MODEL - BASED CONTROL FOR PARALLEL ROBOT KINEMATICS

Květoslav Belda[1], Josef Böhm[1] and Michael Valášek[2]
[1]*Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Department of Adaptive Systems, Pod vodárenskou věží 4, 182 08 Prague 8 - Libeň;* [2]*Czech Technical University, Faculty of Mechanical Engineering, Department of Mechanics, Karlovo nám. 13, 121 35 Prague 2*

Abstract: The paper deals with the application of the model - based control approach within robotic system that includes the parallel structure and adequate drives. As an example, several prototypes of planar parallel redundant structures driven by MAXON DC motors are used. The suitable form of mathematical model, both mechanical and drive part, will be introduced and subsequently used for control design. The design is based on discrete predictive control algorithms, which can well compete and surpass the classical feedback control strategies.

Key words: over-actuated parallel kinematic structure; exact linearization; DC motors; predictive control algorithms.

1. INTRODUCTION

Nowadays, the further development in industrial area is constrained by deficit of powerful machines with adequate dynamics and stiffness[4]. Utilization of parallel robots controlled by suitable control algorithms seems to be promising way to improve dynamics, stiffness, accuracy and productivity of machine tools and their centers, which, together with assembly lines, form the backbone of mass production.

The parallel robots – parallel structures[4,5,8] can be simply understood as movable truss constructions or as movable work platforms supported by several parallel arms. The structures are driven by different drives, in almost cases by electromotors. Simple comparison of serial and parallel structures is shown in Fig. 1.

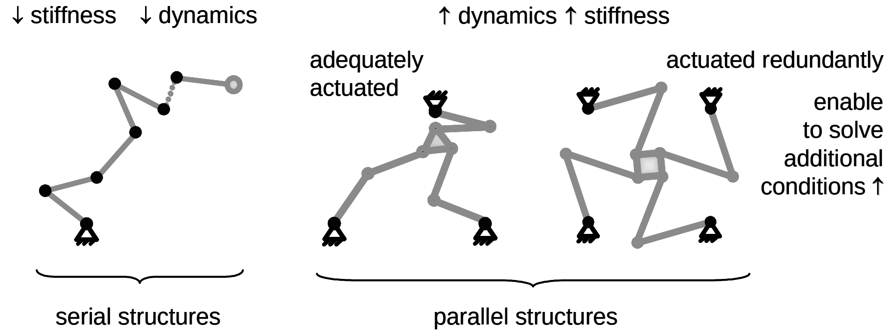

Figure 1. Comparison of serial and parallel structures.

The serial structure has the drives in each joint. The drives in parallel configuration are located in the fixed basic frame.

The fundamental task of the control of mechanical structures driven by electromotors is to design appropriate control actions, which accomplish required movement of the movable platform. In general, the scheme in Fig. 2 is solved.

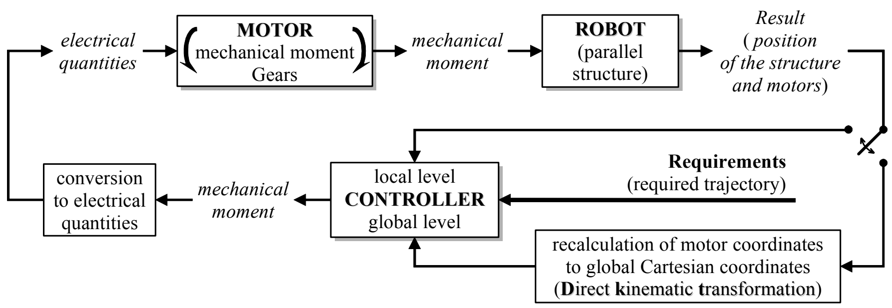

Figure 2. Conceptual scheme of control circuit for local (independent) and global (centralized) model-based level.

There are two levels of the control[5,9] shown in Fig. 2: local and global. Local (decentralized) level[8], represented by continuous PID or discrete PSD controllers, controls each drive independently, without any consideration of mutual relations. The global (centralized) level, represented by predictive controllers[3], uses for control the dynamic model of the robot structure. The model can include also drives, which represent also appreciable dynamics. The complete model is used not only for control design, but also for simulation of a real object - robot structure. The model can be written in different forms.

In comparison, the classical PID/PSD controllers design the control action only from feedback[8]. They work up only output information included in feedback regressively without any assumptions to future. On the other hand, the model-based approach combines feedforward and feedback together[3,7]. The model represents prior information (feedforward) modeling the control actions. Subsequently, the feedback from measurable robot outputs compensates model inaccuracies and certain bounded disturbance. Such design is more optimal and control action can react faster and more effective than PID/PSD structures.

In this paper, the standard model of the robot is defined[2]. Consecutively, its transformation (specific exact linearization) is derived. Then, the theoretical background of the control design follows. It includes new possibilities to utilize the properties of the parallel robots, properties of their mathematical models respectively. The designed predictive controllers are explained both for mechanical and global model. The global model includes additionally model of drives. In this paper, for simplicity DC motors are assumed.

2. MODEL OF ROBOTIC SYSTEM

2.1 Dynamical model of parallel robot structures

Mathematical model of dynamics is represented by equations of motion. They give integral information about relations of kinematic variables of separate bodies, external force effects (external forces, torques generated e.g. by drives), force effects in joints (reactions) and other internal force effects. Robot structure generally represents multibody system. We search for a method, which could provide simple and aimed composition of the equations of motion describing relatively complex parallel structure. For composition of equations of motion in robotics, the Lagrange's equations of second or mixed type, based on expression of kinetic and potential energy, are mostly used. Difference between the types of Lagrange's equations is given by a choice of so-called generalized coordinates, which, in the first case, must be independent. In the most of cases, the expression of energies in independent coordinates is too difficult, therefore let us assume the latter case – mixed type that is more general. According the theory described e.g. by Stejskal and Valášek[2] the model (DAE form) is given partly by structural relations

$$\mathbf{f}(\mathbf{s}(t)) = \mathbf{0}\,, \quad \left(f_k(\mathbf{s}(t) = 0;\ k = 1,\cdots,r\ \ (r = n - i)\right) \qquad (1)$$

and partly by a system of n second-order differential equations (equations of motion)

$$\mathbf{M}\ddot{\mathbf{s}} = \mathbf{g} + \mathbf{T}\mathbf{u} + \mathbf{\Phi}^T\boldsymbol{\lambda}\,, \quad (\mathbf{\Phi} : \phi_{k,j} = \frac{\partial f_k}{\partial s_j};\ \begin{matrix} k = 1,\cdots,r \\ j = 1,\cdots,n \end{matrix}) \tag{2}$$

where **s** are physical coordinates, n is their number, i is a number of degrees of freedom, **M** is a regular symmetric square matrix of an order n, **g** is an n dimensional vector, **T** is a rectangular matrix of type $n\,x\,m$, **u** is an m dimensional input vector, $\mathbf{\Phi}^T$ is a transpose of overall Jacobian of type $n\,x\,r$ and $\boldsymbol{\lambda}$ is a vector of Lagrange's multipliers with dimension $r = n - i$.

2.1.1 Composition of pure equations of motion

To design control more simply, we search for the most compact notation of system equations (1)-(2) with as small as possible number of unknown parameters. As mentioned, the direct derivation of equations of motion in independent coordinates leads to difficult expressions of energies. Therefore, we investigate, if the system (2) could be transformed just to independent coordinates[2]. This transformation would reduce the number of differential equations and mainly it would remove redundant Lagrange's multipliers, connected with structural forces and structural relations causing the design difficult. Thus, we look for such solution, which annuls element $\mathbf{\Phi}^T\boldsymbol{\lambda}$ and reduces physical coordinates (i.e. dependent and independent coordinates) to independent ones. The solution can be a null space of the whole Jacobian $\mathbf{\Phi}$, which is simultaneously interpreted as a Jacobian matrix **R**, fitting the following equality, arising from properties of the null space

$$\mathbf{\Phi}_s\mathbf{R} = \mathbf{R}^T\mathbf{\Phi}_s^T = \mathbf{0} \tag{3}$$

and arising also from recomputation of independent to physical coordinates

$$\mathbf{s} = \mathbf{s}(\mathbf{q}) = \mathbf{s}(\mathbf{x}) \tag{4}$$

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{q}}\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{x}} \tag{5}$$

$$\ddot{\mathbf{s}} = \dot{\mathbf{R}}\dot{\mathbf{x}} + \mathbf{R}\ddot{\mathbf{x}} \tag{6}$$

Now, if we insert Eq. (6) for Eq. (2) and perform its multiplication by transposition of Jacobian matrix $\mathbf{R}^T$, we obtain resultant compact form

$$\mathbf{R}^T\mathbf{M}\mathbf{R}\ddot{\mathbf{x}} + \mathbf{R}^T\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{x}} = \mathbf{R}^T\mathbf{g} + \mathbf{R}^T\mathbf{T}\mathbf{u} \tag{7}$$

Let us analyze obtained Eq. (7). If the system does not include gravitational forces and we assume zero inputs **u**, then all terms are also zero. On the other hand, if the system contains gravitational forces included in a vector **g**, then the robotic structure can change its own position in spite of zero inputs **u**. In such a case, the Eq. (7) can be written in alternative form Eq. (9) arising from following Eq. (8).

$$\mathbf{R}^T\mathbf{M}\mathbf{R}\ddot{\mathbf{x}} = -\mathbf{R}^T\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{x}} + \mathbf{R}^T(\mathbf{g}_0 + \mathbf{g}_r) + \mathbf{R}^T\mathbf{T}\mathbf{u} \tag{8}$$

where $\mathbf{g}_0$ is a vector that is dependent only on velocities and $\mathbf{g}_r$ is a vector containing only gravitational forces;

$$\mathbf{R}^T\mathbf{M}\mathbf{R}\ddot{\mathbf{x}} = -\mathbf{R}^T\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{x}} + \mathbf{R}^T\mathbf{g}_0 + \mathbf{R}^T\mathbf{g}_r + \mathbf{R}^T\mathbf{T}\mathbf{u} \tag{9}$$

which can be transformed to a form with isolated second derivatives

$$\begin{aligned} \ddot{\mathbf{x}} &= (\mathbf{R}^T\mathbf{M}\mathbf{R})^{-1}\mathbf{R}^T(\mathbf{g}_0 - \mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{x}}) + (\mathbf{R}^T\mathbf{M}\mathbf{R})^{-1}\mathbf{F} \\ \ddot{\mathbf{x}} &= \qquad\qquad \mathsf{f}(\mathbf{x},\dot{\mathbf{x}}) \qquad\qquad + \quad \mathsf{g}(\mathbf{x}) \quad \mathbf{F} \end{aligned} \tag{10}$$

where vector **F** represents new inputs - generalized force effects extended by generalized gravitational forces $\mathbf{R}^T\mathbf{g}_r$; i.e.

$$\mathbf{F} = \mathbf{R}^T\mathbf{g}_r + \mathbf{R}^T\mathbf{T}\mathbf{u} \tag{11}$$

Then the Eq. (10) can be simply normalized into state formulation Eq. (12)

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}) + \mathbf{B}_C(\mathbf{X})\,\mathbf{F} \tag{12}$$

This modification guarantees, that $[\dot{\mathbf{x}}, \mathsf{f}(\mathbf{x},\dot{\mathbf{x}})]^T = \mathbf{f}(\mathbf{X}) = \mathbf{0}$ for arbitrary **x** from range of definition and zero time derivatives $\dot{\mathbf{x}} = \mathbf{0}$, in spite of presence of gravities that are added to inputs. Thus, final control actions follow from Eq.(13). This property we utilize in the next section that describes exact linearization.

2.1.2 Algorithm of exact linearization

Since almost all control approaches are based on the model in linear form, possibly with time-varying elements, thus a linearization of nonlinear state function is required. The following derivation introduces one simple specific algorithm of a linearization, which gives exact solution

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}) = \mathbf{A}_C(\mathbf{X})\,\mathbf{X} \tag{13}$$

For solution, let us assume that the nonlinear function **f**(**X**) and point **X** be given, **X** belongs to range of definition of the function and in the case of zero elements in **X**, possibility of their substitution by suitable nonzero number $\kappa \to 0$, which prevents zero division. Furthermore, let us assume two types of state variables: generally outputs **x** and their time derivatives $\dot{\mathbf{x}}$; in detail

$$\mathbf{X} = [\mathbf{x}, \dot{\mathbf{x}}]^T = [\mathbf{x}_1, \mathbf{x}_2]^T = [\mathbf{x}_1 = [x_{11}, x_{12}, x_{13}],\ \mathbf{x}_2 = [x_{21}, x_{22}, x_{23}]]^T \tag{14}$$

Finally, let us repeat the assumption from previous section that

$$\mathbf{f}(\mathbf{X}) = \mathbf{0}\,\Big|_{\mathbf{X}=\mathbf{X}_r=[\mathbf{x}_{r1}\text{arbitrary},\ \mathbf{x}_{r2}=\mathbf{0}]} \tag{15}$$

Then we can obtain the decomposition indicated in Eq. (13). It starts from second group of variables $\mathbf{x}_2$ (i.e. according to Eq. (14): order is $\{x_{21}, x_{22}, x_{23}, x_{11}, x_{12}, x_{13}\}$ i.e. {4, 5, 6, 1, 2, 3}). The order is given by amount of the information included in the function **f**(**X**). Its first elements are only copies of $\mathbf{x}_2$ but its second elements represent its own nonlinear relation expressed by $\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{g}(\mathbf{x})\mathbf{F}|_{\mathbf{F}=\mathbf{0}}$. In view of these assumptions the exact linearization – decomposition can be based on differences[10]

$$\begin{aligned}\mathbf{f}(\mathbf{X}) = {} & \frac{\Delta\mathbf{f}(\circ)}{.\Delta x_{11}}\Delta x_{11} + \frac{\Delta\mathbf{f}(\circ)}{.\Delta x_{12}}\Delta x_{12} + \frac{\Delta\mathbf{f}(\circ)}{.\Delta x_{13}}\Delta x_{13} + \\ & + \frac{\Delta\mathbf{f}(\circ)}{.(x_{21}-0)}(x_{21}-0) + \frac{\Delta\mathbf{f}(\circ)}{.(x_{22}-0)}(x_{22}-0) + \\ & + \frac{\Delta\mathbf{f}(\circ)}{.(x_{23}-0)}(x_{23}-0)\end{aligned} \tag{16}$$

(Note: The dots before state variables in denominators mark division ‘element by element’; division of all elements of differences by scalar Δx_{ij°)

i.e. $\mathbf{f}(\mathbf{X}) =$

$$
\begin{aligned}
&= \frac{\mathbf{f}([x_{11}, x_{12}, x_{13}, 0,0,0]^T) - \mathbf{f}([x_{r11}, x_{12}, x_{13}, 0,0,0]^T)}{.(x_{11} - x_{r11})} (x_{11} - x_{r11}) + \cdot\cdot \\
&+ \frac{\mathbf{f}([x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}]^T) - \mathbf{f}([x_{11}, x_{12}, x_{13}, 0, x_{22}, x_{23}]^T)}{.x_{21}} x_{21} + \cdot\cdot
\end{aligned}
\tag{17}
$$

Let us analyze Eq. (17), the individual fractions are columns of matrix $\mathbf{A}_C(\mathbf{X})$ and internal structure of matrix $\mathbf{A}_C(\mathbf{X})$ (in our example, of sixth order) is following

$$
\mathbf{A}_C(\mathbf{X}) = \left[\begin{array}{ccc|ccc}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 0 & 0 & a_{44} & a_{45} & a_{46} \\
0 & 0 & 0 & a_{54} & a_{55} & a_{56} \\
0 & 0 & 0 & a_{64} & a_{65} & a_{66}
\end{array}\right]
\tag{18}
$$

the first three columns contain only zeros because are composed of differences being also zeros - the vector function equals zeros for zero time derivatives; see Eq. (19) - numerator of the first column of $\mathbf{A}_C(\mathbf{X})$.

$$
\mathbf{f}([x_{11}, x_{12}, x_{13}, 0,0,0]^T) - \mathbf{f}([x_{r11}, x_{12}, x_{13}, 0,0,0]^T) = \mathbf{0} \tag{19}
$$

Now, we can define whole linearized continuous state formulation

$$
\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}) + \mathbf{B}_C(\mathbf{X})\,\mathbf{F} \;\rightarrow\; \dot{\mathbf{X}} = \mathbf{A}_C(\mathbf{X})\mathbf{X} + \mathbf{B}_C(\mathbf{X})\,\mathbf{F} \tag{20}
$$

The formulation Eq. (20) can be already discretized by standard discretization[9] to a form

$$
\begin{aligned}
\mathbf{X}(k+1) &= \mathbf{A}\mathbf{X}(k) + \mathbf{B}\,\mathbf{F}(k) \\
\mathbf{y}(k) &= \mathbf{C}\mathbf{X}(k)
\end{aligned}
\tag{21}
$$

Such model Eq. (21) is prepared for control design in discrete domain.

2.2 Dynamical model of drives

Last but not a little significant part of model of robotic system is a model of its drives. As mentioned as beginning, we consider MAXON brushes DC motors. Such motors, having permanent magnets in stator, are described by ordinary differential equation of second order[13]:

$$\ddot{M}_k + \frac{R}{L}\dot{M}_k + \frac{k_{m_1} k_{m_2}}{JL} M_k = \frac{k_{m_1}}{L}\dot{u} \qquad (22)$$

where k_{m_1} and k_{m_2} are torque and speed constants, R and L is terminal resistance and inductance and J is rotor inertia. Equation (39) corresponds to the scheme in Fig. 3.

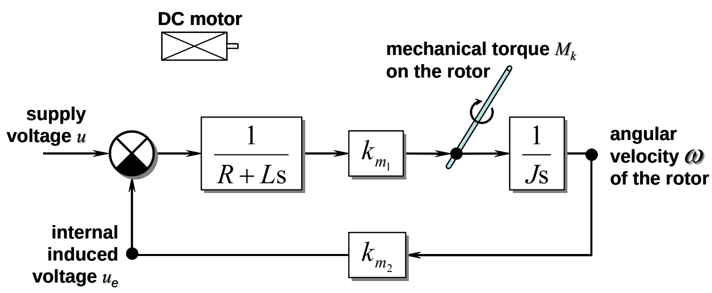

Figure 3. Block scheme of brushes DC motor with permanent magnet in the stator.

In equilibrium case ($\omega = 0$), the order of Eq. (22) is reduced to first order:

$$L\dot{M}_k + RM_k = k_{m_1} u \qquad (23)$$

transformed to normal form, more suitable for model of robotic system

$$\dot{M}_k = -\frac{R}{L} M_k + \frac{k_{m_1}}{L} u_r \qquad (24)$$

In transient process, the necessary voltage u is not a function only of load torque $u_r(M_k)$ Eq. (24) but also function of angular velocity $u_e(\omega)$ of the rotor

$$u = u_r(M_k) + u_e(\omega) = u_r(M_k) + k_{m_2}\dot{\varphi} \qquad (25)$$

where $u_e(\omega)$ represents compensation of internal induced voltage.

3. CONTROL DESIGN

The simplest control approach considered means taking the robots and manipulators, powered by group of independent drives - actuators, separately controlled, as a set of single input - single output systems (setSISO) by simple PID/PSD feedback controllers[8]. Mutual interactions among all drives, caused by different positions during the robot movement, are included as disturbances entering each “single” system constituting the robot.

Mentioned solution, in case of parallel robots has serious problem of mutual conflict of drives. It is indicated by unpredictable increase of integral/sum (I/S) channels in a controller. The solution is described in dissertation[9].

Different approach is the use of model of robotic system and on that base to design suitable control. Such approach generates more effective control actions, because, with knowledge of the mathematical description – model, can estimate possible future behavior. Generally, such approach is called model-based design of control. The next section deals with its one example.

3.1 Generalized Predictive Control (GPC)

Generalized Predictive Control[3,6,9] combines feedback ~ feedforward design as well as similar approach – Linear Quadratic Control (LQC)[7]. It is multi-step control, based on local optimization of quadratic criterion

$$J = \sum_{i=t+1}^{t+N} (y(i)^T Q_y \; y(i) + u(i-1)^T Q_u \; u(i-1)) \qquad (26)$$

where N is a horizon of optimization, Q_y and Q_u are output and input penalizations (here normalized as $Q_y = 1$ and $Q_u = \lambda$) and $y(i)$ and $u(i\text{-}1)$ are outputs and inputs respectively.

3.1.1 Expression of prediction of future outputs

The base of GPC control is a *prediction* (computation, estimation) of future values of outputs, used in the criterion. This section introduces two ways how to predict future outputs, which use the knowledge of mathematical model of the complete robotic system – following summary equations Eqs. (27) and (28)

$$\dot{\mathbf{X}} = \mathbf{A}_C(\mathbf{X})\mathbf{X} + \mathbf{B}_C(\mathbf{X})\,\mathbf{F} \;\rightarrow\; \dot{\mathbf{X}} = \mathbf{A}_C(\mathbf{X})\mathbf{X} + \mathbf{B}_C(\mathbf{X})\mathbf{R}^T\mathbf{T}\,\mathbf{M}_k \quad (27)$$

$$\dot{\mathbf{M}}_k = -\frac{R}{L}\mathbf{M}_k + \frac{k_{m_1}}{L}\mathbf{u}_r \tag{28}$$

Eqs. (27)-(28) can be covered in one state formulation Eq. (29) assuming $a_0 = -R/L$ and $b_0 = k_{m1}/L$

$$\left.\begin{aligned}\dot{\mathbf{X}} &= \mathbf{A}_C(\mathbf{X}, a_0)\,\mathbf{X} + \mathbf{B}_C(b_0)\,\mathbf{u}_r \\ \mathbf{y} &= \mathbf{C}\,\mathbf{X}\end{aligned}\right| \;\mathbf{A}_C(\mathbf{X}, a_0) = \begin{bmatrix}\mathbf{A}_C(\mathbf{X}) & \mathbf{B}_C(\mathbf{X}) \\ \mathbf{0} & diag(a_0)\end{bmatrix}, \mathbf{B}_C(b_0) = \begin{bmatrix}\mathbf{0} \\ b_0 \\ b_0 \\ b_0 \\ b_0\end{bmatrix} \tag{29}$$

C is a unitary matrix of type that is given by number of outputs and dimension of state vector, which in case of considering model of drives is extended for new internal states $\mathbf{M}_k$

$$\mathbf{X} = [\mathbf{x}, \dot{\mathbf{x}}, \mathbf{M}_k]^T = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]^T \tag{30}$$

As a final representative model for model-based approach, let us assume discrete version of Eq. (29)

$$\begin{aligned}\mathbf{X}(k+1) &= \mathbf{A}\,\mathbf{X}(k) + \mathbf{B}\,\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\,\mathbf{X}(k)\end{aligned} \tag{31}$$

representing the both cases: either formulation Eq. (21) or discrete version of formulation Eq. (29). The former case considers inputs **u** = **F** as values of required generalized force effects, finally recomputed to input torques and latter case considers inputs **u** = $\mathbf{u}_r$ as values of supply voltage.

The most usual, *absolute algorithm* of prediction arises directly just from state formulation Eq. (31)

$$\begin{aligned}\hat{\mathbf{X}}(k+1) &= \mathbf{A}\;\mathbf{X}(k) + \mathbf{B}\mathbf{u}(k) \\ \hat{\mathbf{y}}(k+1) &= \mathbf{C}\mathbf{A}\;\mathbf{X}(k) + \mathbf{C}\,\mathbf{B}\mathbf{u}(k) \\ &\;\;\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad \ddots \\ \hat{\mathbf{X}}(k+N) &= \mathbf{A}^N\,\mathbf{X}(k) + \mathbf{A}^{N-1}\,\mathbf{B}\mathbf{u}(k) + \cdots\; \mathbf{B}\mathbf{u}(k+N-1) \\ \hat{\mathbf{y}}(k+N) &= \mathbf{C}\mathbf{A}^N\,\mathbf{X}(k) + \mathbf{C}\mathbf{A}^{N-1}\,\mathbf{B}\mathbf{u}(k) + \cdots \mathbf{C}\,\mathbf{B}\mathbf{u}(k+N-1)\end{aligned} \tag{32}$$

thus the prediction written in matrix notation is

$$\hat{\mathbf{y}} = \mathbf{f} + \mathbf{G}\mathbf{u}\,, \quad \mathbf{f} = \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^N \end{bmatrix} \mathbf{X}(k), \quad \mathbf{G} = \begin{bmatrix} \mathbf{C} & \mathbf{B}\cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{C}\,\mathbf{A}^{N-1} & \mathbf{B}\cdots & \mathbf{CB} \end{bmatrix} \tag{33}$$

The alternative, *incremental algorithm* arises from modification of Eq. (31)

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{A}\mathbf{X}(k) + \mathbf{B}\mathbf{u}(k-1) + \mathbf{B}\Delta\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{X}(k) \\ \mathbf{u}(k+1) &= \mathbf{u}(k-1) + \Delta\mathbf{u}(k) \end{aligned} \tag{34}$$

$$\begin{bmatrix} \hat{\mathbf{y}}(k+1) \\ \hat{\mathbf{y}}(k+2) \\ \vdots \\ \hat{\mathbf{y}}(k+N) \end{bmatrix} = \begin{bmatrix} \mathbf{C}\,\mathbf{A} \\ \mathbf{C}\,\mathbf{A}^2 \\ \\ \mathbf{C}\,\mathbf{A}^N \end{bmatrix} \mathbf{X}(k) + \begin{bmatrix} \mathbf{CB} \\ \mathbf{C}(\mathbf{A}+\mathbf{I})\mathbf{B} \\ \vdots \\ \mathbf{C}(\mathbf{A}^{N-1}+\cdots+\ \mathbf{A}\ +\mathbf{I})\mathbf{B} \end{bmatrix} \mathbf{u}(k-1) + $$
$$+ \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \cdots & \mathbf{0} \\ (\mathbf{A}+\mathbf{I})\mathbf{B} & \mathbf{CB} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{C}(\mathbf{A}^{N-1}+\cdots+\mathbf{A}+\mathbf{I})\mathbf{B} & \mathbf{C}(\mathbf{A}^{N-2}+\cdots+\mathbf{A}+\mathbf{I})\mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \\ \Delta\mathbf{u}(k+N-1) \end{bmatrix} \tag{35}$$

the prediction of $\hat{\mathbf{y}}$ based on incremental algorithm, written in matrix notation, is

$$\begin{aligned} \hat{\mathbf{y}} \quad = \quad \mathbf{f} \quad + \quad \mathbf{f}_{\mathbf{f}} \quad + \\ + \quad\quad \mathbf{G} \quad \Delta\mathbf{u} \end{aligned} \quad \Bigg| \begin{aligned} &\hat{\mathbf{y}} = [\hat{\mathbf{y}}(k+1), \hat{\mathbf{y}}(k+2), \cdots, \hat{\mathbf{y}}(k+N)]^T \\ &\Delta\mathbf{u} = [\Delta\mathbf{u}(k), \Delta\mathbf{u}(k+1), \cdots, \Delta\mathbf{u}(k+N-1)]^T \end{aligned} \tag{36}$$

The both ways of the output prediction, following from matrix notations Eqs. (33) and (36), can be generalized to one expression

$$\hat{\mathbf{y}} = \mathbf{f} + \mathbf{G}\mathbf{u} \tag{37}$$

The implication of described predictions (Eqs. (33) and (36)) consists in the use. The former, absolute form is the simplest in use. The latter, incremental form represents integration that can solve steady state error. For design of predictive control differences in Eq. (37) are not significant and they do not change its derivation that proceeds from quadratic criterion (26).

3.1.2 Minimization of quadratic criterion

The criterion (26) can be simply modified to the following matrix product

$$J_k = \left[[\hat{\mathbf{y}} - \mathbf{w}]^T, \mathbf{u}^T \right] \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda} \end{bmatrix} \begin{bmatrix} [\hat{\mathbf{y}} - \mathbf{w}] \\ \mathbf{u} \end{bmatrix} = \mathbf{J}^T \mathbf{J} \tag{38}$$

For design, only square root is necessary

$$\mathbf{J} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda} \end{bmatrix} \begin{bmatrix} [\hat{\mathbf{y}} - \mathbf{w}] \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda} \end{bmatrix} \left(\begin{bmatrix} \hat{\mathbf{y}} \\ \mathbf{u} \end{bmatrix} - \begin{bmatrix} \mathbf{w} \\ \mathbf{0} \end{bmatrix} \right) \tag{39}$$

In criterion the prediction Eq, (37) is substituted for $\hat{\mathbf{y}}$ and the criterion is modified to over-determined system on which QR decomposition[14,15] is applied and the final control actions are given by Eq. (40)

$$\begin{aligned} \mathbf{A}\mathbf{u} &= \mathbf{b} \qquad / \mathbf{Q}^T \\ \mathbf{Q}^T \mathbf{A}\mathbf{u} &= \mathbf{Q}^T \mathbf{b} \\ \mathbf{R}\mathbf{u} &= \mathbf{c} \;\Rightarrow\; \mathbf{u} \end{aligned} \tag{40}$$

The illustrative graphical representation of described control is shown in following figures.

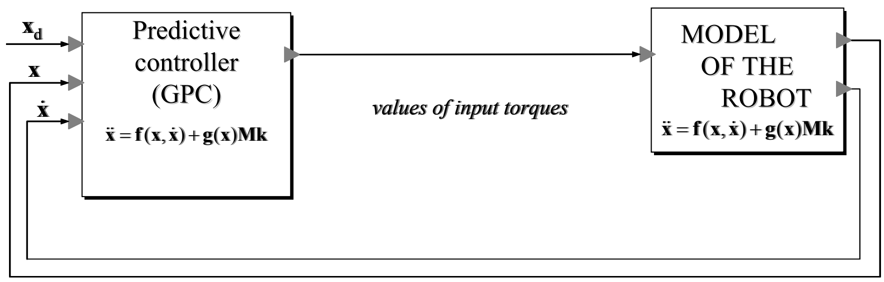

Figure 4. Predictive control of mechanical structure (assumption of ideal drives; transfer = 1).

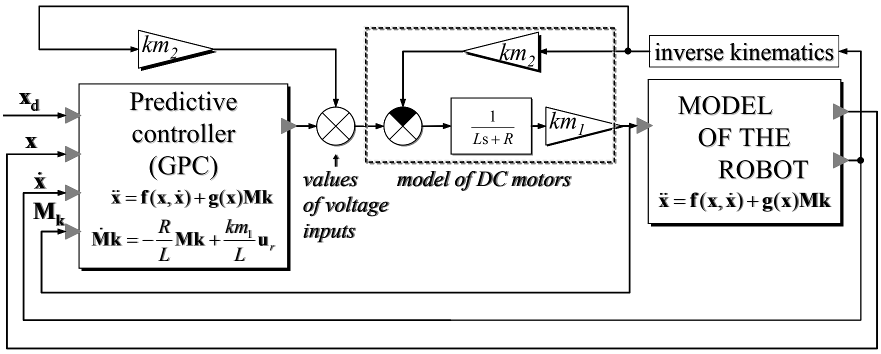

Figure 5. Predictive control of simple robotic system (mechanical structure and its drives).

The scheme in Fig. 9 corresponds to equation (7) and also to equation (10) with condition back recomputation of generalized force effects to really used torques on drives. The scheme in Fig. 10 is in accordance with equation (7), (24) and (25). In the both schemes based on derivation above, in computations of predictive controllers, the penalization **λ** (one parameter in quadratic criterion) determines magnitude of the redistributed loss on considered horizon of the prediction *N* (second parameter in quadratic criterion).

The different choice of penalization **λ** (mostly in a range 0 ÷ 1) together with length of horizon *N* enables to level generate control actions so that the available drives were not fitfully exerted. However, distributed changes of torques are achieved with cost of certain loss (error), theoretically equal to value of the criterion.

4. EXAMPLES AND CONCLUSIONS

The following figure - Fig. 6 demonstrates predictive controller, which considers the global robot model (29), i.e. it takes into account the model of the drives (DC motors). As a model, the redundantly actuated structure shown in Fig. 1 was used.

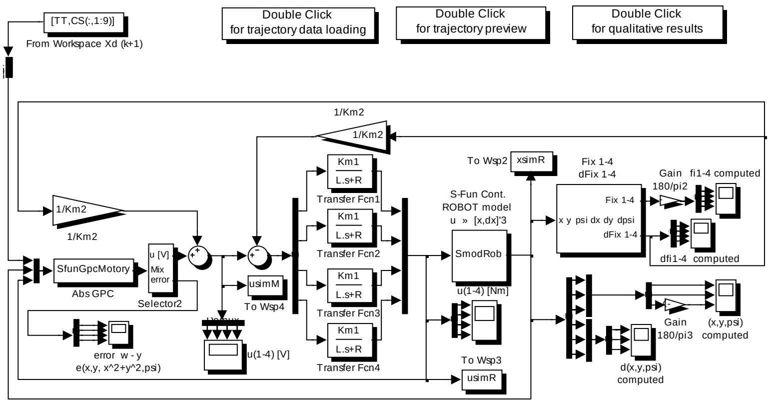

Figure 6. Predictive Control circuit with global robot model.

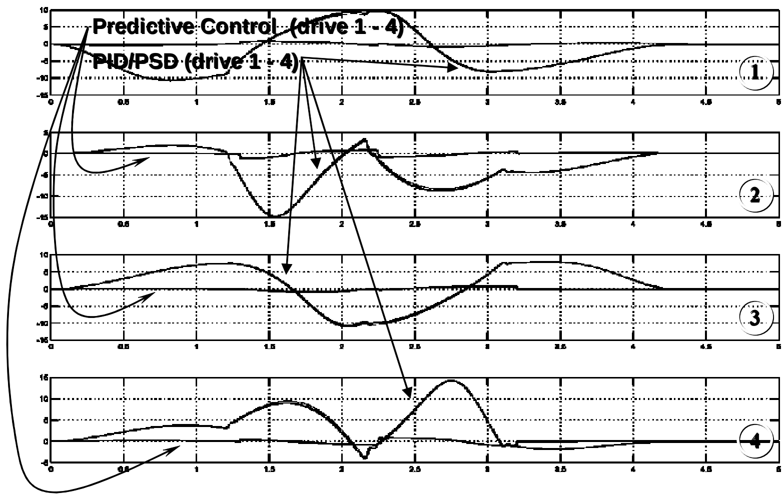

Figure 7. Time histories of four inputs (four torques)
during parallel simulation of PID/PSD and predictive controllers.

The time histories Fig. 7 were recorded during the motion along "S - shaped" trajectory. Fig. 7 illustrates the both predictive control algorithms, i.e. both mechanical and global model. The results of predictive controllers look similar, but precise values are different.

From practical point of view, the setting of parameters of PID/PSD is not simple task. The Fig. 7 illustrates the properties of PID/PSD control deducing the actions only from the feedback, which cannot be energetically effective. At the same time, the Fig. 7 shows the advantages of the combination of feedforward action modulation with feedback inaccuracy and disturbance compensation –

more effective control (GPC) way with lower energetic demands. It was shown, that the classical approaches are energetically consuming. They should be, in a future, replaced by model-based strategies e.g. predictive or LQ control, which consider most of the available information on given controlled object, here robotic system with DC motors that is good example of mechatronic system.

ACKNOWLEDGEMENTS

This research is supported by GA ČR (101/03/0620, 2003-2005) “Redundant drives and measurement for hybrid machine tools” and GA ČR (102/02/0204, 2002-2004) “Design of adaptive intelligent controllers”.

REFERENCES

1. L.-W. Tsai, Robot analysis, The Mechanics of Serial and Parallel Manipulators, John Wiley & sons, Inc. 1999.
2. V. Stejskal, and M. Valášek, Kinematics and Dynamics of Machinery, Marcel Dekker, Inc. 1996.
3. A. Ordys, D. Clarke, A State - Space Description for GPC Controllers, INT. J. Systems SCI., vol.24, no.9, 1993, pp. 1727 - 1744.
4. M. Valášek, V. Bauma, S. Miláček, J. Spurný, Z. Šika, T. Vampola, Synthesis of Redundant parallel Robots, Computational Mechanics, Nečtiny 2000, pp. 413-420.
5. J. Böhm, K. Belda, and M. Valášek, Study of Control of Planar Redundant Parallel Robot. Proceedings of the IASTED Int. conference MIC 2001, Innsbruck 2001, pp. 694-699.
6. K. Belda, J. Böhm, and M. Valášek, State-Space Generalized Predictive Control for Redundant Parallel Robots, Mechanics Based Design of Structures and Machines, Vol. 31, No. 3, 2003, pp. 413-432.
7. L. Phillips, T. Nagle, Digital control systems analysis and design, New Jersey, Prentice Hall, 1995.
8. L. Sciavicco, B. Siciliano, Modeling and control of robot manipulators, The McGraw-Hill Companies, Inc. New York 1996.
9. K. Belda, Control of Redundant Parallel Structures of Robotic Systems, dissertation, Czech Technical University in Prague, 2002.
10. M. Valášek, and P. Steinbauer, Nonlinear Control of Multibody Systems, In: Euromech 99, Lisbon 1996, pp. 437 - 444.
11. K. Belda, Real-Time Simulation and Visualization of High-Level Model-Based Control of Redundant Parallel Robots, In Proc. MATLAB 02, conference VŠCHT, Prague, pp. 21-27.
12. R. Neugebauer, (editor), Development Methods and Application Experience of Parallel Kinematics, Reports from the IWU Vol. 16, ISBN 3-928921-76-2, IWU Chemnitz 2002.
13. P. Souček, Drives of Production Appliances, script of CTU in Prague, 1997, (in Czech).
14. Ch. Lawson, and R. Hanson, Solving Least Square Problems, Prentice - Hall, Inc., 1974.
15. H. G. Golub, Ch. F. L. Van, Matrix Computations, The Johns Hopkins Univ. Press, 1989.