

STATE-SPACE GENERALIZED PREDICTIVE CONTROL FOR REDUNDANT PARALLEL ROBOTS

Květoslav Belda, Josef Böhm, Michael Valášek

Institute of Information Theory and Automation,

Academy of Sciences of the Czech Republic,

Department of Adaptive Systems.

Pod vodárenskou věží 4, 182 08 Praha 8 – Libeň, Czech Republic.

KEYWORDS: Parallel Robot Construction, Generalized Predictive Control,
Drive Redundancy.

ABSTRACT:

The paper deals with the design and properties of Generalized Predictive Control (GPC) for path control of redundant parallel robots. Redundant parallel classification means redundant number of actuators, i.e. more actuators than degrees of freedom of the robot. Control of such structures suffers from several new control problems like potential inconsistency of steady state positions or non-uniqueness of control actions. The paper explains classical direct derivation of GPC and its modification based on square root two-step design of control actions for solving the control problems. As an example for verification of algorithms, a prototype of a planar redundant parallel robot is used. Both design approaches are compared and several possibilities of extensions are presented for taking into consideration additional requirements like smooth course of actuators or fulfillment of the anti-backlash condition.

1. INTRODUCTION

With continuous demand for new powerful equipments for industrial production, the majority of automation of all technological processes is being developed and improved. However, the next development in industrial area is constrained by deficit of powerful machines like robot manipulators with proportional dynamics and stiffness. Their constructions should provide the achieving of higher accuracy with keeping of their dexterity.

In order to satisfy the requirements on increased dynamics and stiffness of robot manipulators there are currently developed so called parallel robots or parallel kinematics. They have promising mechanical properties, however they bring new control problems to be solved for their successful application.

The parallel structures in robotics [1, 2] (Fig.1) can be simply understood as a movable truss constructions or as robots with movable platform (tool or workpiece gripper) supported by more links. Their advantages are given by the fact that mass is decreased by fixing almost all drives to the robot frame, and by the fact that the overall stiffness is increased as known from trusses, both improving the ratio between movable mass and stiffness. On the other hand, the workspace of such structures is generally smaller and moreover it can contain singular positions (singularities) where the structure acquires additional degrees of freedom for example by intersection of some links.

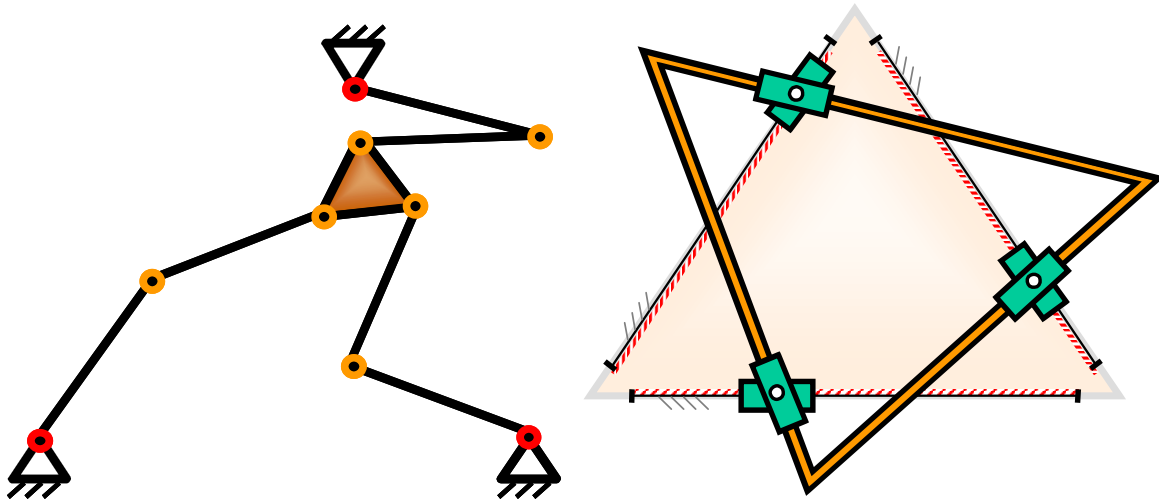


Fig. 1. Examples of parallel structures;
 (3 RRR [13] and 3 PRP [1] parallel robot; /R-rotational, P-prismatic pair/).

In order to profit from advantages of parallel structures, their disadvantages have to be eliminated. One possible solution may be the redundancy of actuators, i.e. providing more actuators than degrees of freedom (DOFs) of the parallel robot. The platform of the robot is then supported by more links than necessary (Fig. 2). Its idea can be simply formulated [13] as follows: if a certain group of links is in a position that leads to a singularity, then another group of links can be used among many available combinations that is not in a singular position.

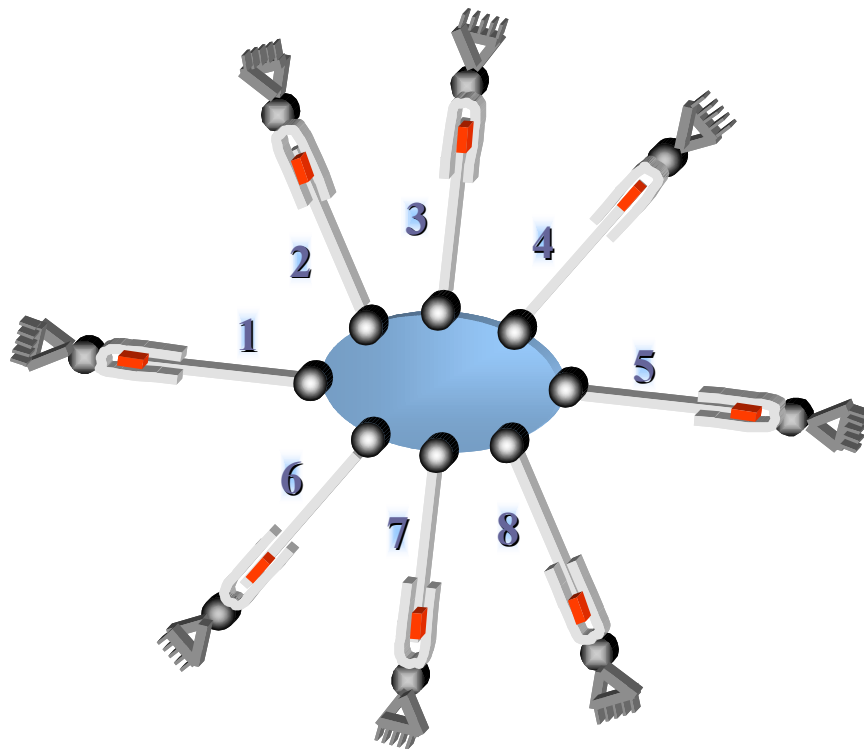


Fig. 2. The principle of redundant parallel robots.

Examples of planar redundant structures are shown in Fig. 3, spatial ones are given in [13].

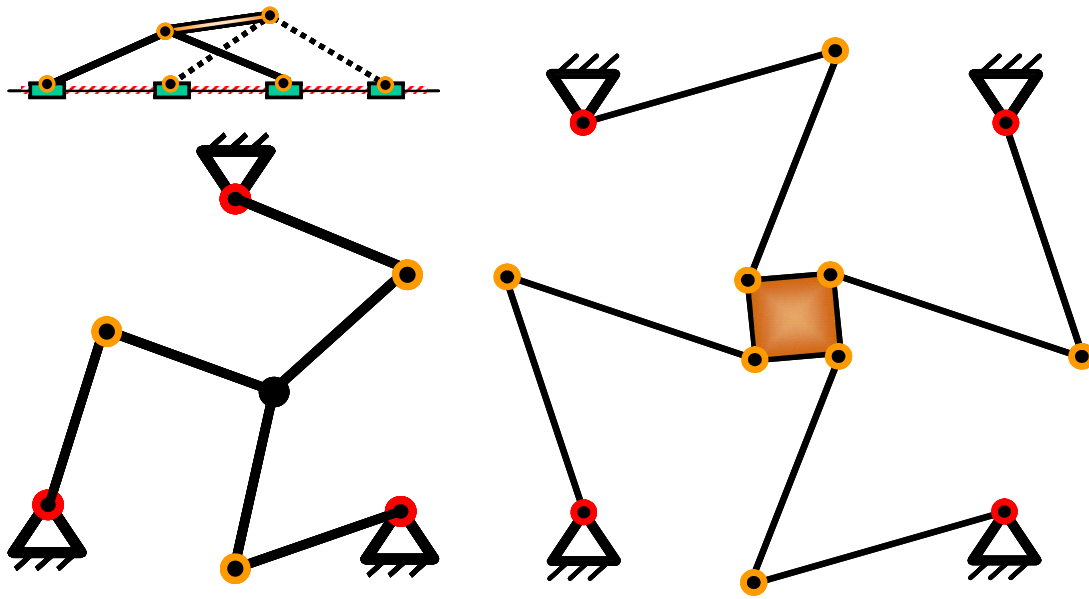


Fig. 3. Examples of redundant parallel structures; (4 PRR [14], 3 RR&R [11] and 4 RRR [13] parallel robot, /R-rotational, P-prismatic pair/).

The redundant parallel structures reduce the problems of parallel kinematics and moreover improve the stiffness, dynamics and accuracy of robot manipulators. However, there have arisen new problems regarding their control. The first fundamental problem is the potential inconsistency of steady state positions, more precisely the inconsistency between the model used for the control and the real structure. It appears especially in case of presence of integrating components in the controller and may lead to actuator mutual fighting. The second fundamental problem is the non-uniqueness of control actions, i.e. unique redundant robot motion corresponds to many actuator behaviors. Even more, the traditional literature (e.g. [15]) about robot control does not deal with control of parallel robots or even with redundant parallel robots. There appears only initial approaches to this topic (e.g. [11]).

The aim of this paper is to investigate the control approach by Generalized Predictive Control (GPC) [3, 4] as an example of high-level model-based control adjusted for redundant parallel robots. This approach decreases the first fundamental problem, because the presence of robot model in GPC controller without integrating components enables to achieve comparable accuracy as by traditional control with integrating components. This approach also offers the solution of the second fundamental problem by systematic design of an efficient unique solution among infinite many, e.g. for fulfilling additional requirements (anti-backlash or smoothing) on the control [5, 6].

Section 2 of the paper focuses on model description of a redundant parallel structure, then it continues with description of GPC technique both classical in section 3 and square root two-step in section 4 modified for redundant structures. The paper is concluded by an example in section 5 and final remarks in section 6.

2. DESCRIPTION OF THE ROBOT MODEL

The robot (e.g. Fig. 1 or Fig. 3) is a multibody system, which can be described in general by Lagrange's equations of mixed type [10]. These equations lead to a DAE system (the Differential-Algebraic Equations) in the following form:

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

where \mathbf{M} ($m \times m$) is a mass matrix, \mathbf{s} ($m \times 1$) is a vector of physical coordinates (their number m being higher than the number of degrees of freedom n /DOF/), Φ_s ($(m-n) \times m$) is an overall Jacobian of the system, $\boldsymbol{\lambda}$ ($(m-n) \times 1$) are Lagrange multipliers, \mathbf{g} ($m \times 1$) is a vector of right sides, \mathbf{T} ($m \times r$) is an unitary matrix adjusting the dimension of inputs \mathbf{u} ($r \times 1$) (torques), and $\mathbf{f}(\mathbf{s}(t)) = \mathbf{0}$ ($(m-n) \times 1$) represents geometrical constraints. Since the system is redundantly actuated, we have $r > n$.

The physical coordinates \mathbf{s} consist of the independent coordinates \mathbf{x} (coordinates of the cutting tool or gripper, in our example Cartesian coordinates $[x, y, \psi]$, Fig. 4), coordinates of drives \mathbf{q}_1 and other auxiliary geometrical coordinates \mathbf{q}_2 .

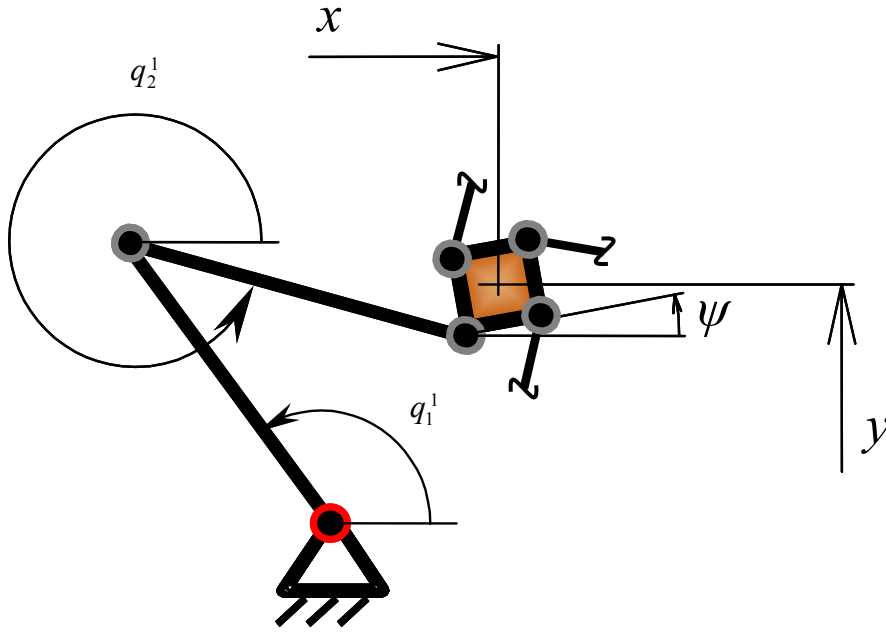


Fig. 4. Example of coordinates appearing in the robot structure, independent $\mathbf{x} = [x, y, \psi]$, drive $\mathbf{q}_1 = [q_1^1, q_1^2, \dots]$, and auxiliary $\mathbf{q}_2 = [q_2^1, q_2^2, \dots]$ coordinates.

For control design, we search for the most compact notation of system equation (1), i.e. a mathematical model, with minimum number of unknown parameters. The DAE system (1) represents $2m-n$ unknowns altogether consisting of m accelerations \ddot{s}_j and $m-n$ Lagrange multipliers λ_k .

Therefore, the system (1) must be transformed to independent coordinates [10]. This transformation reduces the number of differential equations and mainly it removes redundant Lagrange multipliers, corresponding to reaction forces and structural relations. Thus, we look for such a solution, which annuls element $\Phi^T \lambda$ and reduces physical coordinates (dependent and independent coordinates) to independent ones.

The solution is based on the null space of the Jacobian Φ_s , described by matrix \mathbf{R} fulfilling

$$\Phi_s \mathbf{R} = \mathbf{R}^T \Phi_s^T = \mathbf{0} \quad (2)$$

The matrix \mathbf{R} arises from the computation of the physical coordinates from independent ones

$$\mathbf{s} = \mathbf{s}(\mathbf{x}) \quad (3)$$

and differentiation with respect to time yielding

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{x}} \dot{\mathbf{x}} = \mathbf{R} \dot{\mathbf{x}} \quad (4)$$

It appears also in the second derivatives

$$\ddot{\mathbf{s}} = \dot{\mathbf{R}} \dot{\mathbf{x}} + \mathbf{R} \ddot{\mathbf{x}} \quad (5)$$

If we now insert equation (5) in equation (1) and perform its multiplication by the transposed Jacobian matrix \mathbf{R}^T , the equations of motion expressed in independent coordinates are ordinary differential equations of the second order (ODE)

$$\mathbf{R}^T \mathbf{M} \mathbf{R} \ddot{\mathbf{x}} + \mathbf{R}^T \mathbf{M} \dot{\mathbf{R}} \dot{\mathbf{x}} = \mathbf{R}^T \mathbf{g} + \mathbf{R}^T \mathbf{T} \mathbf{u} \quad (6)$$

System (6) is suitable not only for simulation of the dynamic behavior but also for control design.

As there are more control actions \mathbf{u} ($r \times 1$) than DOFs, which are equal to the dimension of \mathbf{x} ($n \times 1$), it is possible to introduce the force equivalent \mathbf{h} ($n \times 1$) of control actions on each independent coordinate \mathbf{x} in such way that

$$\mathbf{R}^T \mathbf{T} \mathbf{u} = \mathbf{h} \quad (7)$$

that is not uniquely solvable for \mathbf{u} as \mathbf{u} ($r \times 1$), \mathbf{h} ($n \times 1$) and $r > n$.

System (6) can be rewritten to the state-space formulation using state vector $\mathbf{X} = [\mathbf{x}, \dot{\mathbf{x}}]^T$, suitable for control

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}) + \mathbf{g}(\mathbf{X}) \mathbf{u} \quad (8)$$

$$\mathbf{x} = \mathbf{C} \mathbf{X} \quad (9)$$

It is generally a nonlinear state space description of the given system with output described by constant matrix \mathbf{C} ($n \times 2n$).

3 GENERALIZED PREDICTIVE CONTROL (GPC)

The principal task of control of robots is to accomplish their movement along a planned trajectory (technological requirements). For general control approaches (PID controllers), this is sometimes difficult. Therefore, new approaches like model-based control are being developed. They use the knowledge of the mathematical model (1) or (8) and globally optimize the whole process to design suitable control actions. One such global approach is multi-step Generalized Predictive Control (GPC) [3, 4, 5] based on local optimization of a quadratic criterion

$$J = \sum_{i=k+1}^{k+N} (y(i)^T Q_y y(i) + u(i-1)^T Q_u u(i-1)) \quad (10)$$

where y represents outputs, u represents inputs, Q_y and Q_u are output and input penalizations, k is a topical time step and N represents an horizon in which the criterion is considered.

The approach combines feed-back and feed-forward control. For evaluation of the criterion, the linearized state-space robot model is used. While optimizing the quadratic criterion, the model (8) must be linearized [7] and transformed to its discrete form. The discretization provides sufficient computational time for the design.

For lucidity, this chapter will be divided into four main parts, which follow the typical sequence of derivation and full design of Predictive Control:

- Linearization of continuous nonlinear model
- Discretization of linearized state model
- Classical design of control law
- Derivation of control law in root form

3.1 Linearization of Continuous Nonlinear Model

Let us proceed from system (8) and let us decompose it into the form

$$\dot{\mathbf{X}}(t) = \mathbf{A}(\mathbf{X}(t)) \mathbf{X}(t) + \mathbf{B}(\mathbf{X}(t)) \mathbf{u}(t) \quad (11)$$

corresponding to the typical linear form $\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{u}$. For this it is necessary to perform the decomposition of the nonlinear function $\mathbf{f}(\mathbf{X})$ into the form

$$\mathbf{f}(\mathbf{X}) = \mathbf{A}(\mathbf{X}) \mathbf{X} \quad (12)$$

which is not unique. However, several possible solutions are based on the expansion around the equilibrium states (working points) \mathbf{X}_r fulfilling

$$\mathbf{f}(\mathbf{X}) \Big|_{\mathbf{X}=\mathbf{X}_r} = \mathbf{0} \quad (13)$$

The expansion is based on the following identity: Let us have $\mathbf{X} = [x_1, x_2, x_3]^T$, and $\mathbf{f}(\mathbf{X}) \Big|_{\mathbf{X}=[x_1, x_2, x_3]^T}$. Let us choose the order of the state variables as x_3, x_1, x_2 . Then the decomposition around the equilibrium point $\mathbf{X} = [x_{r1}, x_{r2}, x_{r3}]^T$ is

$$\mathbf{A}(\mathbf{X}) = \left[\frac{\mathbf{f}([x_1, x_2, x_{r3}]) - \mathbf{f}([x_{r1}, x_2, x_{r3}])}{x_1}, \frac{\mathbf{f}([x_{r1}, x_2, x_{r3}]) - \mathbf{f}([x_{r1}, x_{r2}, x_{r3}])}{x_2}, \frac{\mathbf{f}([x_1, x_2, x_3]) - \mathbf{f}([x_1, x_2, x_{r3}])}{x_3} \right] \quad (14)$$

Identity (12) can be easily proved for $\mathbf{X} \neq \mathbf{X}_r$ by elementary simplification and is also fulfilled for $\mathbf{X} \rightarrow \mathbf{X}_r$ using l'Hospital rule as proved in [7].

3.2. Discretization of Linearized State Model

To design Predictive Control, we need to provide discretization of the state model, which will be used in real-time algorithms. To explain it, let us start from the decomposed state model (11) and let us assume that the state matrixes \mathbf{A} ($=\mathbf{A}(\mathbf{X}(t))$) \mathbf{B} ($=\mathbf{B}(\mathbf{X}(t))$) are constant within the considered interval of discretization $\langle t, t+\delta \rangle$ where δ is the sampling period. Assuming the matrices \mathbf{A} and \mathbf{B} to be constant yields

$$\dot{\mathbf{X}} = \mathbf{A} \mathbf{X} + \mathbf{B} \mathbf{u} \quad (15)$$

Discretization arises from solution of such first-order linear differential equations. As a solution of system (15), we search for the result in the following form:

$$\mathbf{X}(t) = \Phi(t) \mathbf{c}(t) \quad (16)$$

In order to obtain (16), we must perform two steps, firstly finding the homogenous solution and secondly, the relaxed solution:

1. Homogenous solution by formal integration:

$$\begin{aligned} \dot{\mathbf{X}} &= \frac{d\mathbf{X}}{dt} = \mathbf{A}\mathbf{X} \\ \int_c^x \frac{d\bar{\mathbf{X}}}{\bar{\mathbf{X}}} &= \mathbf{A} \int_{t_0}^t d\bar{t} \\ \mathbf{X} &= e^{\mathbf{A}(t-t_0)} \mathbf{c}(t) \end{aligned} \quad (17)$$

2. Relaxed solution (method of constant variation):

If we substitute the homogenous solution (17) for initial system (15), we obtain the following

$$\dot{\mathbf{c}}(t) = (e^{\mathbf{A}(t-t_0)})^{-1} \mathbf{B} \mathbf{u}(t) \quad (18)$$

with use of the identity $e^{\mathbf{A}(t-t_0)} e^{\mathbf{A}(t_0-t)} = 1$ for matrix inversion $(e^{\mathbf{A}(t-t_0)})^{-1} = e^{\mathbf{A}(t_0-t)}$

$$\mathbf{c}(t) = \mathbf{c}(t_0) + \int_{t_0}^t e^{\mathbf{A}(t_0-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \quad (19)$$

By substitution of expression (19) in homogenous solution (17), we obtain the relaxed solution

$$\begin{aligned} \mathbf{X}(t) &= e^{\mathbf{A}(t-t_0)} \mathbf{c}(t_0) + \int_{t_0}^t e^{\mathbf{A}(t-t_0)} e^{\mathbf{A}(t_0-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \\ &= e^{\mathbf{A}(t-t_0)} \mathbf{c}(t_0) + \int_{t_0}^t e^{\mathbf{A}(t-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \end{aligned}$$

where for $t = t_0$ we may conclude that $\mathbf{c}(t_0) = \mathbf{X}(t_0)$. Thus, we obtain the final solution

$$\mathbf{X}(t) = e^{\mathbf{A}(t-t_0)} \mathbf{X}(t_0) + \int_{t_0}^t e^{\mathbf{A}(t-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau \quad (20)$$

The continuous state response (20) can be used to obtain the desired discrete state formulation.

If we set

$$t_0 = k \delta \quad \text{and} \quad t = (k + 1) \delta \quad (21)$$

and if we mark

$$\mathbf{X}(k) \triangleq \mathbf{X}(t_0) = \mathbf{X}(k\delta), \quad \mathbf{X}(k+1) \triangleq \mathbf{X}(t) = \mathbf{X}(k\delta + \delta) \quad \text{and} \quad \mathbf{u}(k) \triangleq \mathbf{u}(k\delta) \quad (22)$$

then the state response (20) can be retransformed into state formulation (23)

$$\mathbf{X}(k+1) = e^{\mathbf{A}\delta} \mathbf{X}(k) + \int_{k\delta}^{k\delta+\delta} e^{\mathbf{A}(k\delta+\delta-\tau)} \mathbf{B} d\tau \mathbf{u}(k) \quad (23)$$

where we consider the inputs as step functions

$$\mathbf{u}(t) = \mathbf{u}(k\delta) = \text{konst.} \quad \text{for} \quad t \in \langle k\delta, k\delta + \delta \rangle \quad (24)$$

The exact definition of discrete state formulation (23) can be formally modified to form (25)

$$\mathbf{X}(k+1) = \mathbf{A}_D \mathbf{X}(k) + \mathbf{B}_D \mathbf{u}(k) \quad (25)$$

where matrixes \mathbf{A}_D and \mathbf{B}_D , with meaning arising from (23), can be simplified by Taylor series (series of function $e^{\mathbf{X}}$), which respects requirement of finite computational time:

$$\mathbf{A}_D = e^{\mathbf{A}\delta} = \mathbf{1} + \mathbf{A}\delta + \frac{\mathbf{A}^2 \delta^2}{2!} + \dots + \frac{\mathbf{A}^p \delta^p}{p!} + \text{rest}(Oh^{p+1}) \quad (26)$$

and with substitution $\mathbf{V} = \mathbf{A}(k\delta + \delta - \tau)$, $d\mathbf{V} = -\mathbf{A} d\tau$

$$\begin{aligned} \mathbf{B}_D &= \int_{k\delta}^{k\delta+\delta} e^{\mathbf{A}(k\delta+\delta-\tau)} \mathbf{B} d\tau = -\mathbf{A}^{-1} \int_{\mathbf{A}\delta}^0 e^{\mathbf{V}} d\mathbf{V} \mathbf{B} = -\mathbf{A}^{-1} (\mathbf{1} - e^{\mathbf{A}\delta}) \mathbf{B} = \\ &= -\mathbf{A}^{-1} (\mathbf{1} - \mathbf{1} - \mathbf{A}\delta - \frac{\mathbf{A}^2 \delta^2}{2!} + \dots + \frac{\mathbf{A}^p \delta^p}{p!} + \text{rest}(Oh^{p+1})) \mathbf{B} = \\ &= (\mathbf{1}\delta + \frac{\mathbf{A}\delta^2}{2!} + \dots + \frac{\mathbf{A}^{p-1} \delta^p}{p!} + \text{rest}(Oh^{p+1})) \mathbf{B} \end{aligned} \quad (27)$$

Computation can be suitably stopped after p^{th} member in the series, when norm of the rest of series $\|\text{rest}(Oh^{p+1})\|$ is smaller than a chosen computational accuracy ε .

3.3. Classical Design of Control Law

After linearization and discretization we have obtained a discrete version of the state formula which is completed by an output relation:

$$\begin{aligned} \mathbf{X}(k+1) &= \mathbf{A} \mathbf{X}(k) + \mathbf{B} \mathbf{u}(k) \\ \mathbf{x}(k) &= \mathbf{C} \mathbf{X}(k) \end{aligned} \quad (28)$$

where state \mathbf{X} is a composed vector $[\mathbf{x}, \dot{\mathbf{x}}]^T$ and \mathbf{x} has the same meaning as in differential equations (6), i.e. \mathbf{C} is a unitary rectangular matrix.

The idea of Predictive Control, based on optimization of a quadratic criterion, is to predict new unknown output values $\mathbf{x}(k+i)$, $i > 0$, from the real topical state $\mathbf{X}(k)$ and their use in the mentioned criterion. Prediction equations are

$$\begin{aligned} \mathbf{x}(k) &= \mathbf{C} \mathbf{X}(k) \\ \widehat{\mathbf{X}}(k+1) &= \mathbf{A} \mathbf{X}(k) + \mathbf{B} \mathbf{u}(k) \\ \widehat{\mathbf{x}}(k+1) &= \mathbf{C} \mathbf{A} \mathbf{X}(k) + \mathbf{C} \mathbf{B} \mathbf{u}(k) \\ \widehat{\mathbf{X}}(k+2) &= \mathbf{A}^2 \mathbf{X}(k) + \mathbf{A} \mathbf{B} \mathbf{u}(k) + \mathbf{B} \mathbf{u}(k+1) \\ \widehat{\mathbf{x}}(k+2) &= \mathbf{C} \mathbf{A}^2 \mathbf{X}(k) + \mathbf{C} \mathbf{A} \mathbf{B} \mathbf{u}(k) + \mathbf{C} \mathbf{B} \mathbf{u}(k+1) \\ &\vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \ddots \\ \widehat{\mathbf{X}}(k+N) &= \mathbf{A}^N \mathbf{X}(k) + \mathbf{A}^{N-1} \mathbf{B} \mathbf{u}(k) + \mathbf{A}^{N-2} \mathbf{B} \mathbf{u}(k+1) + \dots + \mathbf{B} \mathbf{u}(k+N-1) \\ \widehat{\mathbf{x}}(k+N) &= \mathbf{C} \mathbf{A}^N \mathbf{X}(k) + \mathbf{C} \mathbf{A}^{N-1} \mathbf{B} \mathbf{u}(k) + \mathbf{C} \mathbf{A}^{N-2} \mathbf{B} \mathbf{u}(k+1) + \dots + \mathbf{C} \mathbf{B} \mathbf{u}(k+N-1) \end{aligned} \quad (29)$$

Thus the prediction of \mathbf{x} for future time steps is given by the estimation

$$\widehat{\mathbf{x}} = \mathbf{G} \mathbf{u} + \mathbf{f} \quad (30)$$

where

$$\begin{aligned} \widehat{\mathbf{x}} &= [\widehat{\mathbf{x}}(k+1) \quad \widehat{\mathbf{x}}(k+2) \quad \dots \quad \widehat{\mathbf{x}}(k+N)]^T \\ \mathbf{u} &= [\mathbf{u}(k) \quad \mathbf{u}(k+1) \quad \dots \quad \mathbf{u}(k+N-1)]^T \end{aligned} \quad (31)$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{C} \mathbf{B} & \dots & 0 \\ \vdots & \ddots & \vdots \\ \mathbf{C} \mathbf{A}^{N-1} \mathbf{B} & \dots & \mathbf{C} \mathbf{B} \end{bmatrix} \qquad \mathbf{f} = \begin{bmatrix} \mathbf{C} \mathbf{A} \\ \vdots \\ \mathbf{C} \mathbf{A}^N \end{bmatrix} \mathbf{X}(k) \quad (32 \text{ a, b})$$

By this prescript for the prediction of future values of outputs, we can perform optimization of the quadratic criterion.

Let us assume the quadratic criterion (10) to be in a form suitable for tracking a desired trajectory, i.e.

$$J_k = \mathcal{E} \left\{ (\widehat{\mathbf{x}} - \mathbf{w})^T (\widehat{\mathbf{x}} - \mathbf{w}) + \mathbf{u}^T \boldsymbol{\lambda} \mathbf{u} \right\} = \mathcal{E} \left\{ (\mathbf{G} \mathbf{u} + \mathbf{f} - \mathbf{w})^T (\mathbf{G} \mathbf{u} + \mathbf{f} - \mathbf{w}) + \mathbf{u}^T \boldsymbol{\lambda} \mathbf{u} \right\} \quad (33)$$

where \mathcal{E} is an estimation operator for the mean value, $\widehat{\mathbf{x}}$ is a vector composed of predictions for the outputs $\mathbf{x}(k+i)|_{i=1,2,\dots,N}$ with ($Q_x = 1$), N marks a horizon of prediction and \mathbf{w} is a vector of desired values corresponding to vector $\widehat{\mathbf{x}}$, $\boldsymbol{\lambda}$ is an input penalization ($\cong Q_u$) and \mathbf{u} is a vector of future inputs to the robot for given time instants within the horizon N .

Assuming that the robot model is well known (deterministic system, variance $\sigma_x = 0$), we can omit the mean value operator \mathcal{E} . Then the criterion can be reformulated as

$$J_k = \mathbf{u}^T (\mathbf{G}^T \mathbf{G} + \lambda) \mathbf{u} + \mathbf{u}^T \mathbf{G}^T (\mathbf{f} - \mathbf{w}) + (\mathbf{f}^T - \mathbf{w}^T) \mathbf{G} \mathbf{u} + \mathbf{f}^T \mathbf{f} - \mathbf{f}^T \mathbf{w} - \mathbf{w}^T \mathbf{f} + \mathbf{w}^T \mathbf{w} \quad (34)$$

Searching for a local minimum, i.e. $J_k \stackrel{!}{=} \min$; equals the condition that the derivative with respect to \mathbf{u} is equal to zero. Thus the optimal control action is given as

$$\mathbf{u} = (\mathbf{G}^T \mathbf{G} + \lambda)^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f}) \quad (35)$$

or in expanded form:

$$\begin{aligned} \begin{bmatrix} \mathbf{u}(k) \\ \mathbf{u}(k+1) \\ \vdots \\ \mathbf{u}(k+N-1) \end{bmatrix} &= \left(\begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \dots & \mathbf{CA}^{N-1} \mathbf{B} \\ \mathbf{0} & \mathbf{CB} & & \mathbf{CA}^{N-2} \mathbf{B} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{CB} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{CA}^{N-1} \mathbf{B} & \mathbf{CA}^{N-2} \mathbf{B} & \dots & \mathbf{CB} \end{bmatrix} + \begin{bmatrix} \lambda & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \lambda & & \vdots \\ \vdots & & \lambda & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \lambda \end{bmatrix} \right)^{-1} \\ &\cdot \begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \dots & \mathbf{CA}^{N-1} \mathbf{B} \\ \mathbf{0} & \mathbf{CB} & & \mathbf{CA}^{N-2} \mathbf{B} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{CB} \end{bmatrix} \cdot \left(\begin{bmatrix} \mathbf{w}(k+1) \\ \mathbf{w}(k+2) \\ \vdots \\ \mathbf{w}(k+N) \end{bmatrix} - \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^N \end{bmatrix} \cdot \mathbf{X}(k) \right) \end{aligned} \quad (36)$$

This control law is ready to be used. However, we should note, that only the first element $\mathbf{u}(k)$ from vector \mathbf{u} is applied as control action for time instance $k \delta$.

If the penalization λ is larger than zero, the matrix to be inverted will be always regular. In case of constraints on the control actions we can use quadratic programming (QP algorithm) for solving the optimality problem.

3.4 Derivation of a Control Law in Square Root Form

In the previous section the problem of non-uniqueness of the solution of equation (7) is solved by the selection of a non-zero penalization λ in the quadratic performance index (34). This section deals with the derivation of a control law that enables to choose the non-unique control \mathbf{u} by fulfilling additional conditions. Furthermore, the advantages of the root form of the quadratic criterion (33) are used in this section. They are marked out by compact notation and good preparation for operations with huge matrices.

Let us start from the model in the form (6) with the substitution of (7). To this description there is applied the procedures of linearization (section 3.1), discretization (section 3.2) and composition of expressions for prediction of future outputs (section 3.3). Finally an equation equivalent to (30) is obtained. However, during this derivation and also in the next equations of this section the symbol \mathbf{u} is used instead of \mathbf{h} for convenience of preservation of traditional control notation.

Then the quadratic criterion (33) is rewritten for square root form to the following matrix product

$$J_k = [\widehat{\mathbf{x}} - \mathbf{w}]^T, \mathbf{u}^T \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \lambda^T \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \lambda \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{x}} - \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \mathbf{J}^T \mathbf{J} \quad (38)$$

where the product $\lambda^T \lambda$ in (38) formally substitutes parameter $\bar{\lambda}$ for input penalization. Thus, let us consider the parameter λ to express the square root of penalization not really changing its meaning.

Now we can work only with a vector criterion which arises as square root from (38)

$$\mathbf{J} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \lambda \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{x}} - \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \lambda \end{bmatrix} \left(\begin{bmatrix} \widehat{\mathbf{x}} \\ \mathbf{u} \end{bmatrix} - \begin{bmatrix} \mathbf{w} \\ \mathbf{0} \end{bmatrix} \right) \quad (39)$$

\mathbf{J} is a column vector with its Euclidean norm being the cost value of the root of the scalar criterion (38).

Firstly, the expression for prediction is substituted in the root (39) and it is adjusted to a form, where control actions \mathbf{u} are in a separate term:

$$\mathbf{J} = \begin{bmatrix} \widehat{\mathbf{x}} \\ \lambda \mathbf{u} \end{bmatrix} - \begin{bmatrix} \mathbf{w} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{G}\mathbf{u} + \mathbf{f} \\ \lambda \mathbf{u} \end{bmatrix} - \begin{bmatrix} \mathbf{w} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ \lambda \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{w} \\ \mathbf{0} \end{bmatrix} \quad (40)$$

Consecutively we look for such a \mathbf{u} , which minimizes the Euclidean norm. This is fulfilled, if \mathbf{J} is a zero vector, i.e.

$$\begin{bmatrix} \mathbf{G} \\ \lambda \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{w} \\ \mathbf{0} \end{bmatrix} = \mathbf{0}$$

which is an overdetermined system of equations (more rows than columns) for the optimal control \mathbf{u} :

$$\mathbf{A} \mathbf{u} - \mathbf{b} = \mathbf{0} \quad (41)$$

For solution, the orthogonal triangular decomposition [8, 9] is used. It reduces excess rows of matrix \mathbf{A} $[(2 \cdot N \cdot n) \times (N \cdot n)]$ and elements of vector \mathbf{b} $[2 \cdot N \cdot n]$ (N is a horizon of the criterion and n is the number of DOF) to an upper triangular matrix \mathbf{R} and vector \mathbf{c} by multiplication of whole equation system by orthogonal matrix \mathbf{Q}^T as follows:

$$\mathbf{Q}^T \mathbf{A} \mathbf{u} = \mathbf{Q}^T \mathbf{b}$$

$$\mathbf{R} \mathbf{u} = \mathbf{c}$$

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_z \end{bmatrix} \quad (43)$$

Vector \mathbf{c}_z is a residuum vector, whose Euclidean norm $|\mathbf{c}_z|$ is equal to the square root value of the criterion, i.e. $|\mathbf{c}_z| = \sqrt{J_k}$.

For the solution, we need only the upper part of the system (43), which can be simply solved for the unknown vector of control actions \mathbf{u} :

$$\begin{aligned} \mathbf{R}_1 \mathbf{u} &= \mathbf{c}_1 \\ \mathbf{u} &= (\mathbf{R}_1)^{-1} \mathbf{c}_1 \end{aligned} \quad (44)$$

Since the matrix \mathbf{R}_1 is upper triangular, we can find its inversion directly by back-run procedure.

By computation we obtain only a fictitious general force effect $\mathbf{u}=\mathbf{h}$, because only the first subvector (k^{th} step) $\mathbf{u}(k) = \mathbf{h}(k)$ is used. In order to obtain the real actuator control \mathbf{u} equation (7) must be solved based on \mathbf{h} for \mathbf{u} . As the solution of this equation is not unique the pseudo-inverse operation [8] is applied. Inside this operation the solution with additional conditions can be found.

An example of such additional condition is the anti-backlash control. It consists of the condition that the control actions \mathbf{u} have the same sign during the whole motion, i.e. there is no transition across the zero line.

A graphical representation of Predictive Control is shown in Fig. 5.

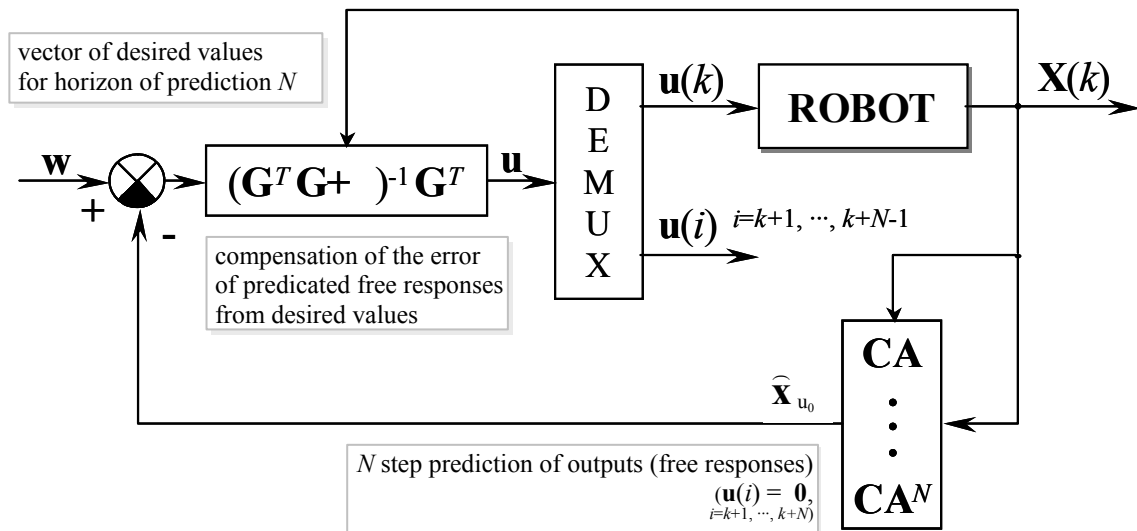


Fig. 5. Scheme of control circuit with Generalized Predictive Control (GPC) and robot.

The scheme generally represents both approaches, the classical (section 3.3) and the square root approach (section 3.4), where the final computation of the control actions in the drives is done by decomposition of \mathbf{u} within the block "DEMUX".

In both cases, the penalization λ controls magnitude of the redistributed loss on considered horizon of the prediction N . In the case of choice of penalization $\lambda = 0$, the system of equations (36) or (41) fragments to the one-step predictions assuming that desired values will be fulfilled just in one time step. In the real case, with changes of desired values, this assumption cannot be met. The loss, arising from model inaccuracies or from interactions with environment, is added to the new requirement on the movement of the system. Then, the new action is again designed on the assumption that the whole requirement will be performed during one time step. The different choice of penalization λ (mostly in a range $0 \div 1$) together with length of horizon N enables homogenous generation of control actions so that the available drives were not fitfully exerted. However, smooth changes of torques are achieved at cost of certain loss (error), theoretically equal to the value of the criterion.

4. EXAMPLES

This section shows different behavior of actuators for different control requirements, applied to planar redundant parallel robot for one selected trajectory (Fig. 6) described by kinematical quantities in Fig. 7.

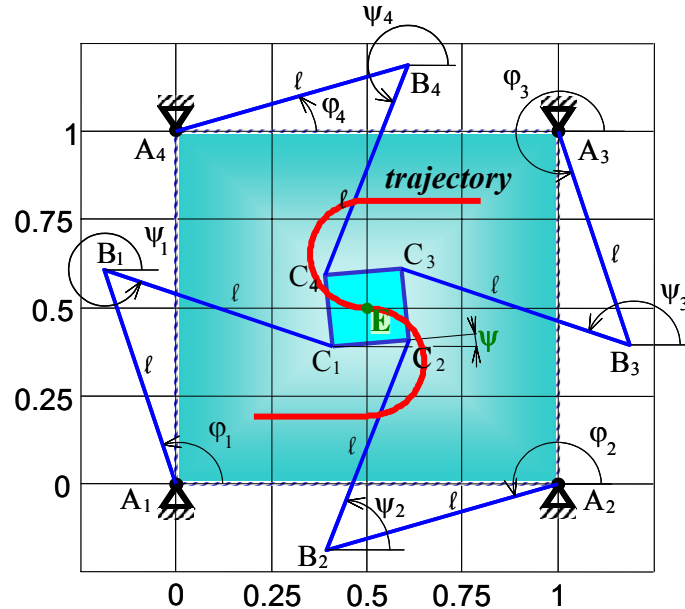


Fig. 6. Scheme of the robot and example of the trajectory (shape “S”).

The robot in Fig. 6 consists of a basic frame, four parallel groups of arms and a movable platform. The arms connect the movable platform with the basic frame. The whole construction enables movement of the platform in two directions (axes x and y) and its rotation, marked by angle ψ about the axis perpendicular to the basic frame.

From a mechanics point of view, this parallel structure has one drive and appropriate pair of arms being redundant (four inputs but three degrees of freedom only). The role of the redundancy is not strictly given. During robot movement, some configuration can lead to a singularity and the redundant elements may help to overcome this undesirable position.

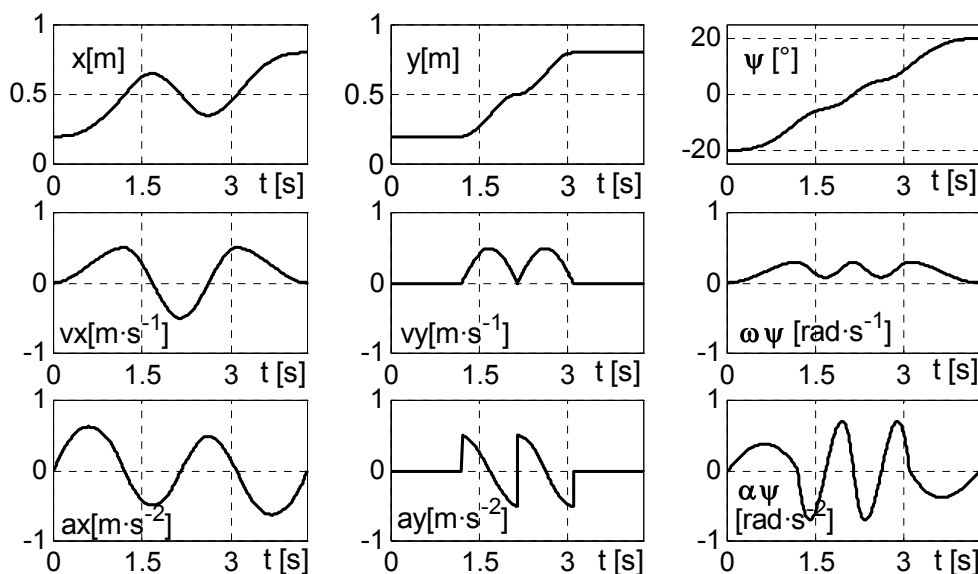


Fig. 7. Kinematical characterizations for trajectory in Fig. 6.

Figure 8 shows trends both without and with constraints. Rapid turns in the trends arise from change of trajectory character e.g. change from arc to abscissa segments.

The dotted lines represent accomplishment of anti-backlash condition. In this situation, the drives are continuously stretched and they generate actions only in one direction, i.e. they are either positive or negative. In spite of the higher input energy, besides solving backlash problem, the application of such actuators leads to higher stiffness. These actuators are designed according to solution of equations (36) or (7), where we search for solution of deficient rank system like minimal solution plus free variation arisen from redundancy or we apply quadratic programming to equation (36).

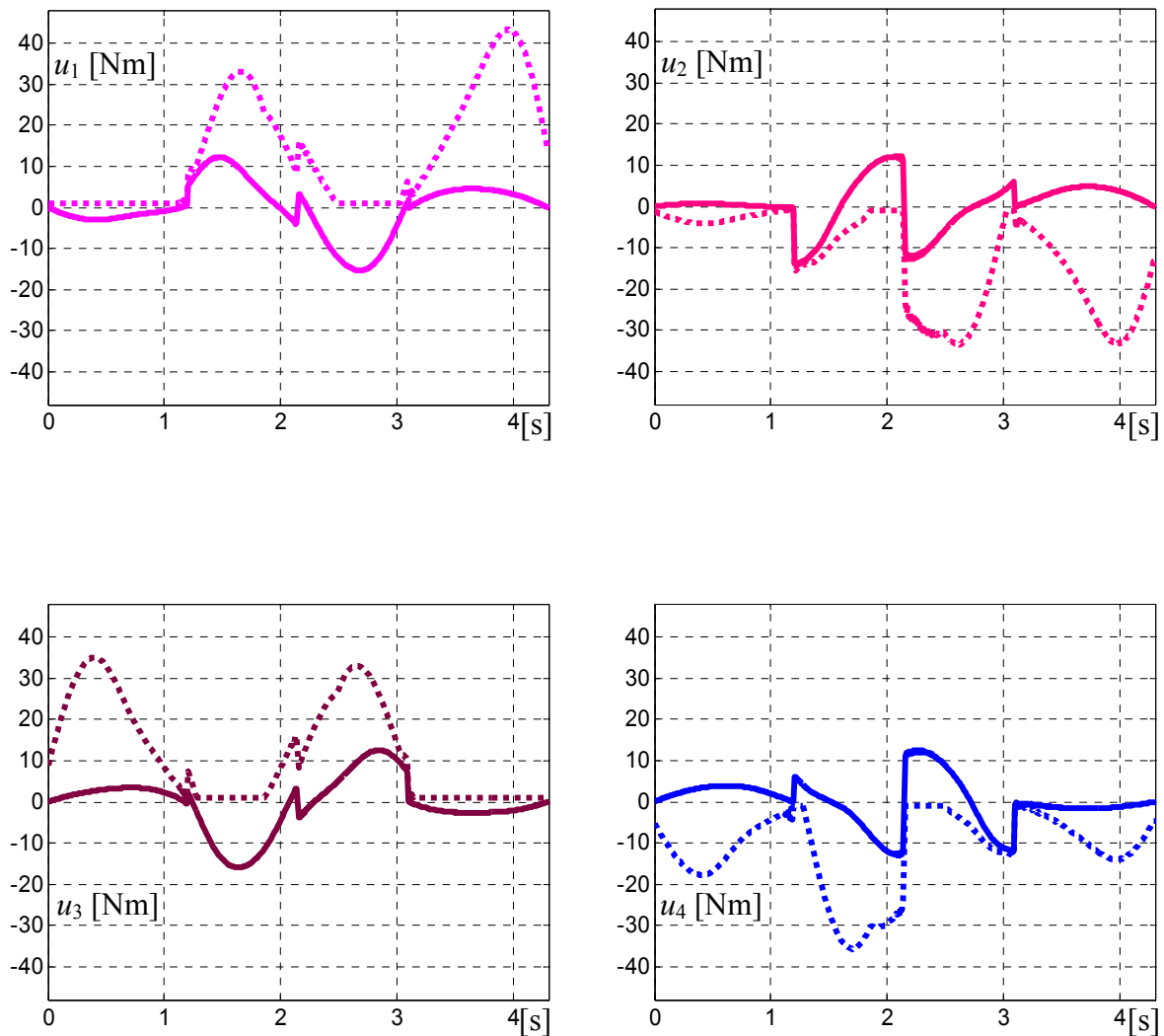


Fig. 8. Time history of four actuators: without any constrains (full lines); constrained case (dotted lines, accomplishment of antibacklash condition), (sampling 0.01s; max. error $1\mu\text{m}$; $\lambda=10^{-14}$; $N = 10$).

Figure 9 shows the solution of a different problem - removing of rapid turns of actuators. The solution uses the parameters of Predictive Control - penalization λ and horizon N . The higher value of penalization and horizon gives smoother actuator trends. Smoothing is useful for motion from point to point, where a final accurate position is required. During the motion, it leads to increase of error from the desired trajectory.

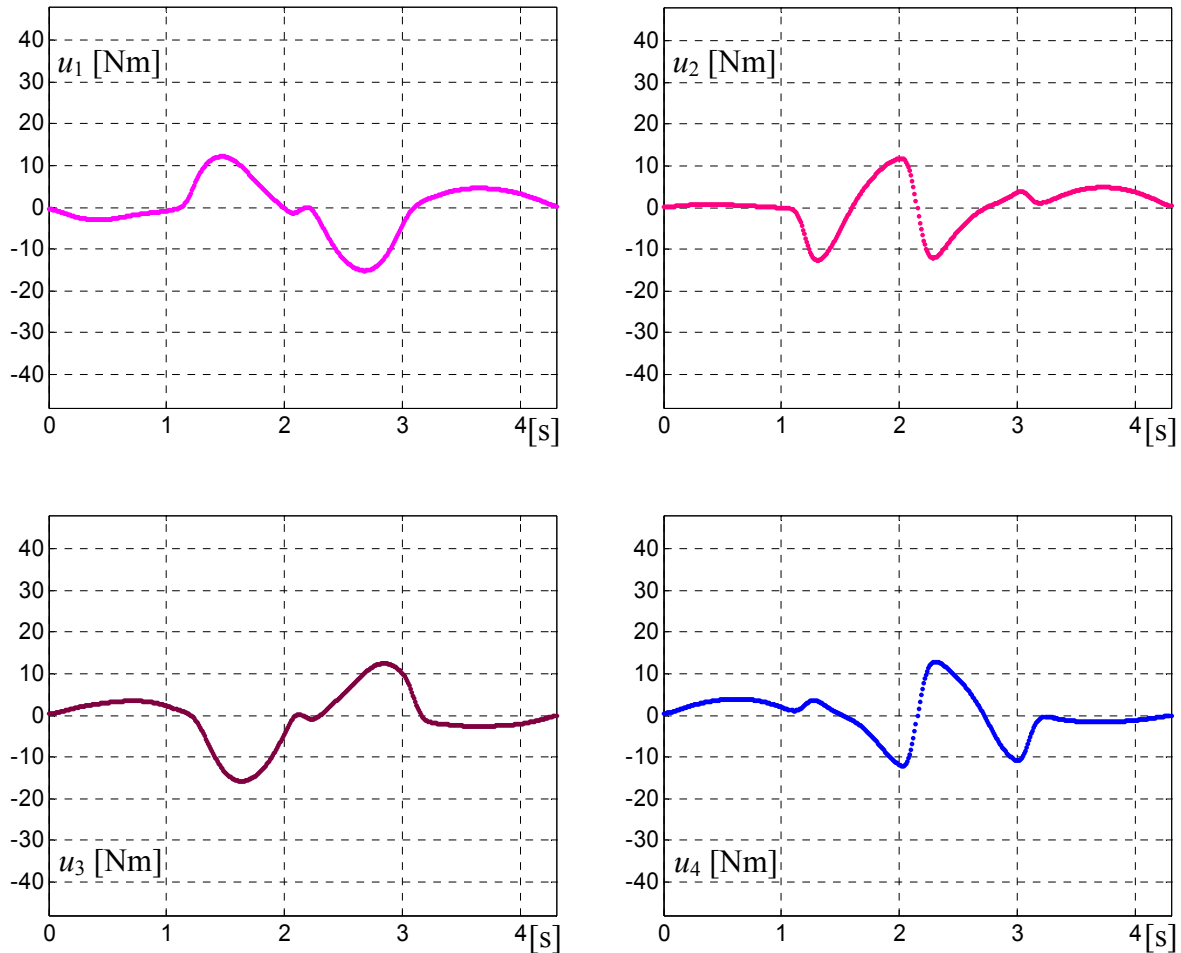


Fig. 9. Time history of four actuators; smoothing (sampling 0.01s; max. error 2.37 mm; $\lambda=5 \cdot 10^{-8}$; $N = 15$).

6. CONCLUSIONS

The redundant parallel structures represent promising solutions of industrial robots or machine tool applications. Their control is not a simple and fully investigated problem. The paper describes advanced high-level model-based design of General Predictive Control, which solves drive cooperation (removes possible drive fighting) in redundant parallel structures and which is able to match additional control requirements like anti-backlash condition for unique solution.

Two versions of GPC control (classical and square root) have been developed that produce the same values of actuator control, however the square root version is computationally more efficient and thus more suitable for real time application. These results have been successfully tested on a real laboratory model with the structure shown in Fig. 6.

ACKNOWLEDGEMENT

This research has been supported by grant (CTU 0204512, 2002), “The study of the properties of independent (decentralized) and centralized control of redundant parallel robots” and grant (GAČR 101/03/0602), “Redundant drives and measurement for hybrid machine tools”.

5. REFERENCES

1. Tsai, L.-W. In *Robot analysis - The Mechanics of Serial and Parallel Manipulators*, John Wiley & sons, Inc., New York, 1999.
2. Neugebauer, R. Eds. In *Development Methods and Application Experience of Parallel Kinematics*, Chemnitz, Germany, Apr 20-24, 2002; Verlag Wissenschaftliche Scripten.
3. Ordys, A.; Clarke, D. In *A state-space description for GPC controllers*, INT. J. Systems SCI., 1993, 24 (9), 1727-1744.
4. Soeterboek, R. In *Predictive Control - A Unified Approach*, Delft University of Technology: the Netherlands Rotterdam, 1990.
5. Böhm, J.; Belda, K.; Valášek, M. In *Study of control of planar redundant parallel robot*. Proceedings of the IASTED conference MIC 2001, Innsbruck, Austria, Feb 19-22, 2001; 694-699.
6. Böhm, J.; Belda, K.; Valášek, M. In *The antibacklash task in the path kontrol of redundant parallel robots*. Proceedings of the 13th International Conference on Process Control 01', Štrbské pleso, Slovakia, Jule 5-10, 2001.
7. Valášek, M.; Steinbauer, P. In *Nonlinear control of multibody systems*. Euromech 404, 1999; 437-444.
8. Lawson, Ch.; Hanson, R.J. In *Solving least squares problems*, Prentice Hall: New York, 1974.
9. Golub, H. G., Loan, Van Ch. F. In *Matrix computations*, The Johns Hopkins University Press, 1989.
10. Stejskal, V; Valášek, M In *Kinematics and Dynamics of Machinery*, Marcel Dekker, Inc.: New York, 1996.
11. Kock, S.; Schumacher, W. In *Control of a Fast Parallel Robot with a Redundant Chain and Gearboxes*, IEEE Conf. Robotics and Automation, 2000, 6 pp.
12. Norrlöf, M. In *Iterative Learning Control: analysis, design, and experiments*, Dissertations No.653, Sweden 2002, ISBN 91-7219-837-0.
13. Valášek, M.; Bauma, V.; Miláček, S.; Spurný, J.; Šika, Z.; Vampola, T. In *Synthesis of Redundant Parallel Robots*, Výpočtová mechanika, Nečtiny 2002, pp. 413-420.
14. Marquet, F.; Krut, S.; Company, O.; Pierrot, F. In *ARCHI: a New Redundant Parallel Mechanism-Modeling, Control and First Results*, LIRMM-UMR 5506, 1999.
15. Sciavicco, L.; Siciliano, B. In *Modeling and control of robot manipulators*, The McGraw-Hill Companies, Inc., New York, 1996.