DESIGN AND SIMULATION OF PREDICTIVE CONTROL OF DRIVES OF PLANAR REDUNDANT PARALLEL ROBOT

BÖHM JOSEF, BELDA KVETOSLAV, VALÁŠEK MICHAEL
Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic,
Pod vodárenskou věží 4 , 182 08 Praha 8 . Emails: bohm@utia.cas.cz , belda@utia.cas.cz

Abstract: Actual industrial robots and manipulators have a low operation speed. This is caused mainly by the maximum reachable acceleration for whole velocity interval. Parallel robots seem to be one of the promising ways for the solution to this problem. This paper deals with design and simulation of generalized predictive control (GPC) of these robots. The main reasons were that the model of the robot is well known, Quadratic criterion used in Predictive control for the optimalization can be easily modified to reflect various control requirements. And at the end the algorithmization is relatively simple and it can be easily implemented in the computer and used for control in real time.

Key words: Planar redundant parallel robot, Predictive control, Quadratic criterion, Drives.

1 INTRODUCTION

Actual industrial robots and manipulators have a low operation speed. This is caused mainly by the maximum reachable acceleration for the whole velocity interval. Parallel robots seem to be one of the promising ways for the solution to this problem. They have several advantages over actual robots:

- All or almost all drives are located on the basic frame i.e. drives don‘t move with robot and they have not any hold on the moving mass and stiffness.
- Truss construction of robots leads to higher stiffness than in serial ones.

On the other hand the parallel robots have also several disadvantages:

- The robot workspace is mostly smaller than for serial robots.
- The winding of robot platform is smaller than in the case of actual robot and it is asymmetric. Overrun of this constraint causes collision of arms with movable platform.

1.1 The description of a parallel robot

A parallel robot (Fig. 1) consists of the basic frame, which is at the same time workspace of robot, four independent drives, movable platform and eight arms, which connect movable platform with the basic frame. The arms can be divided into two following groups : four outside arms and four inside arms. Outside arms are interconnected with the drives on the basic frame and they are driving arms for the four inside arms. In each group there are two pair of parallel arms.

From mechanical point of view there are one drive and one outside and appropriate inside arm redundant, because generally the number of degrees of freedom of body in a plane is only three (movement in the direction of axis x, movement in the direction of axis y and angle of winding). Accordingly to control the robot, only three couples of outside and appropriate inside arms are necessary. But in this case singular position in workspace will appear. Therefore redundant drive was used. This drive helps to solve this problem of singularities [2]. And moreover it improves stiffness and rotation speed of platform.

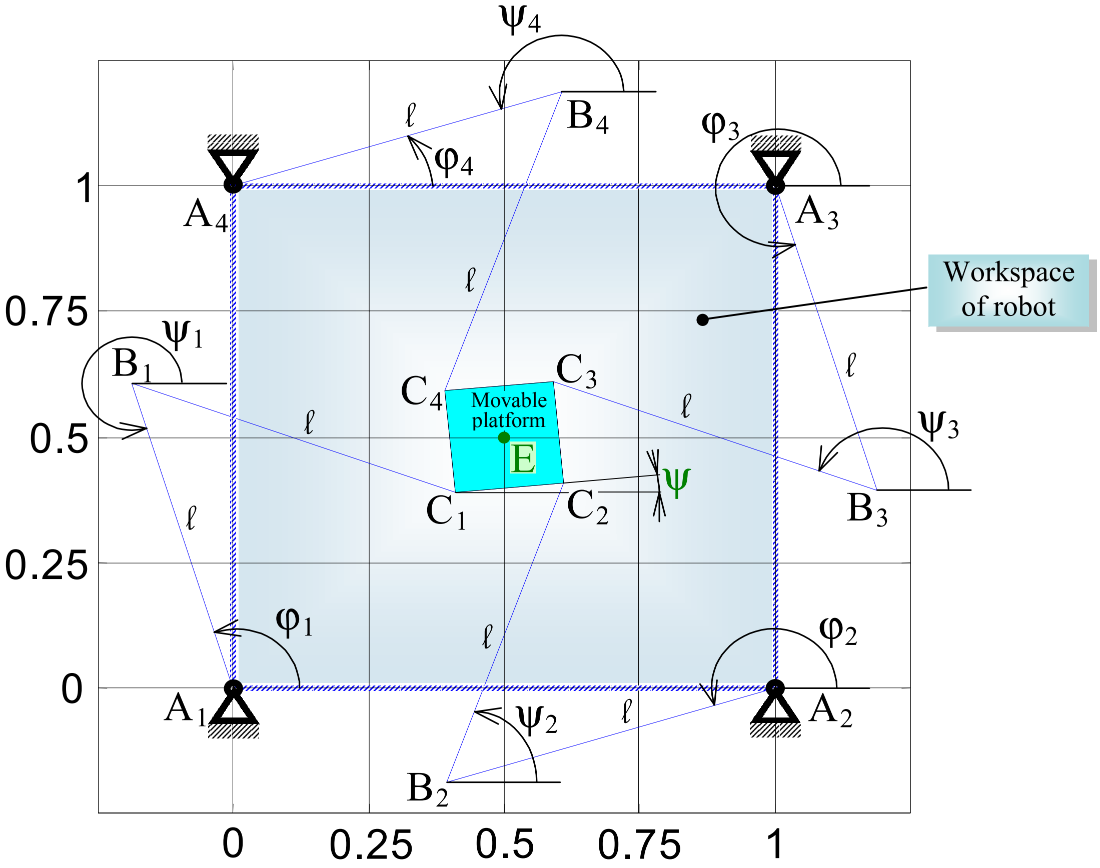

Fig. 1. Scheme of planar parallel robot with the most important geometrical description.
(The coordinates of center E (x_E , y_E) movable platform and its angle of winding ψ)

1.2 The description of the model of the planar redundant parallel robot

The principal task of robot is its movement along a planned trajectory. This motion can be unconstrained, if the forces (moments) are generated by the robot itself or constrained if contact forces appear from relationship between the robot and the environment. This paper deals with only unconstrained case. The above mentioned trajectory is primarily given in the coordinates system useful for the user of the robot. It is called an operational space and typically it can be a Cartesian coordinate system. The motion of the end effector is realized by corresponding motions of robot joints and so the motion can be also described by joints space system. The relation between these two spaces is given by a direct and inverse kinematics. The direct kinematics calculates operational space coordinates from a joins space coordinates and the inverse one does the opposite.

Similarly the control of robot can be described either in the joints space or in the operational space. The joins space design can be considered as a more straightforward. As the most simple one can consider the decentralized feedback control, where the drives are controlled independently and coupling forces are treated as disturbance. Feedforward compensation of various types helps significantly to improve the precision. Typical joints space control scheme is in [3]. Special role plays an inverse dynamics which calculates torques (forces) to a given coordinates and its derivatives and significantly simplifies the control problem. Continuous control design is usually used if the controller is realized by computer.

From the control point of view the robot can be considered as a multivariable plant described by the equation :

$$\mathbf{B}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u} \tag{1}$$

where **q** represents angles of robot's drives which belong to a joints space coordinates.

When operational space coordinates is used [2], equation (1) is transformed in the form :

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

Fortunately it can be written in state formula in this form [2] :

$$\begin{aligned} \dot{\mathbf{X}}(t) &= \mathbf{f}(\mathbf{X}) + \mathbf{g}(\mathbf{X})\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{h}\mathbf{X}(t) \end{aligned} \tag{3}$$

The input variables are torques of all drives moving outside arms. The state variables are composed of the independent coordinates, (i.e. the position of center x_E , y_E , angle of winding ψ movable platform) and their derivatives. The output variables are the independent coordinates.

The functions **f**(**X**), **g**(**X**) are highly nonlinear reflecting the kinematics structure of the parallel robot. The non-linearity stems from nonlinear dependence of the independent coordinates on the coordinates of drives, where torques are applied.

2 DESIGN OF PREDICTIVE CONTROL OF DRIVES

The Control should ensure the best possible compliance of planned trajectory with taking into account the maximum available torques of the drives and effective cooperation between the independent drives and the redundant drive. As solution Predictive Control has been chosen.

The main reasons are the following :

- The model of the robot is well known.
- The quadratic criterion used for the optimization can be easily modified to reflect various control requirements.
- The control is based on the local optimization, where the linarized equation can be used (i.e. it is used only the nearest future control signal).
- The incorporation of drives saturation to the design is simple.
- The approach admits combination of feedback / feedforward parts.
- The algorithmization is relatively simple and it can be easily implemented in computer and used for control in real time.

2.1 Criterion

As mentioned above there is used a quadratic criterion. Accordingly at first the nonlinear description of the robot must be transformed to a form, which can be used for optimization of the quadratic criterion. Strictly speaking the nonlinear model must be linearized [2] and converted from continuous to discrete time. We obtain discrete state formula in this form :

$$\begin{aligned} \mathbf{X}(i+1) &= \mathbf{A}\,\mathbf{X}(i)+\mathbf{B}\,\mathbf{u}(i) \\ \mathbf{y}(i) &= \mathbf{C}\,\mathbf{X}(i) \end{aligned} \tag{4}$$

where **X** is state vector with six elements (independent coordinates (i.e. the position of center x_E , y_E , angle of winding ψ movable part of robot) and their derivatives $\dot{x}_E$, $\dot{y}_E$,$\dot{\psi}$), **u** is vectors with four elements (i.e. torques of drives) and **y** is output vector with three elements (independent coordinates).

Now we can use the following quadratic criterion :

$$J_i = \mathcal{E}\left\{\sum_{k+1}^{k+N}\left[(y_j - w_j)^T (y_j - w_j)\right] + \sum_{k+1}^{k+N}\left[\lambda_j (u_{j-1})^2\right]\right\} \tag{5}$$

where $\mathcal{E}$ is operator of mean value, N is horizon of prediction, $\boldsymbol{y}$ is vector of outputs, $\boldsymbol{w}$ are requisite values, λ is penalization of input and $\boldsymbol{u}$ is vector of inputs in robot.

2.2 Optimization with using prediction to future (Predictive control)

Now we do derivate of the control law :

$$\begin{aligned}
\mathbf{y}(k) &= \mathbf{C}\;\mathbf{X}(k) \\
\mathbf{X}(k+1) &= \mathbf{A}\;\mathbf{X}(k) + \mathbf{B}\,\mathbf{u}(k) \\
\hat{\mathbf{y}}(k+1) &= \mathbf{CA}\;\mathbf{X}(k) + \mathbf{CB}\;\mathbf{u}(k) \\
\hat{\mathbf{X}}(k+2) &= \mathbf{A}^2\;\mathbf{X}(k) + \mathbf{A}\;\mathbf{B}\,\mathbf{u}(k) + \mathbf{B}\,\mathbf{u}(k+1) \\
\hat{\mathbf{y}}(k+2) &= \mathbf{CA}^2\mathbf{X}(k) + \mathbf{CAB}\;\mathbf{u}(k) + \mathbf{CB}\;\mathbf{u}(k+1) \\
\hat{\mathbf{X}}(k+3) &= \mathbf{A}^3\;\mathbf{X}(k) + \mathbf{A}^2\;\mathbf{B}\,\mathbf{u}(k) + \mathbf{A}\;\mathbf{B}\,\mathbf{u}(k+1) + \mathbf{B}\,\mathbf{u}(k+2) \\
\hat{\mathbf{y}}(k+3) &= \mathbf{CA}^3\;\mathbf{X}(k) + \mathbf{CA}^2\mathbf{B}\;\mathbf{u}(k) + \mathbf{CAB}\;\mathbf{u}(k+1) + \mathbf{CB}\,\mathbf{u}(k+2) \\
&\vdots \\
\hat{\mathbf{X}}(k+N) &= \mathbf{A}^N\;\mathbf{X}(k) + \mathbf{A}^{N-1}\,\mathbf{B}\,\mathbf{u}(k) + \mathbf{A}^{N-2}\,\mathbf{B}\,\mathbf{u}(k+1) + \cdots\cdots + \mathbf{B}\,\mathbf{u}(k+N-1) \\
\hat{\mathbf{y}}(k+N) &= \mathbf{CA}^N\,\mathbf{X}(k) + \mathbf{CA}^{N-1}\mathbf{B}\,\mathbf{u}(k) + \mathbf{CA}^{N-2}\mathbf{B}\,\mathbf{u}(k+1) + \cdots\cdots + \mathbf{CB}\,\mathbf{u}(k+N-1)
\end{aligned} \tag{6}$$

$\Rightarrow$ prediction of **y** can be obtained from $\hat{\mathbf{y}}_i = \mathbf{G}\,\mathbf{u}_i + \mathbf{f}_i$ (7)

where

$$\hat{\mathbf{y}}_i = [\hat{\mathbf{y}}_{k+1} \quad \hat{\mathbf{y}}_{k+2} \quad \cdots \quad \hat{\mathbf{y}}_{k+N}]^T \qquad \mathbf{G} = \begin{bmatrix} \mathbf{CB} & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & \ddots & \vdots \\ \vdots & & & \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix} \qquad \mathbf{f}_i = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^N \end{bmatrix} \mathbf{X}_k \qquad (8\,\mathrm{a\,b\,c\,d})$$

$$\mathbf{u}_i = [\mathbf{u}_k \quad \mathbf{u}_{k+1} \quad \cdots \quad \mathbf{u}_{k+N-1}]^T$$

This is the base of predictive control – new unknown values are obtained from actual values.

Now we can optimize criterion, which is changed, because we don't use factual values, but only their predictions.

$$J_i = \mathcal{E}\left\{(\hat{\mathbf{y}}_i - \mathbf{w}_i)^T(\hat{\mathbf{y}}_i - \mathbf{w}_i) + \mathbf{u}_i^T \lambda\, \mathbf{u}_i\right\} = \mathcal{E}\left\{(\mathbf{G}\,\mathbf{u}_i + \mathbf{f}_i - \mathbf{w}_i)^T(\mathbf{G}\,\mathbf{u}_i + \mathbf{f}_i - \mathbf{w}_i) + \mathbf{u}_i^T \lambda\, \mathbf{u}_i\right\} =$$

$$= \mathcal{E}\left\{\left(\mathbf{u}_i^T\mathbf{G}^T + \mathbf{f}_i^T - \mathbf{w}_i^T\right)\left(\mathbf{G}\,\mathbf{u}_i + \mathbf{f}_i - \mathbf{w}_i\right) + \mathbf{u}_i^T \lambda\, \mathbf{u}_i\right\} =$$

On the assumption that the model of robot is well known (σy=0) we can omit operator of mean.

$$= \mathbf{u}_i^T\mathbf{G}^T\mathbf{G}\,\mathbf{u}_i + \mathbf{u}_i^T\mathbf{G}^T\mathbf{f}_i - \mathbf{u}_i^T\mathbf{G}^T\mathbf{w}_i + \mathbf{f}_i^T\mathbf{G}\,\mathbf{u}_i + \mathbf{f}_i^T\mathbf{f}_i - \mathbf{f}_i^T\mathbf{w}_i - \mathbf{w}_i^T\mathbf{G}\,\mathbf{u}_i - \mathbf{w}_i^T\mathbf{f}_i + \mathbf{w}_i^T\mathbf{w}_i + \mathbf{u}_i^T\lambda\,\mathbf{u}_i = \tag{9}$$

$$= \mathbf{u}_i^T\left(\mathbf{G}^T\mathbf{G} + \lambda\right)\mathbf{u}_i + \mathbf{u}_i^T\mathbf{G}^T(\mathbf{f}_i - \mathbf{w}_i) + \left(\mathbf{f}_i^T - \mathbf{w}_i^T\right)\mathbf{G}\,\mathbf{u}_i + \mathbf{f}_i^T\mathbf{f}_i - \mathbf{f}_i^T\mathbf{w}_i - \mathbf{w}_i^T\mathbf{f}_i + \mathbf{w}_i^T\mathbf{w}_i$$

Condition is $J_i \overset{!}{=} \min$ $\quad\Rightarrow\quad \dfrac{\partial J_i}{\partial \mathbf{u}_i} = \left(\mathbf{G}^T\mathbf{G} + \lambda\right)\mathbf{u}_i + \mathbf{G}^T(\mathbf{f}_i - \mathbf{w}_i) \overset{!}{=} 0$ (10)

$$\Rightarrow \qquad \mathbf{u}_i = \left(\mathbf{G}^T\mathbf{G} + \lambda\right)^{-1}\mathbf{G}^T(\mathbf{w}_i - \mathbf{f}_i) \tag{11}$$

And obtained control law in factor form :

$$\begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_{k+N-1} \end{bmatrix} = \left(\begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \cdots & \mathbf{CA}^{N-1}\mathbf{B} \\ 0 & \mathbf{CB} & & \mathbf{CA}^{N-2}\mathbf{B} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots\cdots & & \mathbf{CB} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{CB} & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix} + \begin{bmatrix} \lambda & 0 & \cdots & 0 \\ 0 & \lambda & & \vdots \\ \vdots & & \lambda & 0 \\ 0 & \cdots & 0 & \lambda \end{bmatrix} \right)^{-1} \cdot$$

$$\cdot \begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \cdots & \mathbf{CA}^{N-1}\mathbf{B} \\ 0 & \mathbf{CB} & & \mathbf{CA}^{N-2}\mathbf{B} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots\cdots & & \mathbf{CB} \end{bmatrix} \cdot \left(\begin{bmatrix} \mathbf{w}_{k+1} \\ \mathbf{w}_{k+2} \\ \vdots \\ \mathbf{w}_{k+N} \end{bmatrix} - \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^N \end{bmatrix} \cdot \mathbf{X}_k \right) \tag{12}$$

Now this control law can be used for simulation. It is certainly used only first element $\mathbf{u}_k$ from vector $\mathbf{u}_i$. In case, if penalization is zero or near zero, matrix **G'·G** is badly conditional (one drive is redundant). This problem is solved by orthogonal-triangular decomposition. Considering constraints is solved by the quadratic programming.

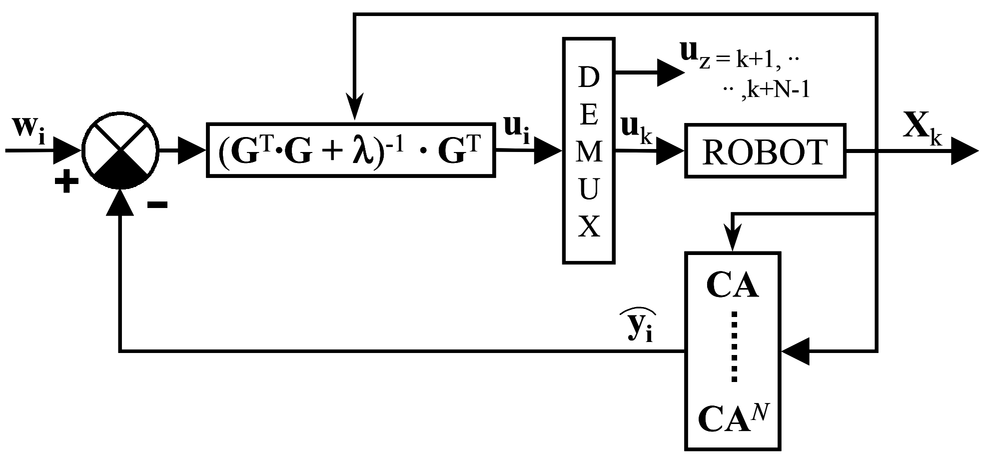

Fig. 2. Appropriate control circuit with parallel robot.

3 SIMULATION OF PLANAR REDUNDANT PARALLEL ROBOT

For simulation of robot some plan of trajectory must be prepared, which must be available for the robot. For example of our simulation there is chosen trajectory composed of bisector segments and arc segments. Trajectory was time-parameterized with constant period where this constant period may be chosen. (Note: It is supposed that sampling period is longer than in a continuous design [3] » less computation time.) At planning trajectory we have the consider kinematics laws.

3.1 Simulation examples and results

There is one chosen simulation : (set parameters : discrete period Ts=0.01s, penalization of actuator $\lambda = 10^{-10}$, contraction of torques $\mathbf{u}_{max} = 15$ N·m, and horizon of prediction $N_1 = 10$).

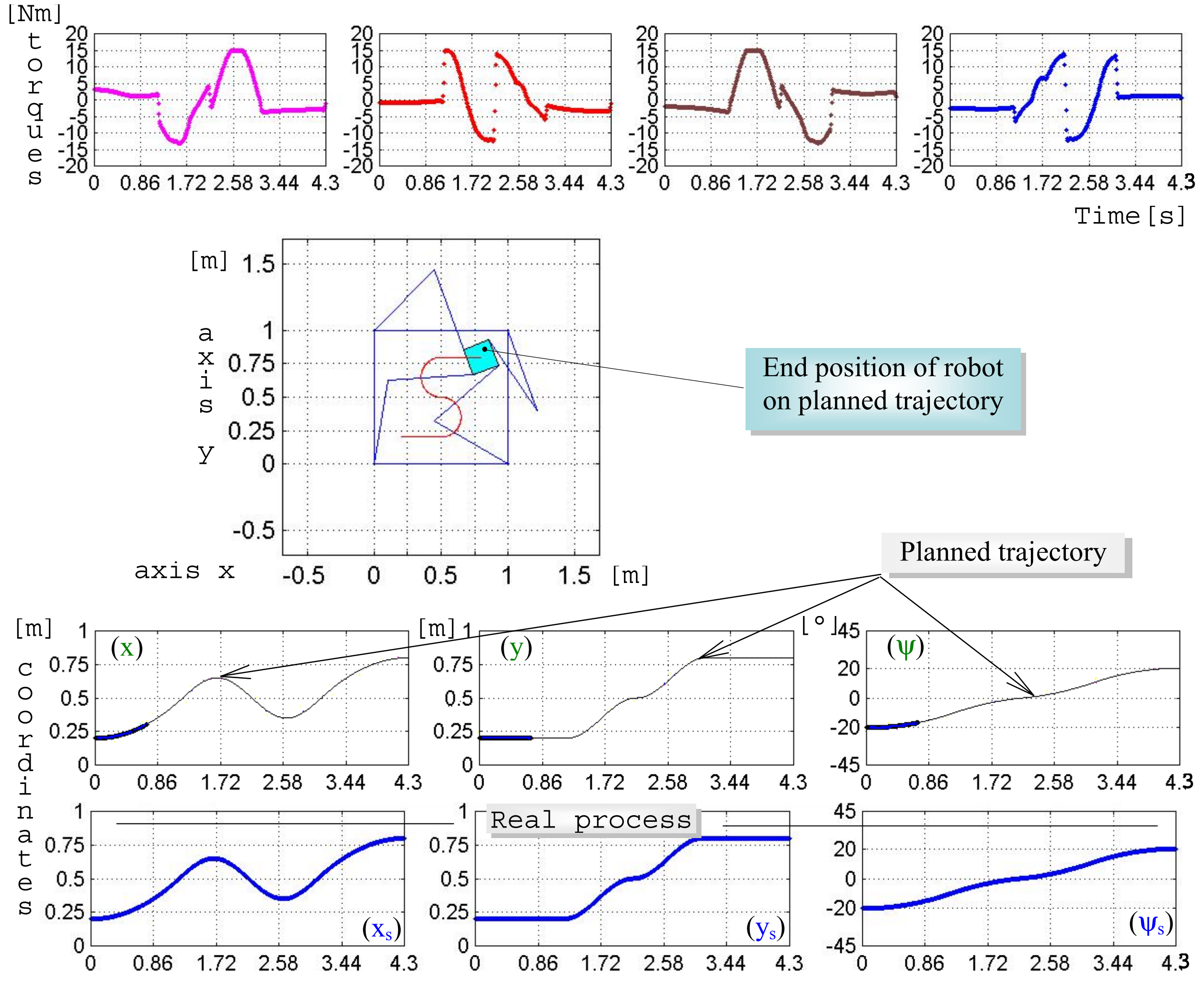

Fig. 3. Process of four inputs, preview of robot and process of three outputs.

Here chosen trajectory is relatively fast. Evidently compliance of requirements is better for the slow trajectory. But our simulation example shows (Fig. 3) that Predictive control can be used even for faster trajectory.

The setting parameters for control of our model parallel robot was appeared as follow : Penalization in a range ($10^{-8} \div 10^{-10}$). High penalization ($1 \div 10^{-6}$) causes, that robot isn't controlled and on other hand low penalization ($10^{-12} \div 0$) causes oscillation of the actuators. Horizon of prediction should be as the shortest (5÷15), because long horizon causes slow computation and it doesn't already contribute to improvement control. Sampling period is chosen in relationship to planning trajectory. For faster trajectory it must be chosen about 1÷10 ms and for slow trajectory it is chosen about 0.1s.

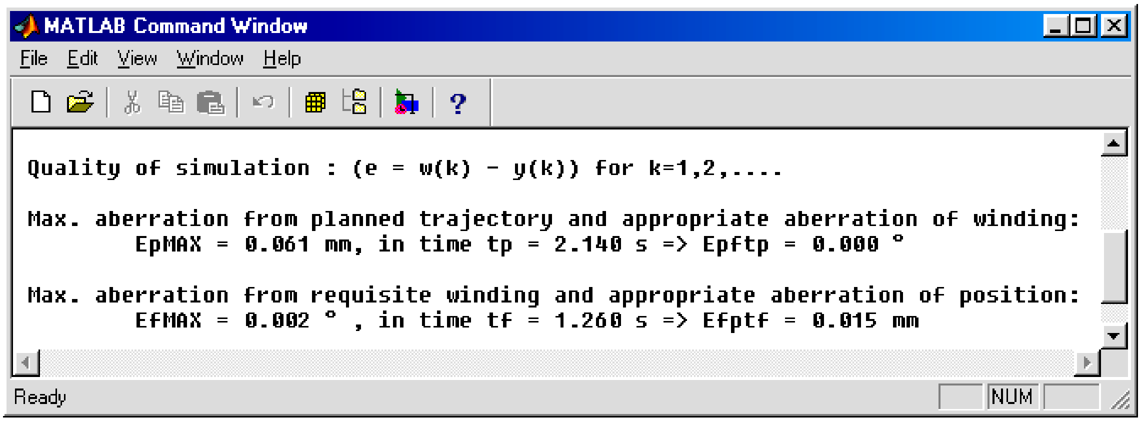

Fig. 4 Quality test of simulation above.

Fig. 4 shows the maximum aberrations from planned trajectory and time in which they were registered. (Note: Planning of trajectory and simulation of movement of robot was realized under MATLAB – SIMULINK Release 11.)

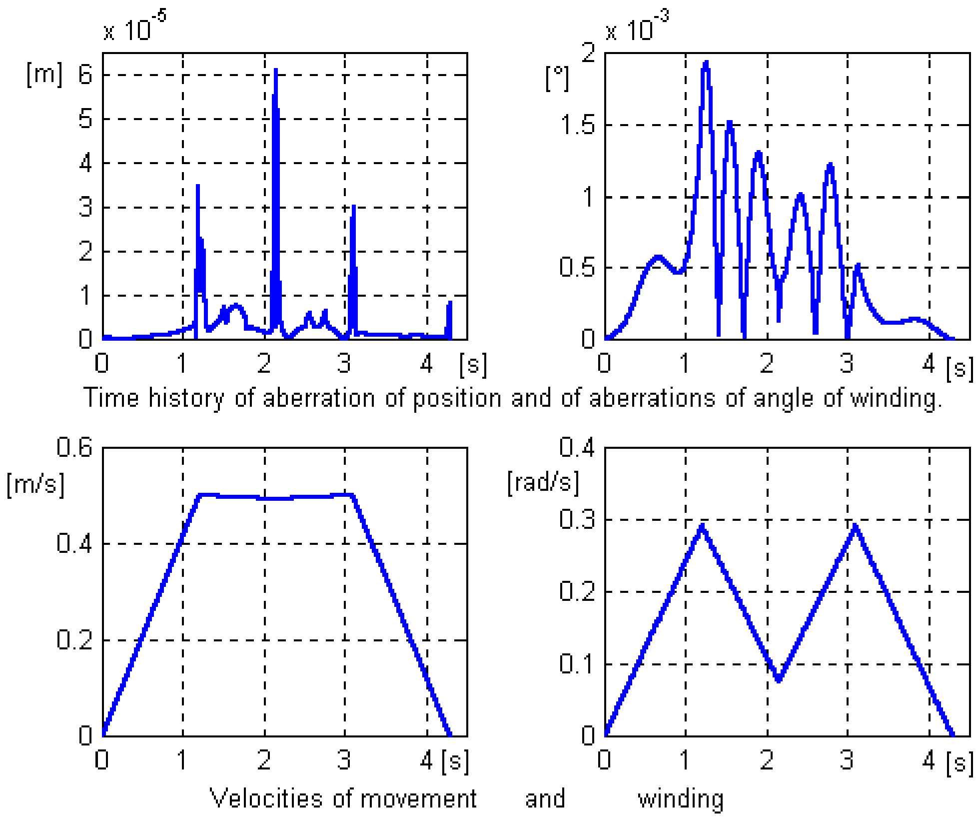

Fig. 5. Time history of Aberrations and velocities.

4 CONCLUSION

The approach indicated here (Predictive control) is certainly way to control given redundant parallel robot. The results of experiments with a computer model have shown that the technique used could be applied in practice. The tuning of parameters in a controller synthesis was not critical. The main concern was to test how the controller will cope with the non-linearity of the plant and with the redundancy of drives.

References

1. ORDYS, A. – CLARKE, D.: A state–space description for GPC controllers. INT. J. Systems SCI., 1993, vol. 24, no.9, pp. 1727 – 1744.
2. VALÁŠEK, M. – STEINBAUER, P.: Nonlinear control of multibody systems. Euromech 99, p.447.
3. SCIAVICCO, L. – SICILIANO, B.: Modeling and control of robot manipulators. McGraw-Hill Co., New York 1996.

Supported by Czech Grant Agency under the contract (GA ČR 101/99/0729, 1999-2001) :
” Redundant parallel robot and its control “.