Predictive Control of Redundant Parallel Robots and Trajectory Planning

Belda, K.; Böhm, J.

Abstract

The paper deals with the design of model-based multi-step discrete predictive control of redundantly actuated parallel robots. In the paper, there is an introduction of specific compensation of gravitation for vertical parallel robot configurations. Then, the effective design of control actions by predictive algorithms is presented. Finally, the trajectory planning (a time parameterization of desired robot paths) both from kinematical and dynamical optimization point of view is discussed. The paper concludes by demonstration of several experiments realized on real laboratory model of redundant parallel robot 'Sliding Star'.

1 Introduction

Nowadays, the further development of industrial robots, machine tools and centers depends also on results and capabilities of present research of control design. In practice, quality local (decentralized) PID-based control is usually available. There exists key question here: Is this type of control sufficient and safe for newly developed machines such as parallel robots? The answer is not on first view clear.

However, the long-run experiments can expose, besides difficulties with selection of the PID parameters, some problematic states e.g. antagonistic behavior or over-large energy consumption. To overcome these undesirable states, some model description of energy decoupling is required. Herewith requirement, the main advantage of PID control, control without knowledge of model, is disappeared.

In case, that the model is available then the use of some global (centralized) model-based control strategy is more effective. The suitable selection of such strategy is generalized predictive control (GPC) [1] considered in this paper. Nevertheless, the local control can be usefully used as a fast sub-drive-control, of which desired values are generated by main model-based controller e.g. by GPC.

Subsequent question is a transformation of the demands on the robot motion, i.e. to fulfill some path given e.g. by technology, to proper form usable in control design. The motion is a process running in time, therefore some time parameterization of the path i.e. trajectory planning has to be considered. In the paper, the trajectory planning will be outlined from kinematical and dynamical point of view. The main difference is that the kinematical way provides only time parameterization of given path independently of the robot. On the contrary, the dynamical way uses dynamical model dependent on real robot and performs simultaneously design of a trajectory shape and a trajectory time parameterization of its realization. The explanation of planning is placed after sections dealing with the control design.

2 Model arrangement

Parallel robots represent multi-body system, of which dynamical model can be described in physical coordinates by Lagrange's equations of mixed type leading to a system of differential algebraic equations [3]

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

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), $\boldsymbol{\Phi}_{\mathbf{s}}$ 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 $\mathbf{f(s)} = \mathbf{0}$ represents geometrical constraints.

Model (1) can be transformed to more suitable form in independent coordinates (labeled as $\mathbf{y}$),of which number corresponds to number of degrees of freedom. Such transformation eliminates Lagrange's multipliers and algebraic constraints. Herewith the differential algebraic model (1) changes to model of ordinary differential equations

$$\mathbf{R}^{T}\mathbf{M}\mathbf{R}\ddot{\mathbf{y}} + \mathbf{R}^{T}\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{y}} = \mathbf{R}^{T}\mathbf{g} + \mathbf{R}^{T}\mathbf{T}\mathbf{u} \tag{2}$$

where $\mathbf{R}$ is a Jacobian matrix representing null space of Jacobian $\boldsymbol{\Phi}_{\mathbf{s}}$. Obtained model (2) has a suitable form both for simulation and for real control.

2.1 Compensation of gravitation

The effects of gravitation have to be considered only in vertical planar and generally spatial robots. Individual elements of these robots change their positions, thus change also distribution of effects of gravitation on drives.

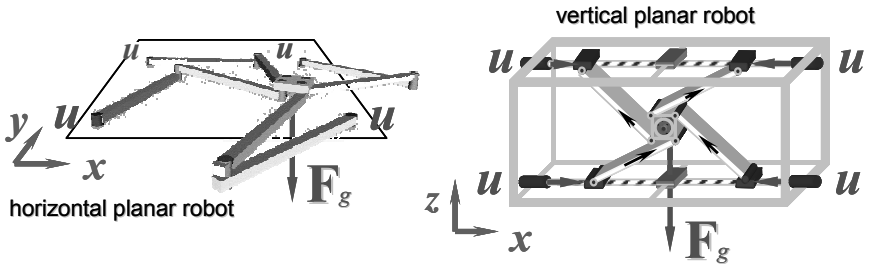

Figure 1: *Distribution of effects of gravitation on drives for different robot configurations*

The gravitational effects themselves are involved in the vector $\mathbf{g}$. To simplify control design, the model (2) is suitable to transform to the following form

$$\mathbf{R}^T\mathbf{M}\mathbf{R}\ddot{\mathbf{y}} + \mathbf{R}^T\mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{y}} = \mathbf{R}^T(\mathbf{g}_0 + \mathbf{g}_r) + \mathbf{R}^T\mathbf{T}\mathbf{u} \tag{3}$$

where a vector $\mathbf{g}_0$ is a factor of $\mathbf{g}$ depending only on velocities and a vector $\mathbf{g}_r$ is a factor of $\mathbf{g}$ containing only elements caused by effects of gravitation. Further, some terms in the model (3) can be entitled due to their meaning

$$\mathbf{F} = \mathbf{R}^T\mathbf{g}_r + \mathbf{R}^T\mathbf{T}\mathbf{u} = \mathbf{F}_g + \mathbf{F}_\tau \tag{4}$$

where $\mathbf{F}_g$ is a vector of force effects from gravitation, $\mathbf{F}_\tau$ is a vector of generalized force effects from inputs $\mathbf{u}$, altogether is a vector $\mathbf{F}$, force resultant computed for the coordinates of outputs $\mathbf{y}$, which represent the coordinates of movable platform. The model can be rewritten to a form and simplified

$$\ddot{\mathbf{y}} = (\mathbf{R}^T\mathbf{M}\mathbf{R})^{-1}\mathbf{R}^T(\mathbf{g}_0 - \mathbf{M}\dot{\mathbf{R}}\dot{\mathbf{y}}) + (\mathbf{R}^T\mathbf{M}\mathbf{R})^{-1}\mathbf{F} = \mathbf{f}_{(\mathbf{y},\dot{\mathbf{y}})} + \mathbf{g}_{(\mathbf{y})}\mathbf{F} \tag{5}$$

The obtained model is used for control design. Its form guarantees, that term $\mathbf{f}_{(\mathbf{y},\dot{\mathbf{y}})}$ in (5) equals zeros for arbitrary $\mathbf{y}$ and zero time derivatives $\dot{\mathbf{y}} = \mathbf{0}$. This property is needful for used linearization in next section. Real control actions are finally computed by expression unknown inputs $\mathbf{u}$ from (4) as follows

$$\mathbf{u} = \mathbf{T}^T\mathbf{R}(\mathbf{R}^T\mathbf{T}\mathbf{T}^T\mathbf{R})^{-1}(\mathbf{F} - \mathbf{F}_g) \tag{6}$$

2.2 On-line difference linearization for parallel robots

On-line difference linearization is a specific algorithm, which against standard linearization does not require any partial derivatives. It will be needful for multi-step predictive algorithms. To introduce it in brief, let us arise from nonlinear continuous model (5), of which nonlinear term $\mathbf{f}_{(\mathbf{y},\dot{\mathbf{y}})}$ is supposed to be linearized as follows [2]

$$\mathbf{f}_{(\mathbf{y},\dot{\mathbf{y}})} = \mathbf{a}_{\mathbf{1}(\mathbf{y},\dot{\mathbf{y}})}\mathbf{y} + \mathbf{a}_{\mathbf{2}(\mathbf{y},\dot{\mathbf{y}})}\dot{\mathbf{y}} \tag{7}$$

Let us suppose, that coordinates and derivatives representing current robot state $\mathbf{X} = [\mathbf{y},\dot{\mathbf{y}}]^T = [y_1,\cdots y_n,\dot{y}_1,\cdots \dot{y}_n]^T$ are available; zeros in $\mathbf{X}$ are replaced by suitable nonzero epsilon to avoid zero division. Finally, suppose some reference state $\mathbf{X}_r$ fulfilling the properties from section 2.1: $\mathbf{f}_{(\mathbf{y},\dot{\mathbf{y}})} = \mathbf{0} \,\big|_{[\mathbf{y},\dot{\mathbf{y}}]=[\mathbf{y}_r \text{ is arbitrary, } \dot{\mathbf{y}}_r = \mathbf{0}]}$.

Then the algorithm of the linearization is given by the following expression

$$\mathbf{f}_{(\mathbf{y},\dot{\mathbf{y}})} = \frac{\Delta\mathbf{f}_{11}(\circ)}{.\Delta y_1}\Delta y_1 + \cdots \frac{\Delta\mathbf{f}_{1n}(\circ)}{.\Delta y_n}\Delta y_n + \frac{\Delta\mathbf{f}_{21}(\circ)}{.\dot{y}_1}\dot{y}_1 + \cdots \frac{\Delta\mathbf{f}_{2n}(\circ)}{.\dot{y}_n}\dot{y}_n$$

$$\begin{aligned}
&\Delta\mathbf{f}_{1i}(\circ) = \mathbf{0},\ i = 1,n && \Rightarrow\ \mathbf{a}_{1(\mathbf{y},\dot{\mathbf{y}})} = \mathbf{0} \\
&\Delta\mathbf{f}_{21}(\circ) = \mathbf{f}([\mathbf{y},\dot{\mathbf{y}}]^T) - \mathbf{f}([\mathbf{y},0,\dot{y}_2,\cdots \dot{y}_n]^T),\cdots \\
&\Delta\mathbf{f}_{2n}(\circ) = \mathbf{f}([\mathbf{y},0,\cdots 0,y_n]^T) - \mathbf{f}([\mathbf{y},\mathbf{0}]^T) \\
&\qquad\qquad \Rightarrow \mathbf{a}_{2(\mathbf{y},\dot{\mathbf{y}})} = \left[\frac{\Delta\mathbf{f}_{21}(\circ)}{.\dot{y}_1},\cdots \frac{\Delta\mathbf{f}_{2n}(\circ)}{.\dot{y}_n}\right]
\end{aligned} \tag{8}$$

The linearized original function $\mathbf{f}_{(\mathbf{y},\dot{\mathbf{y}})}$ can be expressed as a state matrix $\mathbf{A}_c(\mathbf{X})$

$$\mathbf{f}(\mathbf{X}) = \begin{bmatrix} \dot{\mathbf{y}} \\ \mathbf{f}(\mathbf{y},\dot{\mathbf{y}}) \end{bmatrix} = \mathbf{A}_c(\mathbf{X})\,\mathbf{X} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{a}_1 & \mathbf{a}_2 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \dot{\mathbf{y}} \end{bmatrix} \tag{9}$$

The algorithm is performed on-line during real control with respect to current state of the robot.

3 Predictive control

Predictive control is a multi-step approach combining feedforward and feedback control design. Feedforward part is realized via predictions based on robot model. Feedback part, connected from output measurement, compensates some model inaccuracies and limited external disturbances. The design consists in repeated local minimization of quadratic criterion. It includes predictions from equations of predictions.

3.1 Equations of predictions

Equations of predictions form the character of predictive algorithms [4]:

- algorithms generating full control actions
- algorithms generating increments of control actions

Principle of equations of predictions is repetitive self-insertion of appropriate model. To derive general form of equations of predictions for full control actions, let us suppose discretized state formulation of model (5) with linearization (9) as follows

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{A}\,\mathbf{X}(k) + \mathbf{B}\,\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\,\mathbf{X}(k) \end{aligned} \tag{10}$$

Equations of predictions express future outputs $\hat{\mathbf{y}}$ from current state $\mathbf{X}(k)$

$$\begin{aligned} \hat{\mathbf{X}}(k+1) &= \mathbf{A}\,\mathbf{X}(k) + \mathbf{B}\mathbf{u}(k) \\ \hat{\mathbf{y}}(k+1) &= \mathbf{C}\,\mathbf{A}\,\mathbf{X}(k) + \mathbf{C}\,\mathbf{B}\mathbf{u}(k) \\ &\vdots \qquad\qquad\qquad\quad \vdots \\ \hat{\mathbf{X}}(k+N) &= \mathbf{A}^N\mathbf{X}(k) + \mathbf{A}^{N-1}\mathbf{B}\mathbf{u}(k) + \cdots \mathbf{B}\mathbf{u}(k+N-1) \\ \hat{\mathbf{y}}(k+N) &= \mathbf{C}\mathbf{A}^N\mathbf{X}(k) + \mathbf{C}\mathbf{A}^{N-1}\mathbf{B}\mathbf{u}(k) + \cdots \mathbf{C}\mathbf{B}\mathbf{u}(k+N-1) \end{aligned} \tag{11}$$

$$\text{i.e.} \quad \hat{\mathbf{y}} = \mathbf{f} + \mathbf{G}\,\mathbf{u}, \qquad \begin{aligned} \hat{\mathbf{y}} &= [\hat{\mathbf{y}}(k+1), \hat{\mathbf{y}}(k+2), \cdots \hat{\mathbf{y}}(k+N)]^T \\ \mathbf{u} &= [\mathbf{u}(k), \mathbf{u}(k+1), \cdots \mathbf{u}(k+N-1)]^T \end{aligned}$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{C}\,\mathbf{A} \\ \vdots \\ \mathbf{C}\,\mathbf{A}^N \end{bmatrix} \mathbf{X}(k), \quad \mathbf{G} = \begin{bmatrix} \mathbf{C} & \mathbf{B} & \mathbf{0} \\ \vdots & \ddots & \\ \mathbf{C}\,\mathbf{A}^{N-1}\mathbf{B} & \cdots & \mathbf{C}\,\mathbf{B} \end{bmatrix} \tag{12}$$

Although, the parallel robots have purely astatic character, described equations cannot guarantee zero steady-state error. It is caused e.g. by passive resistances, mechanical backlashes, hysteresis. Therefore, let us focus now on one example of incremental algorithm, which naturally by integrative character removes steady-state error problem.

The Algorithm arises from simple idea: To obtain integrative character, let us insert integrator directly to the model (5). Its realization is represented by following lines

$$\begin{aligned} \ddot{\mathbf{y}} &= \mathbf{f}(\mathbf{y},\dot{\mathbf{y}}) + \mathbf{g}(\mathbf{y})\mathbf{u} = \mathbf{a}_1(\mathbf{y},\dot{\mathbf{y}})\,\mathbf{y} + \mathbf{a}_2(\mathbf{y},\dot{\mathbf{y}})\,\dot{\mathbf{y}} + \mathbf{g}(\mathbf{y})\mathbf{u} \\ \mathbf{u} &= \int \mathbf{du}\, dt \quad \rightarrow \quad \dot{\mathbf{u}}(t) = \mathbf{du} = \tilde{\mathbf{u}} \\ \dddot{\mathbf{y}} &= \tfrac{d}{dt}(\ddot{\mathbf{y}}) = \tfrac{d}{dt}(\mathbf{a}_1\,\mathbf{y} + \mathbf{a}_2\,\dot{\mathbf{y}} + \mathbf{g}\,\mathbf{u}) = \mathbf{a}_1\,\dot{\mathbf{y}} + \mathbf{a}_2\,\ddot{\mathbf{y}} + \mathbf{g}\,\tilde{\mathbf{u}} \end{aligned} \tag{13}$$

After transformation to state-space formulation and discretization, the usual discrete state-space model is obtained

$$\begin{aligned} \tilde{\mathbf{X}}(k+1) &= \tilde{\mathbf{A}}\,\tilde{\mathbf{X}}(k) + \tilde{\mathbf{B}}\,\tilde{\mathbf{u}}(k) \\ \mathbf{y}(k) &= \tilde{\mathbf{C}}\,\tilde{\mathbf{X}}(k) \end{aligned} \tag{14}$$

where state $\tilde{\mathbf{X}}(k)$ consists of $\tilde{\mathbf{X}}(k) = [\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}]^T$. In general case, it is necessary to compute this extended state by state observer. In case of suitable selection of state variables (i.e. having physical interpretation) as in our case, the state can be expressed as follows

$$\tilde{\mathbf{X}}(k) = \left[\mathbf{y}(k), \quad \frac{\mathbf{X}(k) - \mathbf{X}(k-1)}{.Ts} \right]^T \tag{15}$$

Equations of predictions with model (14) are obtained identically as (11) and (12).

In case of model (14), the real values of control actions are computed from previous values. Values $\tilde{\mathbf{u}}(k)$ represent in view of (14) increments of control actions.

$$\mathbf{u}(k) = \mathbf{u}(k-1) + \tilde{\mathbf{u}}(k) \tag{16}$$

3.2 Minimization of quadratic criterion

The second fundamental part of predictive design is a minimization of the criterion.

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

where N, No and Nu are horizons; $\overline{\mathbf{Q}}_{\mathbf{y}}$ and $\overline{\mathbf{Q}}_{\mathbf{u}}$ are penalizations; and $\mathbf{w}_{(k+j)}$ are desired values. Result of the minimization is real values of control actions. To provide effective minimization, square-root approach with matrix representation of the criterion (17) is suitable to use

$$J_k = [(\hat{\mathbf{y}} - \mathbf{w})^T, \mathbf{u}^T] \begin{bmatrix} \mathbf{Q}_{\mathbf{y}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\mathbf{u}} \end{bmatrix}^T \begin{bmatrix} \mathbf{Q}_{\mathbf{y}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\mathbf{u}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ \mathbf{u} \end{bmatrix} \tag{18}$$

where $\mathbf{Q}_{\mathbf{y}} = diag(\overline{\mathbf{Q}}_{\mathbf{y}})$ and $\mathbf{Q}_{\mathbf{u}} = diag(\overline{\mathbf{Q}}_{\mathbf{u}})$. Square-root minimization means that only square-root of criterion representing a vector is minimized. The Euclidean norm of mentioned vector is value of the criterion. The minimization leads to a system of algebraic equations (19)

$$\begin{bmatrix} \mathbf{Q}_{\mathbf{y}} \mathbf{G} \\ \mathbf{Q}_{\mathbf{u}} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{Q}_{\mathbf{y}} (\mathbf{w} - \mathbf{f}) \\ \mathbf{0} \end{bmatrix} = \mathbf{0} \tag{19}$$

In (19), the prediction (12) is considered. The system (19) itself contains more rows than columns and in case of redundant inputs represents deficient rank system, which is solved by orthogonal-triangular decomposition. The decomposition reduces excess rows and can solve even linearly dependent columns.

4 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) 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.

4.1 Kinematically optimal design of robot trajectories

Kinematically optimal design is based on elementary laws of kinematics. During design, the following characteristics have to be successively determined

- length of path and possible range of rotation
- approximate time for reaching the end point of path
- geometrical parameter for positions and rotations
- real coordinates of segments and appropriate time derivatives

Let us outline step by step the determination of mentioned characteristics. The path length ℓ and rotation range ψ can be generally determined as follows

$$\ell = \int_s ds\,, \qquad \psi = \psi_{final} - \psi_{inital} \tag{20}$$

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

$$a = \frac{dv}{dt} \rightarrow t_1 = \frac{2\,\ell}{v_{inital} + v_{final}}\,, \quad \alpha = \frac{d\omega}{dt} \rightarrow t_2 = \frac{2\,\psi}{\omega_{inital} + \omega_{final}} \tag{21}$$

In expressions, double integration is performed and from obtained times t_1 and t_2 the higher, labeled as t_{final} is chosen.

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 $s, v, a, \psi, \omega, \alpha$

$$a(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3\,, \quad \alpha(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 \tag{22}$$

By double integration of (22), the system of algebraic equations for unknown coefficients $a_0, a_1, a_2, a_3, \alpha_0, \alpha_1, \alpha_2, \alpha_3$ is composed. Computed coefficients is used for determination of geometrical parameter for positions and rotations

$$p = s(t),\ p \in \langle 0,\ \ell \rangle,\ \ \rho = \psi(t),\ \rho \in \langle 0,\ \psi_f \rangle \tag{23}$$

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 segmentally 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. At the latter case, case of general curves, the geometrical parameter serves as a selector of coordinates $\mathbf{w} = [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}]$ from their appropriate table of given curve. The velocities and accelerations can be determined by expression

$$\begin{aligned} \mathbf{v} &= [v_{x0}, v_{y0}, v_{z0}, (\boldsymbol{x}(2{:}end) - \boldsymbol{x}(1{:}end{-}1))/Ts, (\Delta \boldsymbol{y})/Ts, (\Delta \boldsymbol{z})/Ts], \\ \mathbf{a} &= [0,0,0, (\mathbf{v}_x(2{:}end{-}1) - \mathbf{v}_x(1{:}end{-}2))/Ts, (\Delta \mathbf{v}_y)/Ts, (\Delta \mathbf{v}_z)/Ts] \end{aligned} \tag{24}$$

This computation is only approximation. Therefore, the parameterization accuracy depends on the selection of initial length of substitutive abscissa segments.

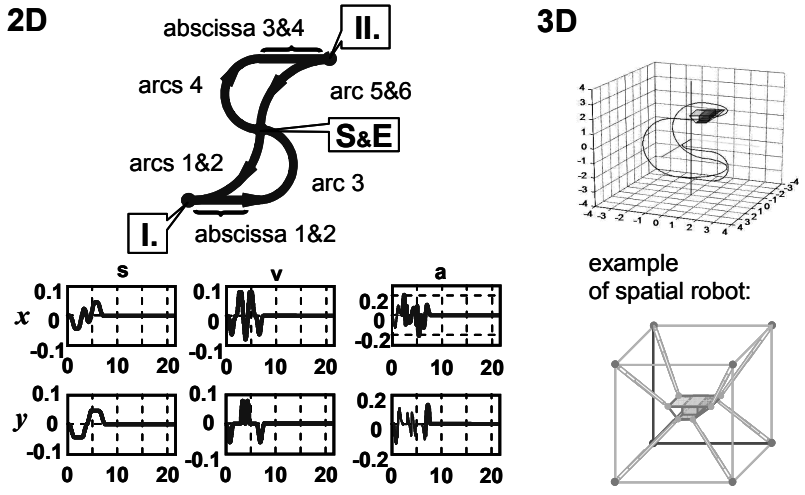

Figure 2: *Example of kinematically optimal design of 2D and 3D robot trajectories*

4.2 Dynamically optimal design of robot trajectories

Dynamically optimal design is another way of preparing the desired values for real control. The design does not use kinematical laws, but it is based on dynamical relations involved in the model of considered robot e.g. model (2). The time-parameterized trajectories are generated by simulation of specific control task. This section outlines two ways:

- Curve-based design – only start and end points are given
- Range-space design – start and end points and bounds are given

4.2.1 Curve-based design of trajectories

The dynamical curve-based design considered here is 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", see Figure 3.

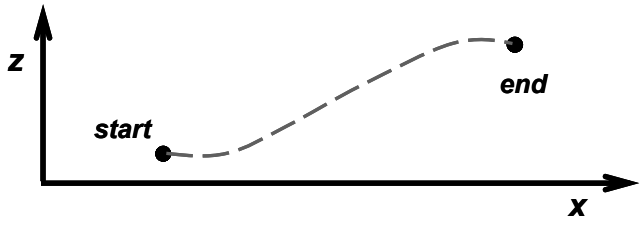

Figure 3: *Situation at dynamically curve-based optimal design*

As a suitable way, predictive control can be used again. Predictive design partly consists in the composition of the equations of predictions, which involve the dynamical robot model and partly consists in minimization of quadratic criterion (17).

The criterion includes several adjustable parameters: the horizons N, No and Nu, penalizations $\overline{\mathbf{Q}}_{\mathbf{y}}$ and $\overline{\mathbf{Q}}_{\mathbf{u}}$, 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 design are defined so that correspond to end state, respectively, to the end position i.e. $\mathbf{w} = [\mathbf{w}_{(k+j)}, \cdots \mathbf{w}_{(k+N)}] \,|\, \mathbf{w}_{(k+j)} = const., j = No+1, \cdots N$. To distribute the input energy in whole time interval determined by time T, i.e. to design the trajectory with suitably distributed energy, the described parameters of 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, in the criterion (18), the matrix $\mathbf{G}$ and corresponding differences $(\mathbf{w} - \mathbf{f})$ contain only last k rows and elements, respectively

$$\left[\begin{array}{cc|c} \mathbf{Q}_y & \mathbf{0} & \mathbf{G} \\ \hline \mathbf{0} & \mathbf{Q}_u & \mathbf{1} \end{array}\right]\left[\begin{array}{c|c} \mathbf{w}-\mathbf{f} \\ \hline \mathbf{0} \end{array}\right] = \left[\begin{array}{ccc|c} \mathbf{CA}^{N-No}\mathbf{B} & \cdots & \mathbf{CB} & (\overline{\mathbf{w}}-\overline{\mathbf{f}})_{N-No} \\ \vdots & \ddots & & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} \cdots & \mathbf{CB} & (\overline{\mathbf{w}}-\overline{\mathbf{f}})_{N} \\ \hline \overline{\mathbf{Q}}_u & & \cdots \mathbf{0} & \mathbf{0} \\ & \ddots & & \vdots \\ \mathbf{0} & \cdots & \overline{\mathbf{Q}}_u & \mathbf{0} \end{array}\right] \tag{25}$$

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

$$N := Nmax, Nmax - 1, \cdots, Nmin + 1, Nmin, \quad Nmin > k \tag{26}$$

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

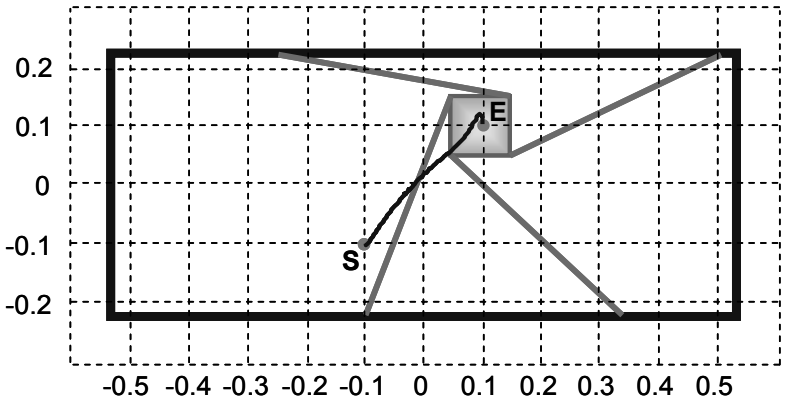

Figure 4: *Example of dynamically planned trajectory for vertical robot 'Sliding Star'*

4.2.2 Range-space design of trajectories

The dynamical range-space design serves for design of trajectories, which are defined by start point and number of bounds (e.g. in 2D $\boldsymbol{r}_a, \boldsymbol{r}_b$; see Figure 5) specifying free corridor for robot motion.

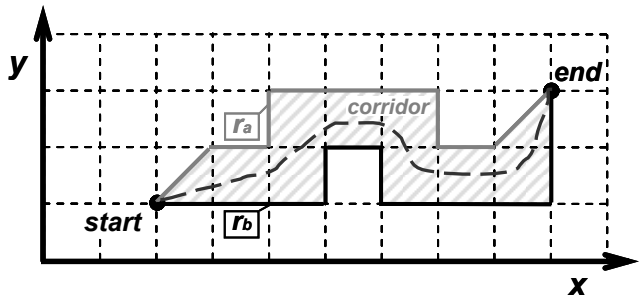

Figure 5: *Situation at dynamically range-space optimal design*

The range-space design provides planning of smooth trajectories for robot motion within specified bounds. For described design, the predictive control is used again. It partly includes robot dynamics and also provides optimal distribution of input energy. Used quadratic criterion is modified due to new required bounds [5], which substitute usual desired values $\mathbf{w}$

$$J_k = \sum_{j=No+1}^{N} \left(\left\| (\hat{\mathbf{y}}_{(k+j)} - \mathbf{r}_{a(k+j)}) \overline{\mathbf{Q}}_{ra} \right\|^2 + \left\| (\hat{\mathbf{y}}_{(k+j)} - \mathbf{r}_{b(k+j)}) \overline{\mathbf{Q}}_{rb} \right\|^2 \right) + \sum_{j=1}^{Nu} \left\| \mathbf{u}_{(k+j-1)} \overline{\mathbf{Q}}_{\mathbf{u}} \right\|^2 \quad (27)$$

in matrix representation

$$J_k = \left[(\hat{\mathbf{y}} - \mathbf{r}_a)^T, (\hat{\mathbf{y}} - \mathbf{r}_b)^T, \mathbf{u}^T \right] \begin{bmatrix} \mathbf{Q}_{ra} & \cdots & \mathbf{0} \\ \vdots & \mathbf{Q}_{rb} & \vdots \\ \mathbf{0} & \cdots & \mathbf{Qu} \end{bmatrix}^T \begin{bmatrix} \mathbf{Q}_{ra} & \cdots & \mathbf{0} \\ \vdots & \mathbf{Q}_{rb} & \vdots \\ \mathbf{0} & \cdots & \mathbf{Qu} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{r}_a \\ \hat{\mathbf{y}} - \mathbf{r}_b \\ \mathbf{u} \end{bmatrix} \quad (28)$$

In this design is also important selection of new output penalizations $\overline{\mathbf{Q}}_{ra}$ and $\overline{\mathbf{Q}}_{rb}$, which are selected in usual control process as identity matrices. The penalizations are not constant, but during the design are dynamically changed according to current bounds.

In simple terms, changing bound has higher weight, therefore the appropriate penalization has higher values and vice versa. In case, that the current bounds have constant values then the penalizations have the same values.

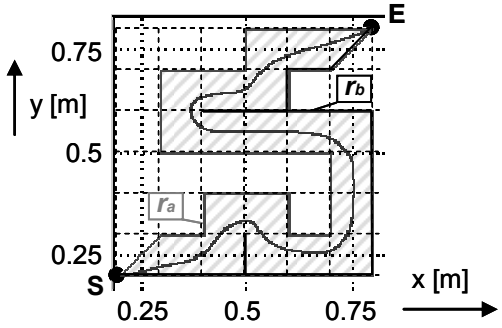

Figure 6: *Example of dynamically planned trajectory via range-space control.*

5 Real experiments

Real experiments were performed on the laboratory model of robot 'Sliding Star' (Figure 7). The robot represents vertical planar parallel configuration with redundant actuation. Individual elements lie on different levels of potential energies and during motion change their own potential energy too, therefore force effects of gravitation has to be considered e.g. as it is indicated in section 2.1.

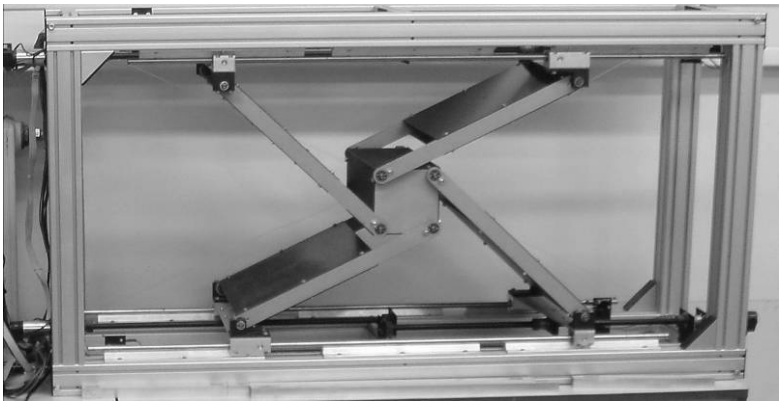

Figure 7: *Laboratory model of parallel robot 'Sliding Star'*

The kinematically optimal trajectory for experiments is shown in Figure 8.

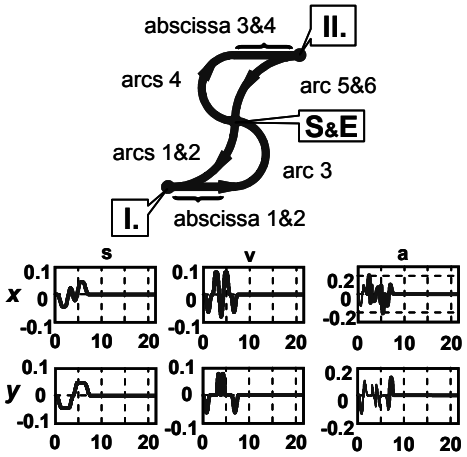

Figure 8: *Testing trajectory for robot 'Sliding Star'*

The aim of the experiments was to prove potency of algorithms of predictive control applied to parallel redundantly actuated robots. Individual algorithms in C code were implemented in MATLAB-Simulink environment (see Figure 8) and compiled to digital signal processor from dSPACE company.

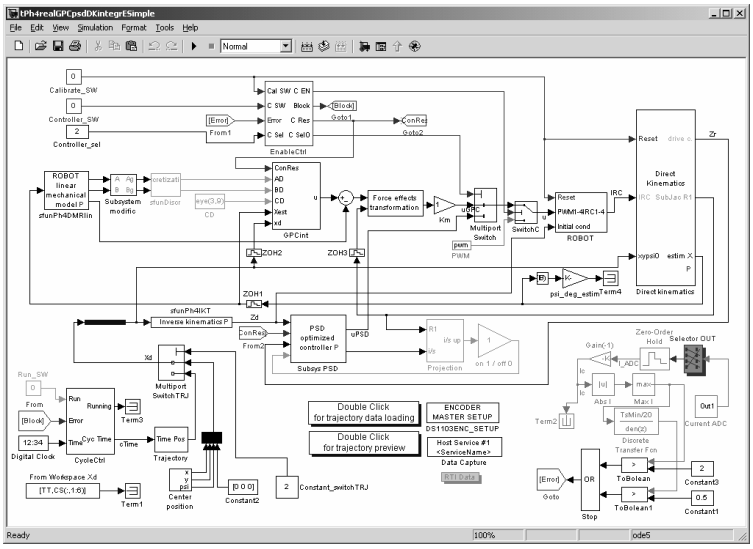

Figure 9: *Simulink scheme for predictive control (PSD controller serves only for comparison)*

The following figure illustrates the results of experiments. Time histories of control errors and control actions represented by measured currents on drives are shown.

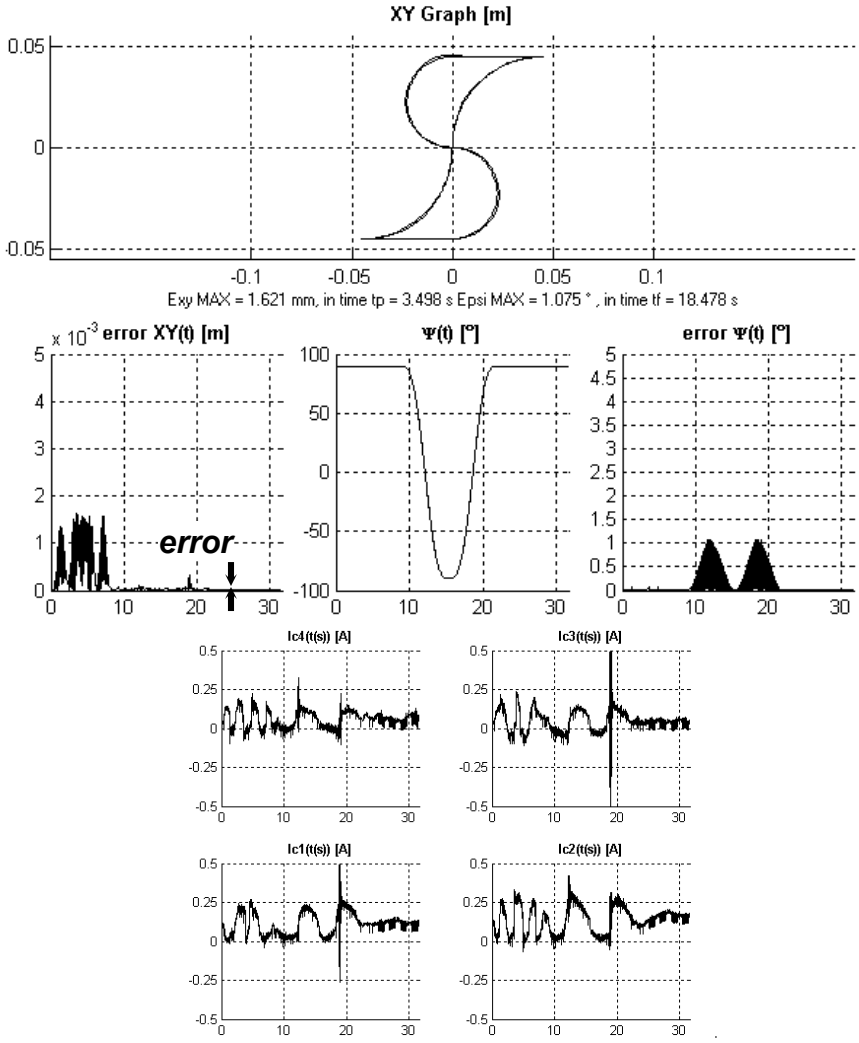

Figure 10: *Time histories of predictive algorithm generating increments of control actions*

The time histories, respective, the real experiments prove to successful process of control and possibility to use predictive control – model based control strategy – for improvement of current local approaches.

6 Conclusion

The paper presents brief outline of theory background of predictive control and also its utilization in dynamically optimal trajectory planning. The theory is proved by real experiments on laboratory model of parallel robot 'Sliding Star'. As a conclusion for future research into control of parallel robots is a utilization of model-based approaches, which provide optimal distribution of input energy, supplemented with fast local independent controllers build in individual drives.

The trajectory planning is another important task. The dynamically optimal design of trajectories described in the paper establishes basic idea, how to avoid obstacles in robot workspace. Especially range-space predictive control, possibly achieved on-line, is interesting for feedback-camera control systems.

Acknowledgements

The authors thank to prof. M. Valášek from Dept. of Mechanics of Bodies, FME, enabling them to test algorithms on robot model 'Sliding Star' and to eng. P. Píša from Dept. of Control Engineering, FEE, for support at implementation. They are from Czech Technical University in Prague. Moreover, the authors appreciate kind support of Grant Agency of the Czech Republic by grants (102/06/P275, 2006/08) 'Model-based Control of Mechatronic Systems for Robotics' and (102/05/0271, 2005/07) 'Methods of Predictive Control, Algorithms and Implementation'.

Literature

[1] Ordys, A.; Clarke, D.: A State - Space Description for GPC Controllers. In: Int. Journal Systems SCI., Vol. 24, No. 9, 1993, pp. 1727 - 1744.

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

[3] Stejskal, V.; Valášek, M. In: Kinematics and Dynamics of Machinery, Marcel Dekker, New York, 1996.

[4] Belda, K.; Böhm, J. et al: In: http://as.utia.cz/asc, KBweb © 2005.

[5] Pekař, J. et al: Control of CSTR using MPC based on Mixture Distribution. In: 6th IFAC Symposium on Nonlinear Control Systems, Stuttgart, 2004.