CONCEPTS OF MODEL-BASED CONTROL AND TRAJECTORY PLANNING FOR PARALLEL ROBOTS

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, Czech Republic
belda@utia.cas.cz, bohm@utia.cas.cz

Pavel Píša
Department of Control Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo nám. 13, 121 35 Prague 2, Czech Republic
pisa@cmp.felk.cvut.cz

ABSTRACT

The paper deals with the concepts of model-based control and trajectory planning intended for industrial parallel robots. These robots are characterized by very good dynamical properties arisen from small number of moving masses in comparison with conventional configurations. In the paper, multi-level (hierarchical) control will be investigated. It can be specified as a model-based control providing positional and speed loops with addition of fast low-level current-loop control. As a suitable representative of model-based control, predictive control is considered. Described concept can offer more possibilities to manage the control process than usual cascade control. Finally, the paper outlines two different concepts of trajectory planning. The first concept considers only pure geometrical features (curve-based planning) without relation to the real robots. The second concept conversely takes into account the dynamical features of the real robot with initial and final points (point-to-point planning).

KEY WORDS

Multi-level control, predictive control, trajectory planning

1. Introduction

The further development of industrial robots, machine tools and centers depends on capabilities of used control. In practice, conventional cascade PID control is usually used [10] (Fig. 1). It represents only local (decentralized, independent) way of control [4]. The question is if such way is sufficient, safe and economic for new machine development [7].

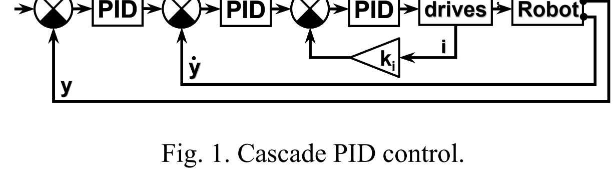

Fig. 1. Cascade PID control.

In the last decade, the research and development was renewed in the direction of the robots based on parallel structures. These structures are useful for constructions of machine tools and their centers, from which the high flexibility and productivity is expected. The first parallel structures were appeared in the sixties ([1] Fig. 2). In general, they can be simply understood as movable truss constructions or as movable platforms supported by several links, where the movable platform serves as a place for a tool or for a gripper. They represent closed-loop kinematical structures. Their flexibility (high dynamics) allowing high productivity is possible due to small number of moving masses.

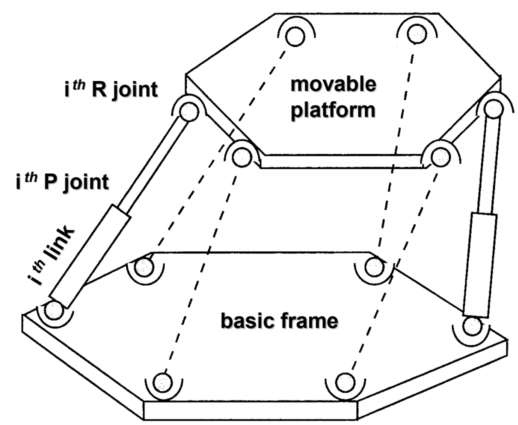

Fig. 2. Steward platform.

The main advantages of parallel structures can be formulated as follows:

Parallel structures

- make possible to fix almost all drives directly on basic frame without loading of movable parts of the robots, and thus contribute to the decrease of inertial forces
- contribute to the increase of the robot stiffness without losing dexterity due to properties of truss constructions; this improves position accuracy

In regards to the question of control, such robots need appropriate control strategy providing cooperation of each drive in parallel structure. Conventional local techniques like PID control have no energy optimization [4]. It limits robot capacity. In some cases, it is not safe, because the interrelations among individual drives through parallel links and movable platform are not considered and it can cause undesired antagonistic behavior of drives [8].

To prevent the drives from these undesirable events, some model description of energy decoupling in the robot is necessary. However, when some model is considered, then the use of some global (centralized) model-based control strategy is more effective [8]. Nevertheless, the local control can be useful as a fast sub-drive control, of which desired values are generated on higher level by some model-based controller, which provides optimal and safe energy distribution.

One such combination "multi-level (hierarchical) control" will be investigated here (Fig. 3). It can be clearly specified as a high-level model-based control providing positional and speed loops with addition of fast low-level current-loop control. The high-level model-based control corresponds to the dynamics of mechanical part of the robots and on the other hand, low-level control corresponds to the dynamics of the drives (electrical part). This part is faster than dynamics of mechanical part and therefore is separated from high-level control. This concept can offer more possibilities to manage the control process than usual use of cascade control.

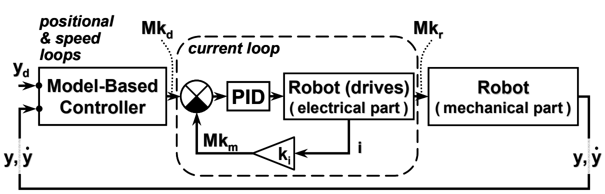

Fig. 3. Multi-level (hierarchical) control circuit.
(Torques: Mk_d - desired, Mk_m - measured, Mk_r - real)

As a suitable representative of model-based control, predictive control is considered. It will be introduced at the beginning parts of the paper. Firstly, a composition of the robot model will be introduced. Then the main points of design of predictive control will be explained.

Another issue connected to the control is a correct specification of the robot motion. Let us call this issue as trajectory planning. It represents time-parameterization of desired geometrical path (e.g. contour of the workpiece). In general, there are two different concepts. The first concept considers only pure geometrical features (curve-based planning) without relation to the real robot. The second concept conversely takes into account the dynamical features of the real robot and initial and final path points (point-to-point planning). Trajectory planning will be summarized in the second part of the paper.

2. Model Composition

To use model-based control, it is necessary to have some suitable model of the controlled system, in our case, model of the robot. In general, parallel robots represent multi-body systems, which can be straightforwardly described in physical coordinates by Lagrange's equations of mixed type [1]. These equations lead to the system of differential algebraic equations – DAE (1)

$$\begin{aligned} \mathbf{M}\,\ddot{\mathbf{s}} - \mathbf{\Phi}_s^T \boldsymbol{\lambda} &= \mathbf{g} + \mathbf{T}\,\mathbf{u} \\ \mathbf{f}(\mathbf{s}) &= \mathbf{0} \end{aligned} \tag{1}$$

where $\mathbf{M}$ is a mass matrix, $\mathbf{s}$ is a vector of physical coordinates (their number is usually greater than number of degrees of freedom – DOF), $\mathbf{\Phi}_s$ is a Jacobian, $\boldsymbol{\lambda}$ is a vector of Lagrange's multipliers, $\mathbf{g}$ is a vector of other internal relations, matrix $\mathbf{T}$ connects inputs $\mathbf{u}$ to appropriate differential equations and algebraic equations $\mathbf{f(s) = 0}$ represent geometrical constraints.

As mentioned, the model (1) is a DAE system, which is moreover nonlinear. It is not suitable for control design. However, it can be transformed to different form [3], to the system of ordinary differential equations in independent coordinates $\mathbf{y}$, which correspond to DOF:

$$\ddot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \dot{\mathbf{y}}) + \mathbf{g}(\mathbf{y})\mathbf{u} \tag{2}$$

$\mathbf{f}(\mathbf{y}, \dot{\mathbf{y}})$ represents robot dynamics; $\mathbf{g}(\mathbf{y})$ is input matrix. The model (2) can be rewritten in the state-space formula:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\,\mathbf{u} \\ \mathbf{y} &= \mathbf{H}\,\mathbf{x} \end{aligned}\,, \quad \mathbf{x} = [\mathbf{y}, \dot{\mathbf{y}}]^T \tag{3}$$

which is simpler and more transparent for handling with multi-input multi-output systems as robots usually are. Due to discrete realization of the control, the model (3) is discretized. To use standard discretization via expansion of exponential functions, the nonlinear vector $\mathbf{f}(\mathbf{x})$ in (3) has to be linearized. It can be provided by decomposition according to [9] leading to the linear form:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{F}(\mathbf{x})\,\mathbf{x} + \mathbf{G}(\mathbf{x})\,\mathbf{u} \\ \mathbf{y} &= \;\;\mathbf{H}\;\;\mathbf{x} \end{aligned}\,, \quad \mathbf{x} = [\mathbf{y}, \dot{\mathbf{y}}]^T \tag{4}$$

The obtained form (4) represents the robot dynamics identically as model (2) or (3); the individual elements of state and input matrices $\mathbf{F}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ has to be recomputed on-line for appropriate topical robot state $\mathbf{x}$. Output matrix $\mathbf{H}$ is rectangular identity matrix (it is equal output matrix $\mathbf{C}$ in (5)). The used decomposition is described in [9] and its real use is shown in [4] or [8]. Then, after discretization of (4), the obtained model has following form:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(k)\,\mathbf{x}(k) + \mathbf{B}(k)\,\mathbf{u}(k) \\ \mathbf{y}(k) &= \;\;\mathbf{C}\;\;\mathbf{x}(k) \end{aligned}\,, \quad \mathbf{x}(k) = [\mathbf{y}(k), \dot{\mathbf{y}}(k)]^T \tag{5}$$

That form is convenient for predictive control design.

3. Predictive Control

Predictive Control is a multi-step control, which is based on equations of predictions and the local minimization of quadratic criterion [2]. The equations of predictions involve the model and serve for prediction of future robot behavior.

The behavior, i.e. future robot outputs are substituted just by the predictions obtained from topical system state and application of the model. Control actions represent unknown parameters, which are computed.

The predictions are compared with future desired values (desired robot motion) in quadratic criterion. By its minimization, the unknown control actions are determined. The next two subsections deal with these principles.

3.1 Equations of Predictions

Equations of predictions serve for the expression of feedforward within horizon of predictions N. On their basis, dominant part of control actions is determined. Using discrete state-space form (5), the equations have following form:

$$\begin{aligned}
\hat{\mathbf{x}}_{(k+1)} &= \mathbf{A}\,\mathbf{x}_{(k)} + \mathbf{B}\mathbf{u}_{(k)} \\
\hat{\mathbf{y}}_{(k+1)} &= \mathbf{C}\mathbf{A}\,\mathbf{x}_{(k)} + \mathbf{C}\mathbf{B}\mathbf{u}_{(k)} \\
&\vdots \\
\hat{\mathbf{x}}_{(k+N)} &= \mathbf{A}^{N}\mathbf{x}_{(k)} + \mathbf{A}^{N-1}\mathbf{B}\mathbf{u}_{(k)} + \cdots + \mathbf{B}\mathbf{u}_{(k+N-1)} \\
\hat{\mathbf{y}}_{(k+N)} &= \mathbf{C}\mathbf{A}^{N}\mathbf{x}_{(k)} + \mathbf{C}\mathbf{A}^{N-1}\mathbf{B}\mathbf{u}_{(k)} + \cdots + \mathbf{C}\mathbf{B}\mathbf{u}_{(k+N-1)}
\end{aligned} \tag{6}$$

It can be rewritten in more condensed matrix notation:

$$\hat{\mathbf{y}} = \mathbf{f} + \mathbf{G}\,\mathbf{u} \tag{7}$$

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

Vector $\mathbf{f}$ represents free responds from time instant k, i.e. for $\mathbf{u} = \mathbf{0}$. The product $\mathbf{G}\,\mathbf{u}$ compensates differences of the responds from desired values within the horizon N.

3.2 Computation of Control Actions

The control actions are determined on the basis of minimization of quadratic criterion

$$J_k = \sum_{j=1}^{N} \left\{ \left\| (\hat{\mathbf{y}}_{(k+j)} - \mathbf{w}_{(k+j)})\mathbf{Q}_y \right\|^2 + \left\| \mathbf{u}_{(k+j-1)}\mathbf{Q}_u \right\|^2 \right\} \tag{8}$$

where N is a horizon of predictions; $\mathbf{Q}_y$ and $\mathbf{Q}_u$ are penalizations; and $\mathbf{w}_{(k+j)}$ are desired values [5].

That criterion is suitable condensed in matrix notation

$$J_k = [(\hat{\mathbf{y}} - \mathbf{w})^T \ \mathbf{u}^T] \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix}^T \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ \mathbf{u} \end{bmatrix} \tag{9}$$

from which, only one part (square-root) is sufficient to be minimized.

$$\mathbf{J} = \begin{bmatrix} \mathbf{Q_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{Q_y}\,\mathbf{G} \\ \mathbf{Q_u} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{Q_y}(\mathbf{w} - \bar{\mathbf{f}}) \\ \mathbf{0} \end{bmatrix} \tag{10}$$

The minimization leads to the solution of algebraic equations [6] for unknown control actions

$$\begin{aligned}
\begin{bmatrix} \mathbf{Q_y}\,\mathbf{G} \\ \mathbf{Q_u} \end{bmatrix} \mathbf{u} &- \begin{bmatrix} \mathbf{Q_y}\,(\mathbf{w} - \bar{\mathbf{f}}) \\ \mathbf{0} \end{bmatrix} = \mathbf{0} \\
\mathbf{A} \quad\ \mathbf{u} &- \qquad\quad \mathbf{b} \qquad\quad = \mathbf{0}
\end{aligned} \tag{11}$$

Obtained vector $\mathbf{u}$ represents control actions for whole horizon N. However, only first appropriate actions are really applied to the robot. The process of minimization is repeated in every time step for appropriately updated model (5).

4. Local Current Control Loop

In this section, the assumptions and expected properties of addition of local current control loop to model-based control will be introduced. (The model-based control represents positional and speed loop together.)

Properties of the current loop depend on used drive (motor). Usual motors in robotic applications are brushes DC (direct current) motors, brushless EC (electronically commuted) motors (sometimes called brushless DC motors), and synchronous brushless AC (alternating current) motors [10]. From mathematical-physical analysis viewpoint, they have the same description, only the number of activated coils has to be taken into account. For simplicity, let DC motor is considered. Such motor, having permanent magnets in stator, is described by ordinary differential equation (12) of second order:

$$\ddot{M}_k + \frac{R}{L}\dot{M}_k + \frac{k_{m_1}k_{m_2}}{JL}M_k = \frac{k_{m_1}}{L}\dot{u} \tag{12}$$

where k_{m_1} and k_{m_2} are torque and speed constants, R, L is resistance and inductance and J is inertia moment of rotor. Equation (12) corresponds to the scheme in Fig. 4.

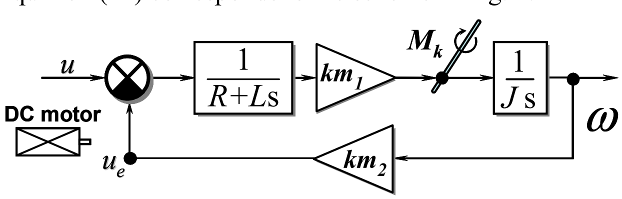

Fig. 4. Block scheme of brushes DC motor with permanent magnet in the stator.

The motors in robotics applications have relatively high gear ratio. Thus, in spite of 'small' speed of robot input, the motor revolutions are not small and moreover, they change their sense very often.

This situation increases undesired influence of internal induced voltage (negative feedback in Fig. 4), which acts against current, and thereby decreases the motor torque. This undesired property can be reduced just by addition of current loop (Fig. 5).

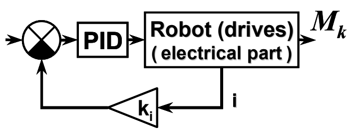

Fig. 5. The local current loop.

The current is usually measured by Hall sensor. The difference of desired value of current (torque from predictive controller) and real value of current is transformed by current loop to desired voltage on the motor, which is realized by pulse-width modulation (PWM). This modification of model based control improves the dynamics of the drives and enables robot to use states near its limits leading to effective distribution energy and increase of machine tool productivity.

5. Concepts of Trajectory Planning

Trajectory planning is one of inherent preparative operations before starting real control process of robot motion. Its objective is to generate the reference inputs $\mathbf{w}$ describing desired motion. Real trajectory is usually given by a number of different parameters: technological (e.g. suitable machining velocities, motion orientation etc.) and constructional (lengths, radiuses, shapes etc.).

The planning can be considered either from kinematical point of view, where paths of the motion are known or from dynamical point of view where real paths are unknown and only start and end points and bounds are given for used robot.

5.1 Curve-Based Planning

Curve-based planning arises from elementary laws of kinematics. During planning process, the following characteristics have to be successively determined

- length of path and possible range of rotation
- time for reaching the end point of path
- geometrical time-depended parameter
- real trajectory coordinates and time derivatives

The determination of mentioned characteristics will be briefly described step by step.

The path length ℓ and rotation range ψ can be generally determined as follows

$$\ell = \int_s ds\,, \qquad \psi = \psi_{final} - \psi_{inital} \tag{13}$$

In cases, when the length cannot be determined analytically, then it can be determined approximately as a sum of lengths of small abscissa segments, which substitute considered path. Appropriate time can be determined from path length and e.g. known velocities v, ω and accelerations a, α in start and end points via simple kinematics' laws by expressions

$$a = \frac{dv}{dt} \rightarrow t_1 = \frac{2\,\ell}{v_{inital} + v_{final}}$$
$$\alpha = \frac{d\omega}{dt} \rightarrow t_2 = \frac{2\,\psi}{\omega_{inital} + \omega_{final}} \tag{14}$$

In expressions, double integration is performed. The time, which has higher value, is chosen and labeled as t_{final}. Now, it is possible to determine the geometrical parameter $p(t)$ and $\rho(t)$, which represents one-dimensional (1D) time parameterization. Its computation arises from selection of polynomials of accelerations e.g. let us consider polynomials of 3rd order and appropriate initial and final conditions for $s, v, a, \psi, \omega, \alpha$

$$a(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$
$$\alpha(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 \tag{15}$$

By double integration of (15), the system of algebraic equations for unknown coefficients $a_i, \alpha_i,\ i = 0, 1, 2, 3$ is obtained. The coefficients are used for determination of geometrical parameter for positions and rotations

$$p = s(t),\ \ p \in \langle 0,\ \ell \rangle,\ \ \rho = \psi(t),\ \ \rho \in \langle 0,\ \psi_f \rangle \tag{16}$$

The parameter controls the real planning in 2D or 3D space. For its computation is suitable to select zero initial and final acceleration to reduce number of equations. The selection of 3rd order provides continuous and in segments smooth curves up to 2nd derivatives. If the polynomial for acceleration is 5th order, then the curves are fully continuous and smooth up to 2nd derivatives.

Now, it is possible to provide real planning of individual trajectory segments via parametric curves or via densely sampled general curves.

In case of general curves, the geometrical parameter serves as a selector of coordinates $\mathbf{w} = [x, y, z]$ from their appropriate table of given curve. This computation is only approximation. The parameterization accuracy depends on the selection of initial length of substitutive abscissa segments.

5.2 Point-to-Point Planning

Point-to-Point planning is another way of preparing the desired values. The planning is based on dynamical relations involved in the model of considered robot. The time-parameterized trajectories are generated during simulation of specific control task.

The planning can be defined this way: "Let us have two points (start and end point) and presumptive time T, in which the real robot should perform the motion between those points. A path is free of hard constraints; only end-point should be achieved" (Fig. 6).

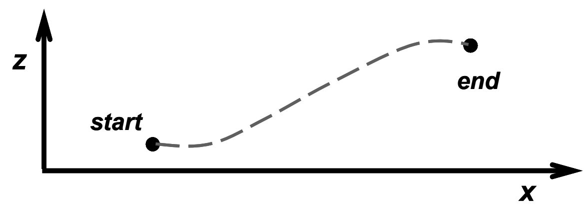

Fig. 6. Point-to-Point planning.

As a suitable way, predictive control can be used. It consists in the composition of the equations of predictions, which involve the dynamical robot model (5) and in minimization of quadratic criterion (8).

The criterion can include several adjustable parameters more:

$$J_k = \sum_{j=No+1}^{N} \left\| (\hat{\mathbf{y}}(k+j) - \mathbf{w}(k+j))\mathbf{Q}_{\mathbf{y}} \right\|^2 + \sum_{j=1}^{Nu} \left\| \mathbf{u}(k+j-1)\mathbf{Q}_{\mathbf{u}} \right\|^2 \quad (17)$$

i.e. the horizon No of initial criterion insensitivity and control horizon Nu; and also the desired values $\mathbf{w}$, which determine the transition from start to end point in the criterion. The values $\mathbf{w}$ considered here serve only for design trajectories, which represents real desired values used for real control.

In our case, the values $\mathbf{w}$ for planning correspond to end state, respectively, to the end position i.e.

$$\mathbf{w} = \left[\mathbf{w}(k+j), \cdots \mathbf{w}(k+N)\right]\Big|_{\mathbf{w}(k+j) = const., \; j = No+1, \cdots N} \quad (18)$$

To distribute the input energy in whole time interval determined by time T, i.e. to design the trajectory with suitably distributed energy, the parameters in the criterion are used.

If we set $N = Nmax = T/Ts$ (Ts – suitable selected sampling) and $No = N - k$, where k is order of the model, then the quadratic criterion will consider only last k differences among predicted end-point and its reference value. The horizon No representing initial insensitivity determines number of free outputs, i.e. outputs without penalization, which enable the algorithm to shift the reaching the end point at later time (step), at time T, with appropriate distribution of input energy. Thus, the first term in the criterion includes only last k appropriate differences $(\hat{\mathbf{y}}(k+j) - \mathbf{w}(k+j))$.

Relating to the control horizon Nu, it fulfils $Nu \le N$. $Nu = N$ is its usual selection, which enables the system to reach the end point, but it does not provide the stabilization of control actions in it. When $Nu < N$ together with equality condition that the last $N - Nu$ control actions are designed to be equal, then the predictive control provides, except stabilization of the system in desired end point, also the stabilization of control actions in this point (motion).

In case of robots (nonlinear systems), during the trajectory design, the model parameters have to be changed according to current state with simultaneous progressively shortened horizon

$$N := Nmax, Nmax - 1, \cdots, Nmin + 1, Nmin, \; Nmin > k \quad (19)$$

The shortening provides that the time limit T is not overrun. The algorithm provides uniform distribution of the input energy in specified time T.

6. Simulative examples

In this section, the simulative experiments will be shown on one planar parallel robot, which is prepared for top milling machine. Here, the desired trajectories designed according to previous section are used.

6.1 Description of Considered Parallel Robot

For simulative tests, robot 'Moving Slide' was used. It is illustrated in Fig. 7. This robot represents horizontal planar parallel configuration, which has 4 × Rotational + Prismatic + Rotational joints. Moreover, it is redundantly actuated, because there are four drives for only three degreases of freedom. This feature furthermore improves robot stiffness.

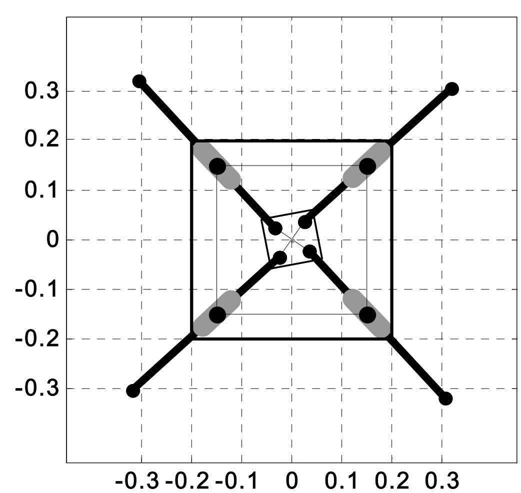

Fig. 7. 4RPR parallel mechanism 'Moving Slide'.

6.2 Desired Trajectory

The desired trajectory (Fig. 8) consists of abscissa and arc segments. Individual segments were planed separately according to subsection 4.1 'Curve-Based Planning'. The motion sense is indicated by numbered arrows. Initial and final points have zero x and y coordinates.

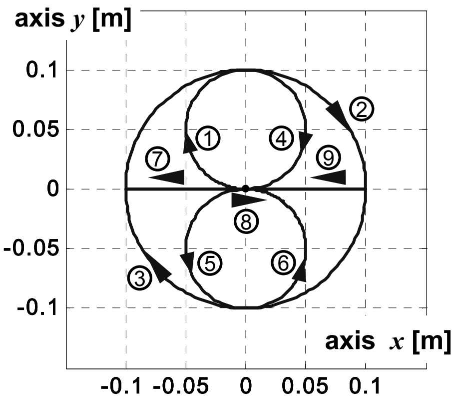

Fig. 8. Desired trajectory.

6.3 Simulative Results

The aim of the experiments was achievement of desired trajectory (desired motion along desired trajectory) using described model-based predictive control. The results relate to the trajectory described above. The following figures show the time histories of four appropriate input torques, which are required from drivers.

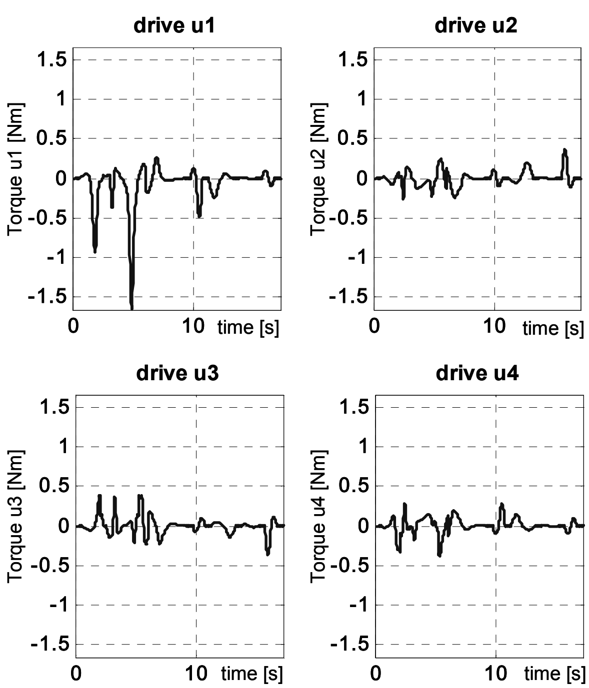

Fig. 9. Time histories of torques for trajectory from Fig. 8.

7. Conclusion

In the paper, the concepts of model-based control and trajectory planning intended for industrial parallel robots were investigated. The multi-level (hierarchical) control (model-based predictive control added by fast current-loop control) was explained. Finally, the paper outlined two different concepts of trajectory planning, at first, based on pure geometrical features and at second, based on inclusion of dynamical features of the real robot.

Acknowledgements

The authors appreciate kind support of Grant Agency of the Czech Republic by the grant (102/06/P275, 2006/08) 'Model-based Control of Mechatronic Systems for Robotics' and grant (102/05/0271, 2005/07) 'Methods of Predictive Control, Algorithms and Implementation'.

References

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

[2] A. Ordys & D. Clarke, A State - Space Description for GPC Controllers. *Int. J. Systems SCI.*, Vol. 24, No. 9, 1993, 1727-1744.

[3] V. Stejskal & M. Valášek, *Kinematics and Dynamics of Machinery* (Marcel Dekker, New York, 1996).

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

[5] K. Belda et al., KBweb, http://as.utia.cz/asc/, © 2005.

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

[7] R. Neugebauer, *Development Methods and Application Experience of Parallel Kinematics* (IWU Chemnitz, Reports from IWU, **Vol. 16**, 2002)

[8] K. Belda, J. Böhm & M. Valášek, Predictive Control Applied to Planar Parallel Robots. *Preprints of the 16th World Congress of the International Federation of Automatic Control*. 16th IFAC World Congress, Prague 2005, 6pp.

[9] M. Valášek & P. Steinbauer, Nonlinear Control of Multibody Systems, *Euromech 1999*, Lisabon, 437-444.

[10] P. Souček, *Servomechanisms in Machine Tools* (CTU in Prague, 2004; in Czech)