PREDICTIVE CONTROL APPLIED TO PLANAR PARALLEL ROBOTS

Květoslav Belda, Josef Böhm, Michael Valášek

Dept. of Adaptive Systems, Institute of Information Theory and Automation,
Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 182 08 Prague 8

Abstract: The design of suitable control is essential part of application of parallel robots. Together with appropriate control, the parallel robots are promising way to significantly improve accuracy and speed of machine tools in industrial production. This paper deals with the design and results of one possible control approach represented by model-based predictive control in absolute formulation. It explains exact linearization of used, initially nonlinear models, which are sequentially transformed to discrete state-space form. Control algorithm is derived in square-root form. The results are substantiated by real laboratory experiments showed in several representative figures. *Copyright © 2005 IFAC*

Keywords: Predictive control, Redundant manipulators, Robotics, State-space realization.

1. INTRODUCTION

Nowadays, the future development of industrial robots - machine tools - requires change in their control. It means replacement of traditional control included in numerical control (NC) systems, which cannot fully utilize mechanical properties of the machine. This approach is only general way providing control of the tool drives as separate units, but it is not approach solving the control from view of the whole machine system. This decentralization is suitable for production lines, however is not effective from view of one workplace - machine tool.

NC programs represented by setting of the sequence of coordinates and technological conditions and traditional control based on PID structures do not take into account particularity of given mechanical structure constituting the machine. One possible way, relatively accurate, but more complex, is a utilization of model-based control approaches. They could be initially implemented directly to the NC programs as user functions or later, after demonstration of their functionality, implemented in new control systems.

At present, the model-based approaches are used almost only in laboratories without any adequate interface: user - control system - machine tool. They wait for proper time, when the customers will require them. Till that time, they will be developed locally, as individual applications.

One of developing fields of machine tools, which can fully utilize model-based control, is parallel robotics.

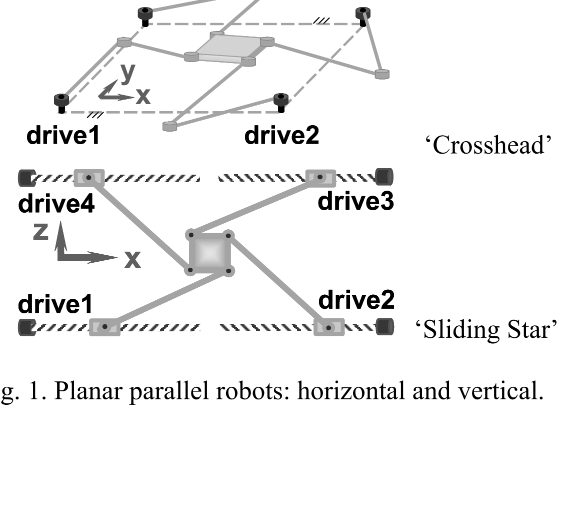


Fig. 1. Planar parallel robots: horizontal and vertical.

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

This paper is focused on questions if model-based control is more effective than PID structures; if it can achieve smaller control errors and moreover if model-based control can meet some additional control requirements. It deals with the design and real-time realization of one example of model-based control based on absolute predictive algorithm. It was tested on two models of parallel robots (Fig. 1.).

2. MODEL-BASED APPROACH TO CONTROL

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 (Sciavicco and Siciliano, 1996).

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

2.1 Model of mechanical system (parallel robot)

Generally, the robot is a multibody system. Its model is represented by pure equations of motion. They are composed mostly from Lagrange's equations (Stejskal and Valášek, 1996). Then the model is given by equation

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

where input vector **u** can represent only forces, caused by torques on drives in case of horizontal configuration

$$\mathbf{u} = \mathbf{F}_{\mathcal{T}} \qquad (2)$$

or these forces enlarged by force effects from gravitation in case of vertical configuration

$$\mathbf{u} = -\mathbf{F}_{g} + \mathbf{F}_{\mathcal{T}} \qquad (3)$$

Equations (2) and (3) serve for final determination of necessary torques on drives. Furthermore, this arrangement provides the equality $\mathbf{f}(\mathbf{y}, \dot{\mathbf{y}}) = \mathbf{0}$ for arbitrary **y** from range of definition and zero time derivatives $\dot{\mathbf{y}} = \mathbf{0}$ in spite of presence of gravitational forces. These forces have been already included in inputs (equation (3)).

The function $\mathbf{f}(\mathbf{y}, \dot{\mathbf{y}})$ produces that equation (1) is nonlinear for state $\mathbf{X} = [\mathbf{y}, \dot{\mathbf{y}}]^T = [\mathbf{x}_1, \mathbf{x}_2]^T$. In this paper, the generalized predictive control is used. Its algorithm works with linear models or possibly with linearized models. In the following subsection one linearizeing exact decomposition is introduced.

2.2 Exact linearization

This subsection deals with the linearization based on differences. The nonlinear part of equation (1) can be linearized as follows (Valášek and Steinbauer, 1999)

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

It is assumed 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; **X** belongs to the range of definition of the function; zero elements in **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.

$$\mathbf{X} = [\mathbf{y}, \dot{\mathbf{y}}]^T = [\mathbf{x}_1, \mathbf{x}_2]^T = [[x_{11}, x_{12}, \cdots], [x_{21}, x_{22}, \cdots]]^T \qquad (5)$$

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

$$\mathbf{f}(\mathbf{y}, \dot{\mathbf{y}}) = \mathbf{f}(\mathbf{X}) = \mathbf{0}\Big|_{\mathbf{X}=\mathbf{X}_r=[\mathbf{x}_{r1}\text{arbitrary},\mathbf{x}_{r2}=\mathbf{0}]} \qquad (6)$$

Then the decomposition indicated by equation (4) can be reached. Its algorithm starts from second state variables $\mathbf{x}_2$, i.e. according to equation (5): order is

$$\{x_{21}, x_{22}, \cdots, x_{11}, x_{12}, \cdots\} \text{ i.e. } \{21, 22, \cdots, 11, 12, \cdots\} \qquad (7)$$

The order is given by amount of the information included in the function **f**(**X**) and assumption (6). The indicated order selection will considerably simplify decomposition, as it will be shown later.

In view of these assumptions, the exact linearization - decomposition is based on differences

$$\mathbf{f}(\mathbf{X}) = \frac{\Delta\mathbf{f}(\circ)}{.\Delta x_{11}}\Delta x_{11} + \frac{\Delta\mathbf{f}(\circ)}{.\Delta x_{12}}\Delta x_{12} + \cdots +$$
$$+ \frac{\Delta\mathbf{f}(\circ)}{.(x_{21} - 0)}(x_{21} - 0) + \frac{\Delta\mathbf{f}(\circ)}{.(x_{22} - 0)}(x_{22} - 0) + \cdots \qquad (8)$$

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

In detail, equation (8) is written that way

$$\mathbf{f}(\mathbf{X}) =$$

$$= \frac{\mathbf{f}([x_{11}, x_{12}, x_{13}, 0,0,0]^T) - \mathbf{f}([x_{r11}, x_{12}, x_{13}, 0,0,0]^T)}{.(x_{11} - x_{r11})}(x_{11} - x_{r11}) + \cdots$$
$$+ \frac{\mathbf{f}([x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}]^T) - \mathbf{f}([x_{11}, x_{12}, x_{13}, 0, x_{22}, x_{23}]^T)}{.x_{21}} x_{21} + \cdots \tag{9}$$

The individual fractions of equation (9) 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

$$\mathbf{a}_1(\mathbf{y},\dot{\mathbf{y}}) = \mathbf{0}\,, \quad \mathbf{a}_2 = \begin{bmatrix} \mathbf{a}_{2_{11}} & \mathbf{a}_{2_{12}} & \cdots \\ \mathbf{a}_{2_{21}} & \mathbf{a}_{2_{22}} & \\ \vdots & & \ddots \end{bmatrix} \tag{10}$$

The first column group (matrix $\mathbf{a}_1(\mathbf{y},\dot{\mathbf{y}})$) contains only zeros because it is composed of 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

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

2.3 Discrete state-space formulation

Let the linear or linearized differential equation or system of differential equations are assumed

$$\ddot{\mathbf{y}} = \mathbf{f}(\mathbf{y},\dot{\mathbf{y}}) = \mathbf{a}_1(\mathbf{y},\dot{\mathbf{y}})\,\mathbf{y} + \mathbf{a}_2(\mathbf{y},\dot{\mathbf{y}})\,\dot{\mathbf{y}} + \mathbf{g}(\mathbf{y})\mathbf{u} \tag{12}$$

then they can be written in state-space form

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{a}_1 & \mathbf{a}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}(\mathbf{x}_1) \end{bmatrix} \mathbf{u}$$
$$\mathbf{y} = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \tag{13}$$

in matrix notation

$$\dot{\mathbf{X}} = \mathbf{A}_C \; \mathbf{X} + \mathbf{B}_C \; \mathbf{u}$$
$$\mathbf{y} = \mathbf{C}_C \; \mathbf{X} \tag{14}$$

For real-time control the system (14) has to be discretized, because continuous realization is not possible. 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)

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

is provided by finite expansion as follows

$$\mathbf{A} = e^{\mathbf{A}_C\delta} = \mathbf{1} + \mathbf{A}_C\delta + \frac{\mathbf{A}_C{}^2\delta^2}{2!} + \cdots + \frac{\mathbf{A}_C{}^r\delta^r}{r!} + rest(Oh^{r+1}) \tag{16}$$

and $\mathbf{B} = \int_{k\delta}^{k\delta+\delta} e^{\mathbf{A}_C(k\delta+\delta-\tau)}\,\mathbf{B}_C\, d\tau =$

$$= (\mathbf{1}\delta + \frac{\mathbf{A}_C\delta^2}{2!} + \cdots + \frac{\mathbf{A}_C{}^{r-1}\delta^r}{r!} + rest(Oh^{r+1}))\,\mathbf{B}_C \tag{17}$$

Then the model in discrete state-space formulation is written in conventional form

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{A}\,\mathbf{X}(k) + \mathbf{B}\,\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\,\mathbf{X}(k) \end{aligned} \tag{18}$$

This model (equation (18)) is initial form for further explication of control design based on generalized predictive algorithm.

3. PREDICTIVE CONTROL

Generalized Predictive Control is a multi-step control as well as similar approach – Linear Quadratic Control (Ordys and Clarke, 1993; Phillips and Nagle, 1995). It is based on local optimization of quadratic cost function (quadratic criterion)

$$J_k = \sum_{j=No+1}^{N} \left\{ \left(y_{(k+j)} - w_{(k+j)} \right)^T Q_y \left(y_{(k+j)} - w_{(k+j)} \right) \right\} +$$
$$+ \sum_{j=1}^{Nu} \left\{ u_{(k+j-1)}{}^T Q_u \, u_{(k+j-1)} \right\} \tag{19}$$

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. *Qy* and *Qu* are output and input penalizations and *y* (*k*+*j*) and *u* (*k*+*j* −1) are input and output values.

The predictive control combines both feed-forward part and feed-back part. The 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 feed-back, closed from measured outputs, compensates some inaccuracies of the model and certain bounded disturbances.

3.1 Predictions in horizon of optimization

The base of the predictive control is a prediction (computation, estimation) of future output values. These values are compared with desired values in the criterion. Using the discrete state-space formulation, the prediction is usually expressed as

$$\begin{aligned}
\hat{\mathbf{X}}_{(k+1)} &= \mathbf{A}\;\mathbf{X}_{(k)} + \mathbf{Bu}_{(k)} \\
\hat{\mathbf{y}}_{(k+1)} &= \mathbf{C}\mathbf{A}\;\mathbf{X}_{(k)} + \mathbf{C}\;\mathbf{Bu}_{(k)} \\
&\vdots \qquad \vdots \qquad \vdots \qquad \ddots \\
\hat{\mathbf{X}}_{(k+N)} &= \mathbf{A}^{N}\,\mathbf{X}_{(k)} + \mathbf{A}^{N-1}\,\mathbf{Bu}_{(k)} + \cdots + \mathbf{Bu}_{(k+N-1)} \\
\hat{\mathbf{y}}_{(k+N)} &= \mathbf{C}\mathbf{A}^{N}\,\mathbf{X}_{(k)} + \mathbf{C}\mathbf{A}^{N-1}\,\mathbf{Bu}_{(k)} + \cdots + \mathbf{C}\mathbf{Bu}_{(k+N-1)}
\end{aligned} \tag{20}$$

and in matrix notation is given that way

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

$$\mathbf{f} = \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{N} \end{bmatrix} \mathbf{x}(k),\; \mathbf{G} = \begin{bmatrix} \mathbf{C} & \mathbf{B} \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{A}^{N-1}\,\mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix} \tag{21}$$

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* (Belda, *et al.*, 2003).

3.2 Computation of control actions

The control actions are obtained by minimization of quadratic criterion (19). It can be simply rewritten to the following matrix product

$$J_k = \underbrace{[(\hat{\mathbf{y}}-\mathbf{w})^T, \mathbf{u}^T] \begin{bmatrix} \mathbf{Qy} & \mathbf{0} \\ \mathbf{0} & \mathbf{Qu} \end{bmatrix}}_{} \underbrace{\begin{bmatrix} \mathbf{Qy} & \mathbf{0} \\ \mathbf{0} & \mathbf{Qu} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}}-\mathbf{w} \\ \mathbf{u} \end{bmatrix}}_{} = [\quad \mathbf{J}^T \quad][\quad \mathbf{J} \quad] \tag{22}$$

where $\hat{\mathbf{y}}$ is a vector composed according to (21) (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 (22) 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 straightforward.

If the square root of the criterion on the right side is selected and expression of prediction (21) is inserted in this square root, then the criterion is given

$$\mathbf{J} = \begin{bmatrix} \mathbf{Qy} & \mathbf{0} \\ \mathbf{0} & \mathbf{Qu} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}}-\mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{Qy}\,\mathbf{G} \\ \mathbf{Qu} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{Qy}\,(\mathbf{w}-\mathbf{f}) \\ \mathbf{0} \end{bmatrix} \tag{23}$$

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

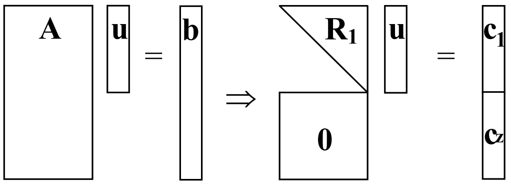
The goal is to search for such **u**, which minimizes the square root (23) i.e. the control **u** should minimize the norm |**J**| of the criterion. It is fulfilled, if the expression (23) (over-determined system, i.e. the system of algebraic equations with more rows than columns) is minimized.

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

$$\mathbf{A} \quad \mathbf{u} - \quad \mathbf{b} \quad = \mathbf{0} \tag{24}$$

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·*N*·*i*)×(*N*·*i*)] and elements of vector **b** [2·*N*·*i*] (*i* is a number of DOF) into upper triangular matrix **R** and vector **c** according to the following scheme:

$$\begin{aligned}
\mathbf{Au} &= \mathbf{b} \qquad /\mathbf{Q}^T \\
\mathbf{Q}^T\mathbf{Au} &= \mathbf{Q}^T\mathbf{b} \\
\mathbf{Ru} &= \mathbf{c}
\end{aligned} \tag{25}$$



(26)

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 actions **u**, only upper part of the system (26) is need

$$\begin{aligned}
\mathbf{R}_1\,\mathbf{u} &= \mathbf{c}_1 \\
\mathbf{u} &= (\mathbf{R}_1)^T\,\mathbf{c}_1
\end{aligned} \tag{27}$$

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 the 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 to 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 differences at the beginning of the horizon *N* are not considered.

The different choice of ratio of penalizations 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.

4. LABORATORY TESTS AND PRACTICAL RESULTS

For laboratory tests, two trial models ‘Crosshead’ (Fig. 3.) and ‘Sliding Star’ (Fig. 4.) were used. The former structure represents planar horizontal configuration. The latter structure – ‘Sliding Star’ represents vertical planar configuration with different levels of potential energies. Both structures are redundantly actuated. The aim is control along a planned trajectory (e.g. Fig. 2).

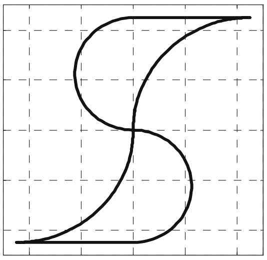


Fig. 2. Trajectory of the robot motion.

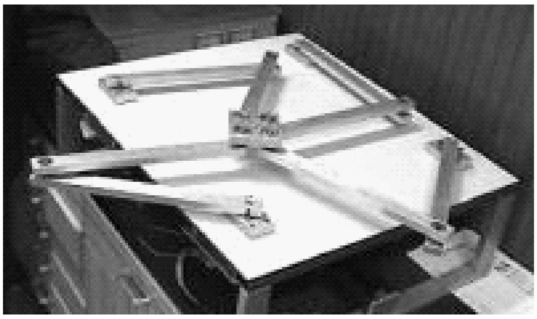


Fig. 3. Lab model of parallel robot ‘Crosshead’.

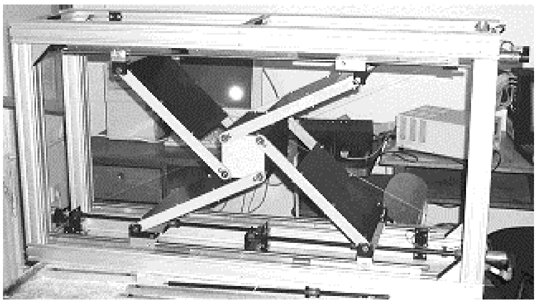


Fig. 4. Lab model of parallel robot ‘Sliding Star’.

Control was realized under MATLAB-Simulink environment as S-functions in C code. The scheme is in Fig. 5. For comparison, the conventional PSD controller was appended.

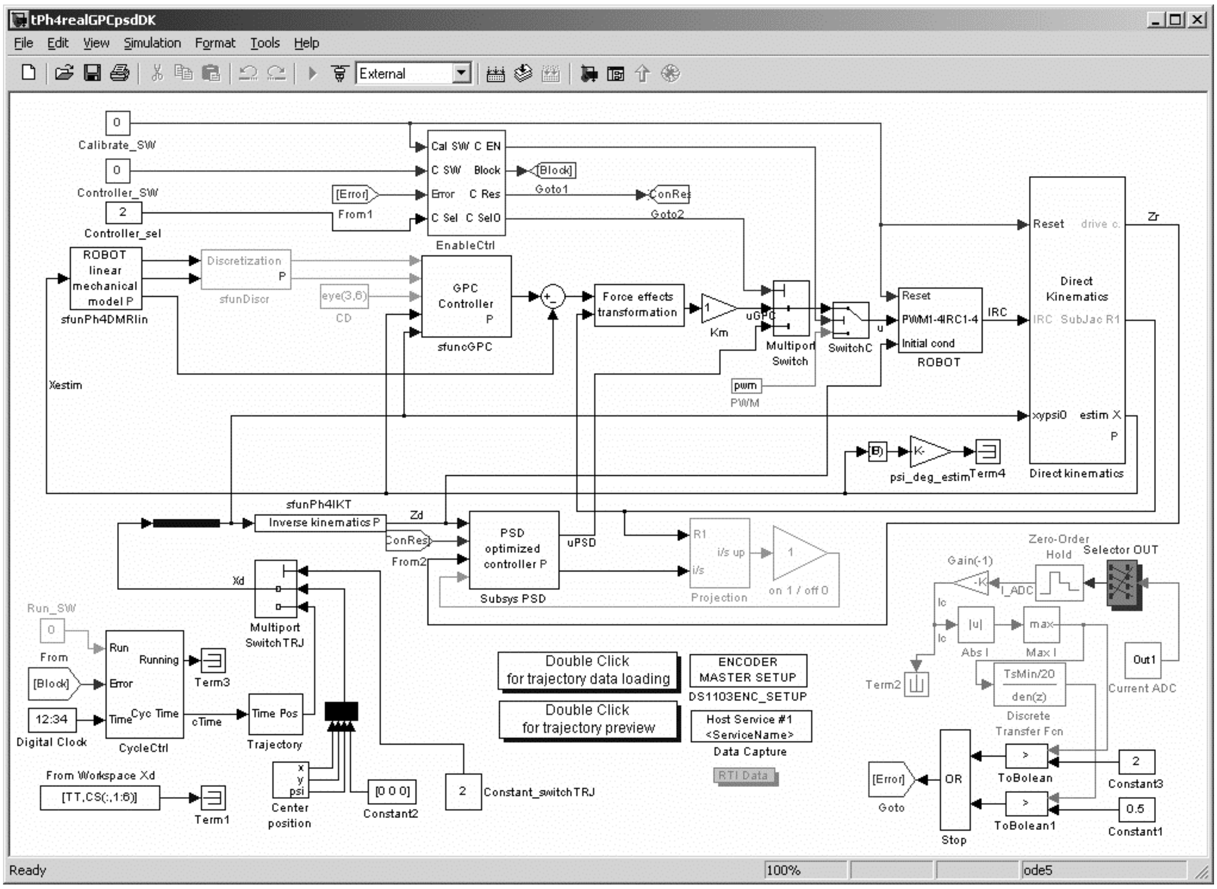


Fig. 5. Control scheme for both Generalized Predictive Control and PSD controller (discrete form of PID).

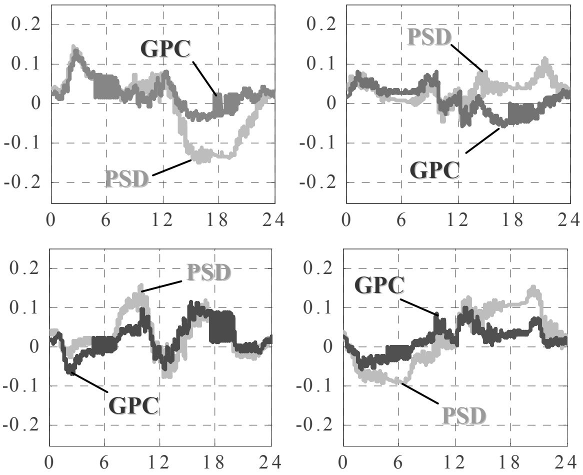


Fig. 6. Comparison of time histories of torques [Nm] (vertical axes) with Predictive Control (absolute algorithm) and PSD control. Crosshead.

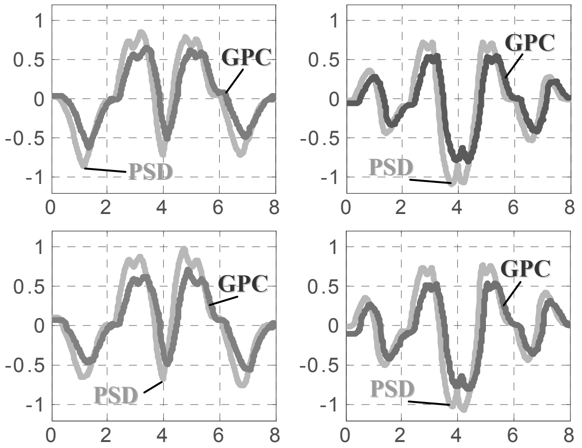


Fig. 7. Comparison of time histories of torques [Nm] (vertical axes) with Predictive Control (absolute algorithm) and PSD control. Sliding Star.

Figs. 6, 7 show, that predictive algorithm needs only 75% of energy, in comparison with PSD. Maximal measured control error of predictive control averages also 75% of PSD in spite of inaccuracy of direct kinematics used for determination of model coordinates.

Moreover, the predictive control achieves smaller control errors even without summative (integrative) channels, which can cause problems within redundantly actuated robots. However, it causes steady state errors about 5%.

5. CONCLUSION

This paper was focused on application of model-based approach represented by discrete generalized predictive control algorithm. It compared this approach with conventional PSD controllers. On the basis of shown results in section 4 (only representative selection), the initial questions can be answered. Model-based control can be more effective than PID/PSD structures, it can achieve smaller control errors on condition that mathematical model is relatively accurate. Model-based control can also meet some additional control requirements, because it can add them to the criterion and such a way can find adequate control actions.

ACKNOWLEDGEMENTS

The authors appreciate the kind support by the grant GAČR (101/03/0620, 2003/2005), “Redundant drives and measurement for hybrid machine tools”.

REFERENCES

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*, **Vol. 31**, No. 3, pp. 413-432.

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

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. (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, Inc. 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 99*, pp. 437 – 444, Lisabon.