^N\mathbf{X}_{(k)} + \mathbf{A}^{N-1}\mathbf{B}\mathbf{u}_{(k)} + \dots + \mathbf{B}\mathbf{u}_{(k+N-1)}\\
\hat{\mathbf{y}}_{(k+N)} &= \mathbf{C}\mathbf{A}^N\mathbf{X}_{(k)} + \mathbf{C}\mathbf{A}^{N-1}\mathbf{B}\mathbf{u}_{(k)} + \dots + \mathbf{C}\,\mathbf{B}\mathbf{u}_{(k+N-1)}
\end{aligned} \qquad (23)$$

and in matrix notation, they are given that way

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

$$\mathbf{f} = \begin{bmatrix}\mathbf{CA}\\ \vdots\\ \mathbf{CA}^N\end{bmatrix}\mathbf{X}(k),\ \mathbf{G} = \begin{bmatrix}\mathbf{C}\quad\mathbf{B} & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots\\ \mathbf{C}\mathbf{A}^{N-1}\mathbf{B} & \cdots & \mathbf{CB}\end{bmatrix} \qquad (24)$$

Vector **f** represents free responds (**u** = **0**) from time instant *k*. The product **G u** compensates differences of free responds from desired values within horizon of optimization *N* (Ordys and Clarke, 1993).

3.2 Computation of Control Actions

The control actions are obtained by minimization of quadratic criterion (17). It can be simply rewritten to the following matrix product (Belda et al. 2002)

$$J_k = [(\hat{\mathbf{y}} - \mathbf{w})^T, \mathbf{u}^T]\underbrace{\begin{bmatrix}\mathbf{Q}_\mathbf{y} & \mathbf{0}\\ \mathbf{0} & \mathbf{Q}_\mathbf{u}\end{bmatrix}}_{\mathbf{J}^T}\;\underbrace{\begin{bmatrix}\mathbf{Q}_\mathbf{y} & \mathbf{0}\\ \mathbf{0} & \mathbf{Q}_\mathbf{u}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{y}} - \mathbf{w}\\ \mathbf{u}\end{bmatrix}}_{\mathbf{J}} = \mathbf{J}^T \times \mathbf{J} \qquad (25)$$

where $\hat{\mathbf{y}}$ is a vector composed according to (24) (time step *k*+1, · · ·, *k*+*N*), **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*, · · ·, *N* - 1).

The product (25) 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 straight-forward.

If the square root of the criterion on the right side is selected and expression of prediction (24) is inserted in this square root, then the criterion is given

$$\mathbf{J} = \begin{bmatrix}\mathbf{Q}_\mathbf{y} & \mathbf{0}\\ \mathbf{0} & \mathbf{Q}_\mathbf{u}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{y}} - \mathbf{w}\\ \mathbf{u}\end{bmatrix} = \begin{bmatrix}\mathbf{Q}_\mathbf{y}\,\mathbf{G}\\ \mathbf{Q}_\mathbf{u}\end{bmatrix}\mathbf{u} - \begin{bmatrix}\mathbf{Q}_\mathbf{y}\,(\mathbf{w} - \mathbf{f})\\ \mathbf{0}\end{bmatrix} \qquad (26)$$

J is a column vector and its Euclidean norm equals a cost of the square root of the criterion.

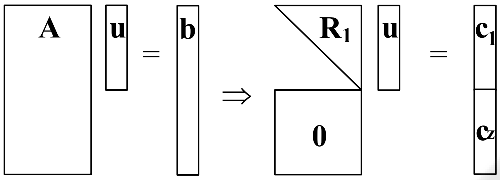
The objective is to search for such **u**, which minimizes the square root (26) i.e. the control **u** minimizes the norm |**J**| of the criterion. In case of square root (26), the minimization leads to a system of algebraic equations with more rows than columns – over-determined system:

$$\begin{bmatrix} \mathbf{Qy\,G} \\ \mathbf{Qu} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{Qy\,(w-f)} \\ \mathbf{0} \end{bmatrix} = \mathbf{0}$$

$$\mathbf{A} \quad \mathbf{u} - \quad \mathbf{b} \quad = \mathbf{0} \qquad (27)$$

For optimization of the criterion, the orthogonal triangular decomposition (Golub and Van, 1989; Lawson and Hanson, 1974) is used. It reduces excess rows of matrix **A** [$(2{\cdot}N{\cdot}i)\times(N{\cdot}i)$] and elements of vector **b** [$2{\cdot}N{\cdot}i$] (i is a number of DOF) into upper triangular matrix **R** and a vector **c** according to the following scheme:

$$\begin{aligned} \mathbf{A\,u} &= \mathbf{b} \qquad /\mathbf{Q}^T \\ \mathbf{Q}^T\mathbf{A\,u} &= \mathbf{Q}^T\mathbf{b} \\ \mathbf{R\,u} &= \mathbf{c} \end{aligned} \qquad (28)$$



(29)

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

$$\begin{aligned} \mathbf{R}_1\,\mathbf{u} &= \mathbf{c}_1 \\ \mathbf{u} &= (\mathbf{R}_1)^T\,\mathbf{c}_1 \end{aligned} \qquad (30)$$

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

The penalizations **Qu** and **Qy** (usually selected as $\mathbf{Qy}=diag(\lambda_y)$, $\lambda_y=1$ and $\mathbf{Qu}=diag(\lambda_u)$, $\lambda_u \in \langle 0,1 \rangle$) determines magnitude of the redistributed loss in considered horizon of the prediction N.

The horizons Nu and No have not direct utilization here. Control horizon Nu is usually equal horizon of prediction N; lower values provide equality of control actions at the end of optimization horizon – useless for robot motion. Initial insensitivity horizon No is also directly useless. It causes, that control differences at the beginning of the horizon N are not considered.

The different choice of ratio of penalization λ_u/λ_y together with horizon N enables to generate control actions that the available drives were not fitfully exerted. However, distributed changes of torques are achieved with the cost of certain loss (error), that theoretically equals value of the criterion.

3.3 Quadraticaly-Optimal Trajectories

As one interesting possibility, the predictive control offers, due to its several horizons (N, No, Nu), planning trajectories by record 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 (Figure 2). 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 (27), are reduced only on their last k rows and elements respectively

$$\left[\begin{array}{cc:c} \multicolumn{2}{c:}{\overline{\mathbf{G}}} & \mathbf{w-f} \\ \hdashline 1 & 0 & \mathbf{0} \\ & \ddots & \vdots \\ 0 & 1 & \mathbf{0} \end{array}\right] = \left[\begin{array}{cccccc:c} \mathbf{CA}^{N-k}\mathbf{B} & \cdots & \mathbf{CB} & \cdots & \mathbf{0} & & (\mathbf{w-f})_{N-k} \\ \vdots & & & \ddots & \vdots & & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & \cdots & \mathbf{CAB} & \mathbf{CB} & & (\mathbf{w-f})_N \\ \hdashline \mathbf{1} & \cdots & & \cdots & \mathbf{0} & & \mathbf{0} \\ \vdots & & \mathbf{1} & & \vdots & & \vdots \\ \mathbf{0} & \cdots & & \cdots & \mathbf{1} & & \mathbf{0} \end{array}\right] \qquad (31)$$

Lower unit matrix in (31) 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.

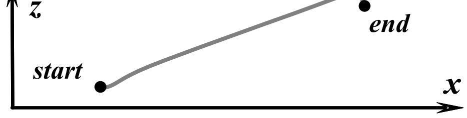


Figure 2: One example of planned trajectory

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

$$N := Nmax, Nmax - 1, \cdots, Nmin + 1, Nmin \qquad (32)$$

where the value $Nmin$ is suitable selected, not exceed number approx. 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.

4 LABORATORY EXPERIMENTS RESULTS AND CONCLUSIONS

For laboratory tests, robot ‘Sliding Star’ was used. It represents vertical planar parallel configuration with different levels of potential energies (i.e. gravitational force has to be considered) and redundant actuation. The aim of the experiments was control based on described predictive algorithms (control circuit in Figure 4) fulfilling a given trajectory (time histories of control process Figure 5 and 6).

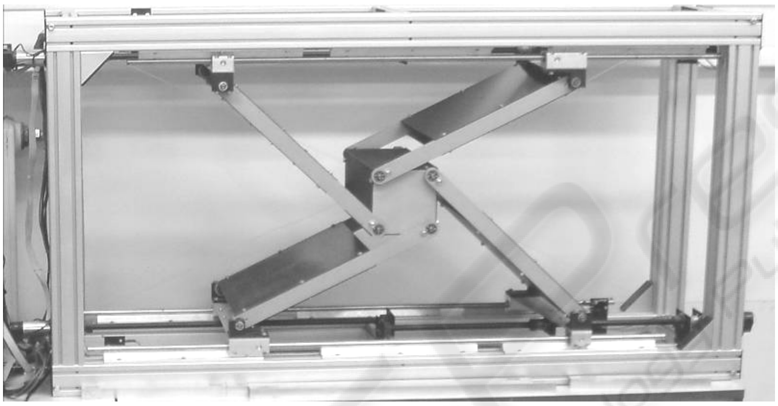


Figure 3: Lab model of parallel robot ‘Sliding Star’

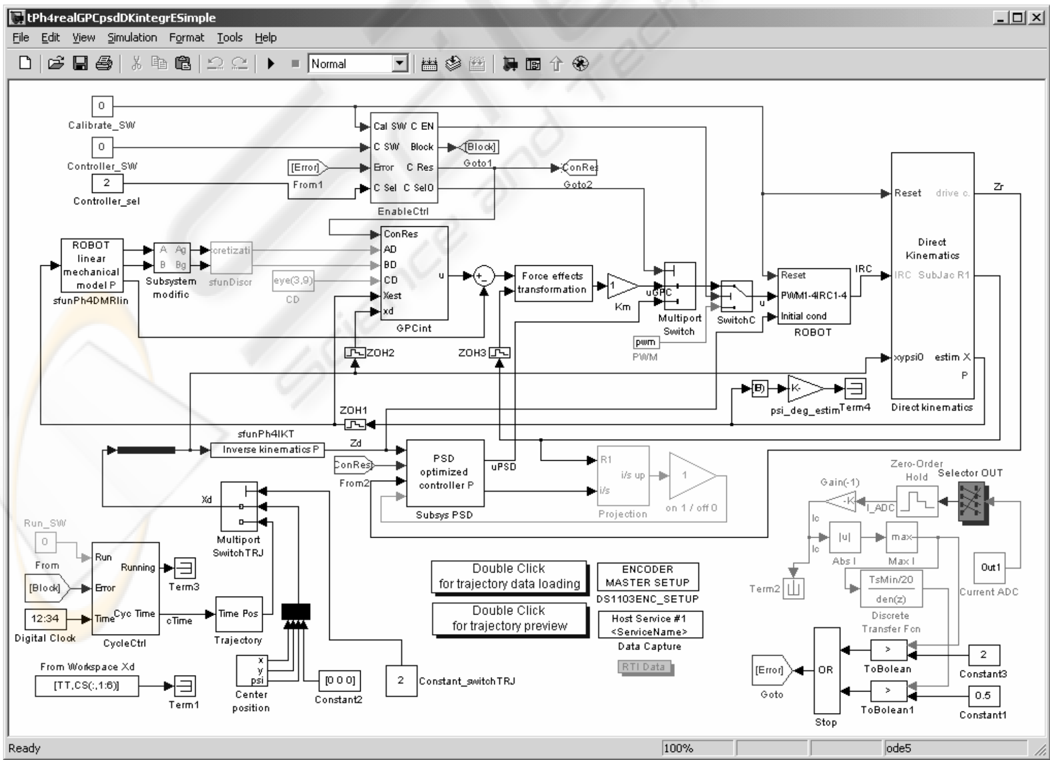


Figure 4: Control scheme for predictive control and PSD controller for comparison (discrete form of PID)

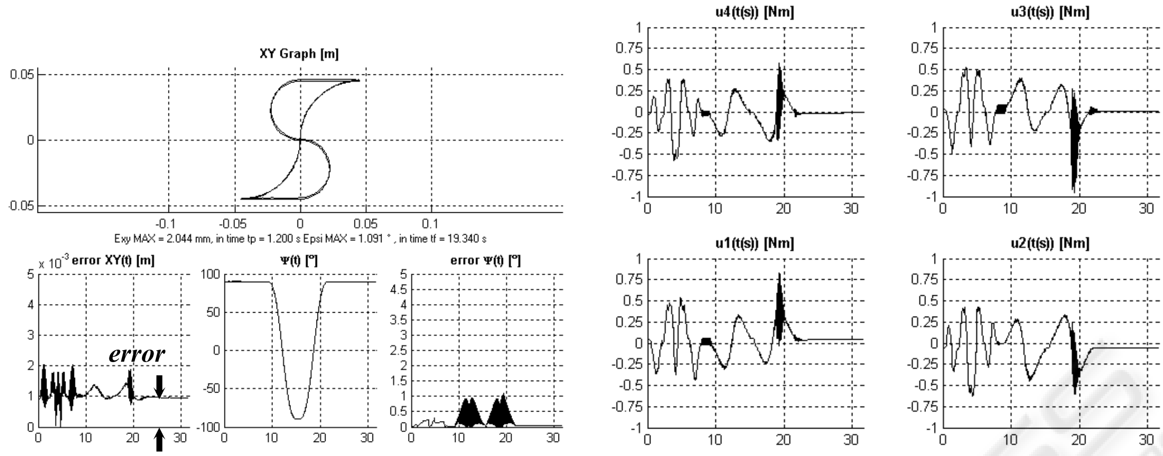


Figure 5: Absolute GPC – Steady state error is evident (ts = 0.02s; N = 10, Nu = 10, No = 0, λ_u = 1e-6)

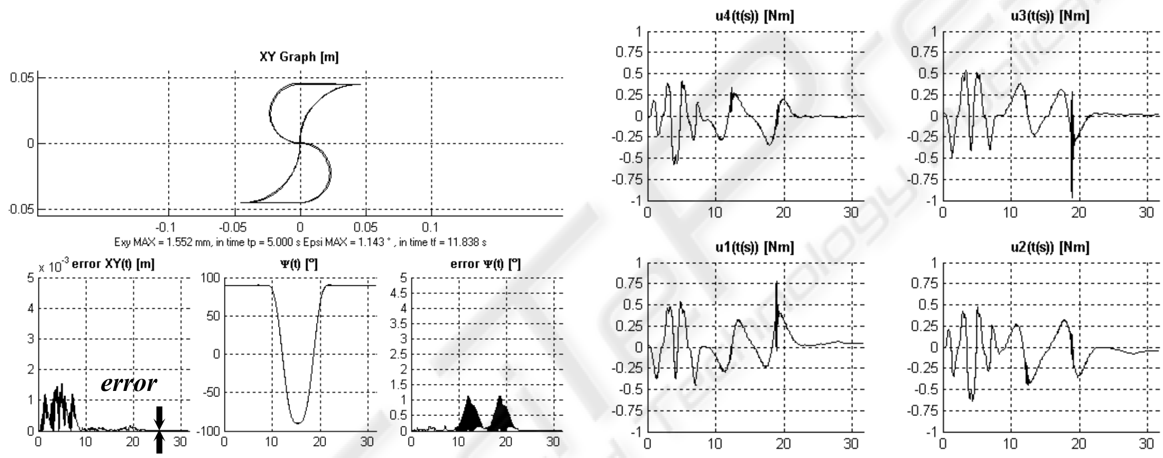


Figure 6: Incremental GPC – Steady state error converges to zero (ts = 0.02s; N = 10, Nu = 9, No = 1, $\Delta\lambda_u$ = 1e-6)

The control circuit (Figure 4) is composed in MATLAB-Simulink environment, where the predictive control algorithm itself was realized as capsulized C-coded s-function in Simulink block (Belda et al. 2004). This control circuit (Simulink scheme), after its compilation and uploading to digital signal processor, was used for real time tests of control on laboratory model of parallel robot 'Sliding Star' (Figure 3). The robot model has three degrees of freedom (movement in direction x and z and rotation ψ around perpendicular axis y), but it is redundantly actuated by fourth rotational DC motors with gearings and motion screws. The screws provide transformation of rotation of the motors to straight-line motion.

The circuit contains not only the block of controller and block representing robot interface, but also other necessary blocks as blocks of generating trajectory, model composition, discretization, kinematical transformations, measurement of current, safety logical blocks etc.

The Figures 5 and 6 well present the difference of the results of real control process with different predictive algorithms.

In the upper parts of figure left sides, the testing trajectory (shape ‚S') is shown. In lower parts, there are consecutively time histories of position error, desired rotation (besides rectilinear motion the robot performs also rotation), and just error of rotation.

On right sides of the figures, there are times histories of foursomes control actions – real values of torques on appropriate drives.

The Figure 6 shows the improvement of control process, when the incremental modification of predictive algorithm is used. Indicated error in position in steady-state (let us say undesirable offset) is evident in Figure 5. On the other hand, the error in Figure 6 converges to zero.

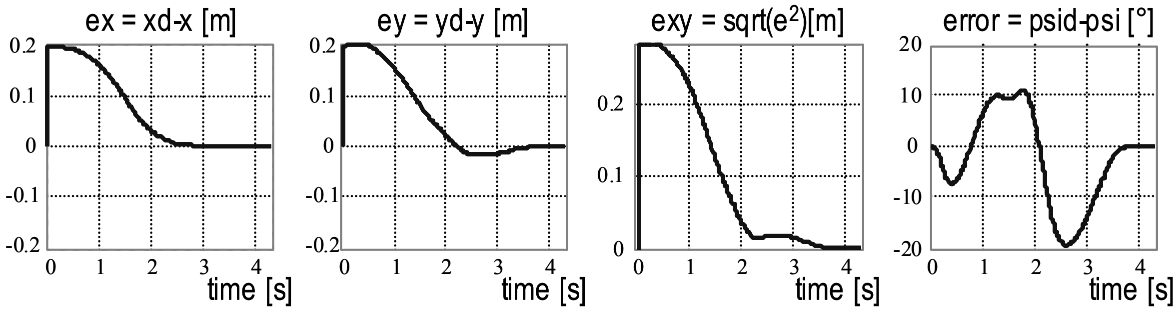


Figure 7: Time histories of differences at quadratically-optimal trajectory planning for trajectory in Figure 8

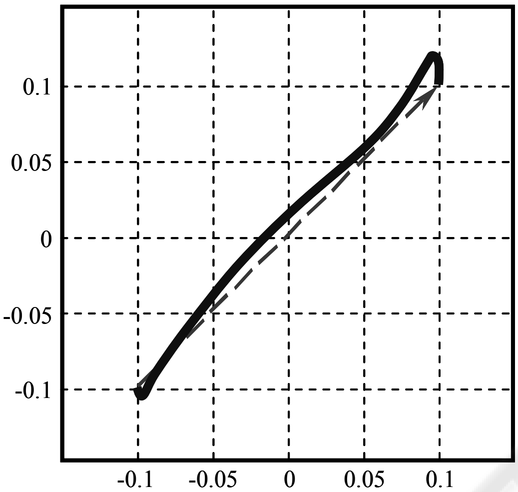


Figure 8: Really planned trajectory for 'Sliding Star'

On the basis of shown results (representative selection only) model-based control can be effective. It can achieve acceptable control errors. Model-based control can also meet some additional requirements not only from control point of view. One of examples can be quadratically-optimal trajectory planning (example in Figures 7 and 8) described in subsection 3.2. The presented simple trajectory planning is basic idea, which can solve avoidance of obstacles not only in robot workspace.

ACKNOWLEDGEMENTS

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), "Methods of Predictive Control, Algorithms and Implementation".

REFERENCES

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

Belda, K., 2002. *Control of Redundant Parallel Structures of Robotic Systems*, Dissertation, Czech Technical University in Prague.

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

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

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.

Neugebauer, R. ed., 2002. *Development Methods and Application Experience of Parallel Kinematics*, Reports from the IWU Vol. 16, ISBN 3-928921-76-2, IWU Chemnitz.

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.

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

Stejskal, V. and Valášek, M. 1996. *Kinematics and Dynamics of Machinery*, Marcel Dekker, New York.

Šulc, B., 1999. *Theory of automatic control*. Script of the Czech Technical University, Prague (in Czech).

Tsai, L.-W., 1999. *Robot analysis, The Mechanics of Serial and Parallel Manipulators*, John Wiley & sons, Inc., New York.

Valášek, M. and Steinbauer, P., 1999. Nonlinear Control of Multibody Systems, *Euromech*, pp. 437-444, Lisabon.