The Direct Kinematics for Path Control of Redundant Parallel Robots

JOSEF BÖHM, KVĚTOSLAV BELDA, MICHAEL VALÁŠEK
Department of Adaptive systems,
Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic.
Pod vodárenskou věží 4, 182 08 Praha 8
CZECH REPUBLIC
bohm@utia.cas.cz, belda@utia.cas.cz, valasek@fsik.cvut.cz

Abstract: - The parallel robots seem to be one of the promising ways to improve accuracy and speed. At their development some new problems appear. This paper deals with design of direct kinematics for real-time path control of planar redundant parallel robots. The main reason for its using is a fact that the direct kinematics gives possibility to use Cartesian coordinates as against joint ones and by this considerably simplifies model of the robot and consecutively computation of high level control based on knowledge of such model. As a subtask of design of the direct kinematics, the trajectory planning is discussed.

Key-Words: - Direct Kinematics, Planar redundant parallel robot, Nonlinear system, DAE-ODE equations, Trajectory.

1 Introduction

The most topical industrial robots and manipulators do not cope with increasing requirements on speed and accuracy. Therefore, new approaches of their construction are being found. Parallel robots, especially redundantly actuated, seem to be one of the promising ways to solve these requirements [4]. And, moreover, they have several advantages over traditional serial robots.

The main is the following:

- All or almost all drives are located on the basic frame and truss construction of the robot leads to higher stiffness than in serial types. It is advantageous for accurate machining and positioning.

On the other hand the parallel robots have one constrain:

- That is given in more possibility of arms collision. But this can be solved if this constraint would be taken into account at the planning of desired trajectory. However, this disadvantage does not markedly keep down movement of the robot.

As an example, let us consider one such redundantly actuated planar parallel robot (Fig.1). It consists of the basic frame, which at the same time encloses workspace of the robot, four independent drives, movable platform and eight arms, which connect the movable platform with the basic frame. The arms are parallelly situated.

From the mechanical point of view, this robot has one drive and one pair of arms redundant, because generally the number of degrees of freedom of body in a plane is only three. Accordingly, for control of the robot and for its mechanical determination, only three pairs of appropriate arms are necessary. But in this case, the singular position in the workspace will appear. Therefore the redundant drive is used in order to overcome the problem. And, moreover, it improves stiffness and rotation speed of movable platform and gives the possibility to comply with the other additional control requirements.

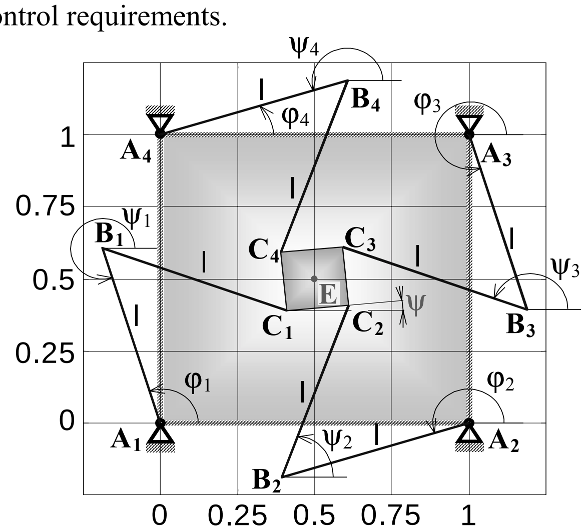

Fig. 1. Scheme of planar parallel robot with the most important geometrical description (The coordinates of center E of movable platform and its angle of winding ψ).

The aim of this paper is investigation of the direct kinematics for real-time control of redundant parallel structure of the robot at using of the specially planned trajectory.

2 Problem formulation

From the high-level control design point of view, the principal task is a choice of suitable model of the robot. On it the design of control depends.

The robot-manipulator is a multibody system, which can be described by Lagrange's equations of mixed type. These equations lead to the differential-algebraic equations (DAE) in the following form:

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

where $\mathbf{M}$ is a mass matrix, $\mathbf{s}$ is a vector of physical coordinates (their number is higher than the number of degrees of freedom), $\mathbf{\Phi}_\mathbf{s}$ is a Jacobian of the system, $\boldsymbol{\lambda}$ are Lagrange's multipliers, $\mathbf{g}$ is a vector of right sides, matrix $\mathbf{T}$ transforms the inputs $\mathbf{u}$ (four torques) into four drives and $\mathbf{f}(\mathbf{s}(t)) = \mathbf{0}$ represents geometrical constrains.

In our case the one possibility exists for the transformation into independent coordinates, which, in this case, may be chosen as Cartesian coordinates of the center. Which is very suitable because DAE model is transformed to ODE model [2]. It means that Lagrange's multiplies disappear and moreover we obtain more transparent relationship between central working position and inputs-torques to the robot.

Then the resulting model of the robot is following:

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

This model can be generally rewritten in the state formula in following form:

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

where the input variables are torques of all drives. The state variables are coordinates of center $\mathbf{E}$ (x, y) of movable platform, its angle of winding (ψ) and their derivative. The output variables are only operational coordinates selected from state variables.

The functions $\mathbf{f}(\mathbf{X})$, $\mathbf{g}(\mathbf{X})$ are highly nonlinear reflecting the kinematic structure of the parallel robot. The non-linearity stems from the nonlinear dependence of the operational coordinates on the joint coordinates.

In order to use model of the robot described by eq. (2), we must provide the availability of independent coordinates i.e. coordinates of the center (x, y, ψ) of which number is equal to the number of degrees of freedom. This problem is solved by the direct kinematics, which recomputes joint-drives coordinate to Cartesian coordinates of movable platform center. Before introduction of possible approaches to the direct kinematics some plan of trajectory must be prepared.

3 Planning of the trajectory

As opposed to classical serial robots the planning trajectory for parallel robots demands certain harmonization of their movement. It is caused by interconnection of arms through the movable platform vide Fig.2.

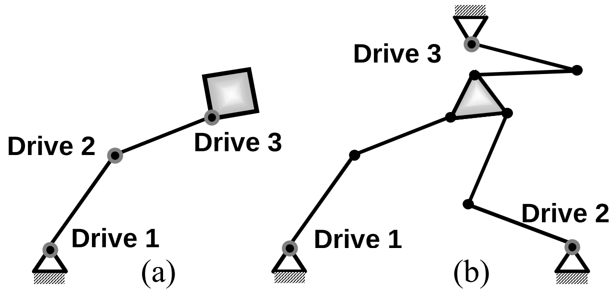

Fig.2. Classical (a) and parallel (b) type of the robot.

The winding of all drives at the parallel robot gives simultaneously position and winding of the movable platform against classical serial robots, where angle of winding is not strictly dependent on all drives, but it can be provided by last drive (e.g. Fig.2.(a), Drive 3) in the chain.

The trajectory is usually given by technological requirements. And for control, it must be time parameterized as discrete set of ordered pairs of time and Cartesian coordinates

$$[\mathrm{t}, \mathbf{X} = [x\; y\; \psi\; \dot{x}\; \dot{y}\; \dot{\psi}]]\big|_{\mathrm{t=k\cdot Ts}}$$

at sampling period Ts. In majority cases the whole trajectory is composed of simple segments, like bisectors or arcs segments, which are simply mathematically described. In remaining cases the group of points is given and this leads on approximation like smoothing problem. With using parametric interpolation or approximation curves e.g. like Ferguson's cubic, Bezier's curves or B-spline curves, this problem is transformed on the former cases.

The general requirement on planning of trajectory is position of tool and its angle of winding in certain time. Cartesian coordinates and required velocities give this. Note that the position and angle is identical with position and angle of winding of movable platform.

The design of the trajectory is based on simple kinematic laws. At the first, the distance-length of segment and angle of winding, which is performed together, must be counted. Generally it is given by expressions

$$s = \int_s ds \qquad\qquad \Psi = \Psi_{final} - \Psi_{inital} \tag{4 a,b}$$

where ds is an element of segment of the trajectory.

At using expressions for accelerations in a form:

$$a = \frac{dv}{dt}, \qquad \alpha = \frac{d\omega}{dt} \qquad (5\text{ a,b})$$

and their double integration in frame of one segment we obtain expressions for orientational working times:

$$t_1 = \frac{2\,s}{v_{inital} + v_{final}} \quad t_2 = \frac{2\,\psi}{\omega_{inital} + \omega_{final}} \qquad (6\text{ a,b})$$

From them the higher, labeled as t_{final}, is chosen and rounded to the nearest value, which is multiple of sampling period Ts. That provides sufficient time for performing of movement.

Now the own parameterization can be made. For the smooth connecting and the accomplishment of simultaneous movement and rotation of the movable platform, the trajectory should have the first derivation continuous and smooth and at the same time the second derivation should be continuous and segmentally smooth curve with zero border values. In order to perform these requirements it is suitable to prescribe the equation of acceleration as following:

$$a = a_0 + a_1 t + a_2 t^2 \qquad (7)$$

consecutively velocity and position in the form:

$$v = v_{inital} + a_0 t + \frac{1}{2} a_1 t^2 + \frac{1}{3} a_2 t^3 \qquad (8)$$

$$s = v_{inital} t + \frac{1}{2} a_0 t^2 + \frac{1}{6} a_1 t^3 + \frac{1}{12} a_2 t^4 \qquad (9)$$

with conditions of initial and final state (as spoken above) in the form:

for $t = 0$: $v(0) = v_{inital}$; $s(0) = 0$; $a(0) = 0$,
for $t = t_{final}$: $v(t_{final}) = v_{final}$;
$s(t_{final}) = s_{final}$; $a(t_{final}) = 0.$ (10)

From eq. (7) (8) (9) and conditions (10) the parameters a_0, a_1, a_2 can be obtained.

If we used the similar equations for rotation

$$\alpha = \alpha_0 + \alpha_1 t + \alpha_2 t^2 \qquad (11)$$

$$\omega = \omega_{inital} + \alpha_0 t + \frac{1}{2} \alpha_1 t^2 + \frac{1}{3} \alpha_2 t^3 \qquad (12)$$

$$\psi = \omega_{inital} t + \frac{1}{2} \alpha_0 t^2 + \frac{1}{6} \alpha_1 t^3 + \frac{1}{12} \alpha_2 t^4 \qquad (13)$$

and conditions in the form:

for $t = 0$: $\omega(0) = \omega_{inital}$; $\psi(0) = 0$; $\alpha(0)=0$.
for $t = t_{final}$: $\omega(t_{final}) = \omega_{final}$;
$\psi(t_{final}) = \psi$; $\alpha(t_{final}) = 0.$ (14)

then we obtain also the parameters for winding.

Now if we prepare time vector as following

$$T=[0,\ Ts,\ 2Ts \cdots kTs] \quad \text{for integer } k = \frac{t_{final}}{Ts} \qquad (15)$$

and substitute it into eq. (7)-(9) and eq. (11)-(14) we obtain consequence of positions, velocities and accelerations, which can be directly decomposed in directions x,y (Fig.3 and Fig.4). The equations (7)-(9) and eq. (11)-(14) can be used for computation of the other geometrical parameterization [1]. Then required ordered pairs of time and coordinates with their derivatives are obtained.

This time parameterization of the trajectory can be already used for control, even for the design and testing of the direct kinematics, which is discussed in the following section.

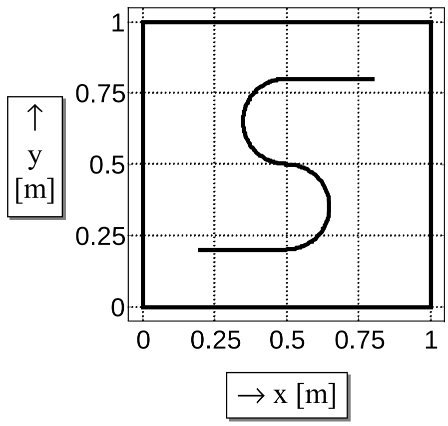

Fig.3. Example of one trajectory, composed of bisector and arc segments, for the control of the parallel robots.

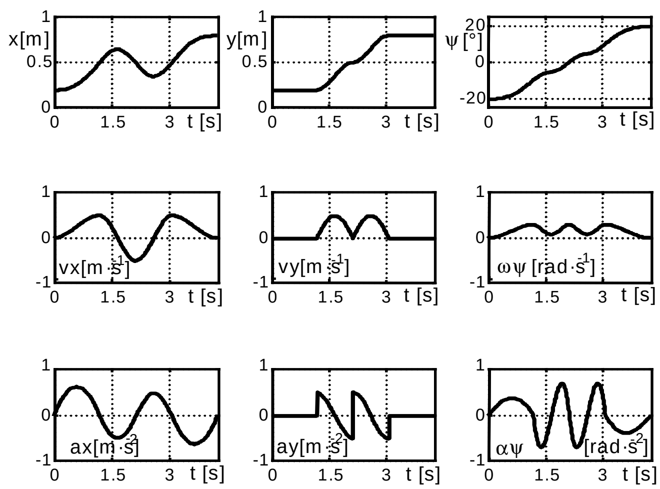

Fig.4. Kinematic component characterizations of the trajectory from Fig. 4. (positions [x, y, ψ], velocities and accelerations).

4 Direct kinematics

The coordinates appearing in the branch of robots and manipulators may be divided into drive coordinates $\mathbf{q}_1$, operational $\mathbf{x}$ and other ancillary coordinates $\mathbf{q}_2$. All these coordinates are either independent (their number equals number of degrees of freedom) or dependent. Between them there is relations generally expressed by system of nonlinear equations:

$$f(\mathbf{x}, \mathbf{q}_1, \mathbf{q}_2) = \mathbf{0} \qquad (16)$$

Direct kinematics solves problem of recomputing the drives coordinates $\mathbf{q}_1$ on operational, in our case (Fig.1), independent coordinates $\mathbf{x}$. In comparison with the classical robots, where it is not difficult, the direct kinematics of the parallel robots especially redundantly actuated is not simple task. Then, we find the function $\mathbf{x} = f(\mathbf{q}_1)$, which unfortunately is not analytically solved. We have several possibilities, how to solve this. Either we can use classical numerical solution or engineering solution in the form of control task.

4.1 Numerical solution

Now we briefly describe usual numerical solution with the Newton's method, which is suitable.

The method is based on the Taylor expansion to the first order in the neighbourhood of initial value $\mathbf{Z}^{(0)} = [\mathbf{x}^{(0)}; \mathbf{q}_2^{(0)}]$, which can be equal to desired values.

$$\mathbf{f}(\mathbf{Z}^*, \mathbf{q}_1) = \mathbf{f}(\mathbf{Z}^{(k)}, \mathbf{q}_1) + \frac{\partial \mathbf{f}(\mathbf{Z}^{(k)}, \mathbf{q}_1)}{\partial \mathbf{Z}} \Delta \mathbf{Z}^{(k)} = 0 \quad (17)$$

From this we obtain a system of linear equations for $\Delta\mathbf{Z}^{(k)}$

$$\frac{\partial \mathbf{f}(\mathbf{Z}^{(k)}, \mathbf{q}_1)}{\partial \mathbf{Z}} \Delta \mathbf{Z}^{(k)} = -\mathbf{f}(\mathbf{Z}^{(k)}, \mathbf{q}_1) \qquad (18)$$

then $k+1$ iteration is $\mathbf{Z}^{(k+1)} = \mathbf{Z}^{(k)} + \Delta \mathbf{Z}^{(k)}$ (19)

This we can substitute into eq. (18) and repeat procedure until $\left|\Delta \mathbf{Z}^{(k)}\right| \le \varepsilon$, where ε is such difference given before, which is already not critical for control.

Consecutively the velocities and accelerations are given by

$$\frac{d}{dt}\left(\mathbf{f}(\mathbf{Z}, \mathbf{q}_1) = 0\right)$$

$$\frac{\partial \mathbf{f}(\mathbf{Z}, \mathbf{q}_1)}{\partial \mathbf{Z}} \dot{\mathbf{Z}} + \frac{\partial \mathbf{f}(\mathbf{Z}, \mathbf{q}_1)}{\partial \mathbf{q}_1} \dot{\mathbf{q}}_1 = 0 \;\Rightarrow\; \dot{\mathbf{Z}}$$

$$\Phi_Z \quad \dot{\mathbf{Z}} + \quad \Phi_{q_1} \quad \dot{\mathbf{q}}_1 = 0 \;\Rightarrow\; \dot{\mathbf{Z}} \qquad (20)$$

$$\frac{d}{dt}\left(\mathbf{\Phi}_Z \dot{\mathbf{Z}} + \mathbf{\Phi}_{q_1} \dot{\mathbf{q}}_1 = 0\right)$$

$$\dot{\Phi}_Z \dot{\mathbf{Z}} + \Phi_Z \ddot{\mathbf{Z}} + \dot{\Phi}_{q_1} \dot{\mathbf{q}}_1 + \Phi_{q_1} \ddot{\mathbf{q}}_1 = 0 \;\Rightarrow\; \ddot{\mathbf{Z}} \qquad (21)$$

With using the previous results of eq. (19) for eq (20) and eq. (19) and (20) for eq (21), the eq. (20) and (21) represent systems of linear algebraic equations. From them the operational coordinates arise.

4.2 Solution in the form of control task

As has been mentioned, the direct kinematics of the parallel robots is more complicated and it has not direct analytical solution as at classical kinematic structures, where the direct kinematics is simple and conversely there is a problem with kinematics inversion there. When we consider the previous numerical solution, we can see, that after derivation of system equations (16), we obtain linear relations. This fact we can use. If we have possibility to extract the relation only between independent operational coordinates $\mathbf{x}$ and drive coordinates $\mathbf{q}_1$:

$$\bar{f}(\mathbf{x}, \mathbf{q}_1) = \mathbf{0} \quad \Rightarrow \quad \mathbf{q}_1 = f_1(\mathbf{x}) \qquad (22)$$

and derivate it according to time we obtain system of linear differential kinematic equations

$$\dot{\mathbf{q}}_1 = \frac{df_1}{d\mathbf{x}} \dot{\mathbf{x}}$$

$$\dot{\mathbf{q}}_1 = \mathbf{J}_1 \dot{\mathbf{x}} \qquad (23)$$

where $\mathbf{J}_1$ is Jacobian. Then eq. (23) makes the basis of two following approaches.

The first provides the design of the direct kinematics by feedforward control scheme with using desired values and later deals with feedback scheme independent on desired values.

4.2.1 Feedforward direct kinematics

Suppose that desired values $\mathbf{x}_d$ of the trajectory are available and the same may be said about initial conditions on position and angle of winding.

By considering eq. (23) with regular square matrix $\mathbf{J}_1$ the operational coordinates can be obtained via simple inversion

$$\dot{\mathbf{x}} = (\mathbf{J}_1)^{-1} \dot{\mathbf{q}}_1 \qquad (24)$$

In the case, when parallel robot is redundant (our case Fig.1) the Jacobian has more rows than columns and it can't be inverted. We can use left pseudo-inverse.

$$\dot{\mathbf{x}} = \left(\mathbf{J}_1^{\mathrm{T}} \mathbf{J}_1\right)^{-1} \mathbf{J}_1^{\mathrm{T}} \dot{\mathbf{q}}_1 \qquad (25)$$

The Jacobian $\mathbf{J}_1$ is a function of operational coordinates $\mathbf{x}_d$ and if difference $\mathbf{x}_d$-$\mathbf{x}$ is below a given tolerated threshold then the eq. (24) or (23) can be integrated. For real implementation, it is mostly needful to rewrite these equations to discrete form. Note the Jacobian at the certain instant is only static relation, which is not directly dependent on time.

So we can write:

$$\mathbf{x}(t_k)=\mathbf{x}(t_{k-1})+\dot{\mathbf{x}}(t_k)\cdot \mathrm{Ts}$$

$$\mathbf{x}(t_k)=\mathbf{x}(t_{k-1})+\mathbf{J}_1^{\dagger}(\mathbf{x}_{\mathrm{d}}(t_k))\cdot \dot{\mathbf{q}}_1(t_k)\cdot \mathrm{Ts} \qquad (26)$$

where $\mathbf{J}_1^{\dagger}(\mathbf{x}_{\mathrm{d}})$ is either simple inversion eq.(24) or left pseudo-inverse eq. (25) of Jacobian according to type of the robot.

The graphical representation of eq. (26) in correspondence eq. (25) with discrete time integrator is in Fig.5.

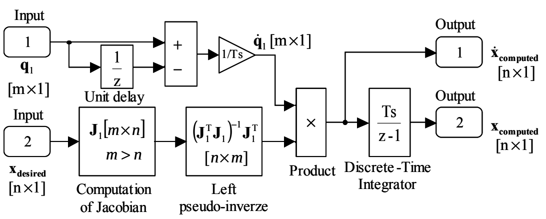

Fig. 5 Feedforward direct kinematics.

4.2.2 Feedback direct kinematics

In previous part it was shown how to provide the direct kinematics by using the differential kinematic equations with knowledge of desired value. However in practice the differences $\mathbf{x}_{\mathrm{d}}$-$\mathbf{x}$ are higher than a given tolerated threshold or we need to move with the robot freely without the knowledge of desired trajectory.

Then in numerical implementation of eq. (26), computation of operational velocities is obtained by using the inverse of the Jacobian evaluated at the previous instant of time

$$\mathbf{x}(t_k)=\mathbf{x}(t_{k-1})+\mathbf{J}_1^{\dagger}(\mathbf{x}(t_{k-1}))\cdot \dot{\mathbf{q}}_1(t_k)\cdot \mathrm{Ts} \qquad (27)$$

Eq. (27) does not satisfy eq. (24), eq. (25) respectively. This inconvenience can be overcome by approach to such scheme, which takes into account the difference between actual topical measured joint-drive coordinates and recomputed joint-drive coordinates from computed (estimated) operational coordinates. Let

$$\mathbf{e}=\mathbf{q_{1M}}-\mathbf{q}_1(\mathbf{x}) \qquad (28)$$

be the expression of such difference.

Consider the time derivative of eq. (27)

$$\dot{\mathbf{e}}=\dot{\mathbf{q}}_{\mathbf{1M}}-\dot{\mathbf{q}}_1(\mathbf{x}) \qquad (29)$$

which, according to the differential kinematics eq. (23), can be rewritten as

$$\dot{\mathbf{e}}=\dot{\mathbf{q}}_{\mathbf{1M}}-\mathbf{J}_1(\mathbf{x})\,\dot{\mathbf{x}} \qquad (30)$$

This equation leads to a feedback scheme of the direct kinematics and relation eq. (30) between operational velocity $\dot{\mathbf{x}}$ and derivative of difference $\dot{\mathbf{e}}$ gives a differential equation, which describes difference evolution over time. Nonetheless, it is necessary to choose a relation between $\dot{\mathbf{x}}$ and $\mathbf{e}$ that ensures convergence of the difference to zero.

Assume the choice

$$\dot{\mathbf{e}}=\dot{\mathbf{q}}_{\mathbf{1M}}-\mathbf{J}_1(\mathbf{x})\,\mathbf{x}\overset{!}{=}-\mathbf{K}(\mathbf{q_{1M}}-\mathbf{q}_1(\mathbf{x}))=-\mathbf{Ke} \qquad (31)$$

which leads to the equivalent linear system

$$\dot{\mathbf{e}}+\mathbf{Ke}=\mathbf{0} \qquad (32)$$

If $\mathbf{K}$ is a positive definite (usually diagonal) matrix, the system eq. (32) is asymptotically stable. The difference tends to zero along measured joint-drive coordinates $\mathbf{q_{1M}}$ with convergence rate that depends on the eigenvalues of matrix $\mathbf{K}$; the larger the eigenvalues, the faster the convergence. Since the scheme is practically implemented as a discrete-time system, it is reasonable to predict that an upper bound exists on the eigenvalues; depending on the sampling period, there will be a limit for the maximum eigenvalue of $\mathbf{K}$ under which asymptotic stability of difference system is guaranteed.

Then we can rewrite eq. (30) to the form

$$\mathbf{J}_1(\mathbf{x})\,\dot{\mathbf{x}}=\dot{\mathbf{q}}_{\mathbf{1M}}+\mathbf{K}(\mathbf{q_{1M}}-\mathbf{q}_1(\mathbf{x})) \qquad (33)$$

and consecutively

$$\dot{\mathbf{x}}=\mathbf{J}_1^{\dagger}(\mathbf{x})\,(\dot{\mathbf{q}}_{\mathbf{1M}}+\mathbf{K}(\mathbf{q_{1M}}-\mathbf{q}_1(\mathbf{x}))) \qquad (34)$$

where $\mathbf{J}_1^{\dagger}(\mathbf{x})$ has a similar meaning, as in previous section, either simple inversion or left pseudo-inverse accordingly to type of the robot.

After this is equal to expression

$$\mathbf{x}(t_k)=\mathbf{x}(t_{k-1})+\mathbf{J}_1^{\dagger}(\mathbf{x}(t_{k\text{-}1}))\cdot$$
$$\cdot(\dot{\mathbf{q}}_{\mathbf{1M}}(t_k)+\mathbf{K}(\mathbf{q_{1M}}(t_k)-\mathbf{q}_1(\mathbf{x}(t_{k-1}))))\cdot \mathrm{Ts} \qquad (35)$$

The block scheme corresponding to the feedback direct kinematics algorithm eq. (35) for redundant case is in Fig. 6.

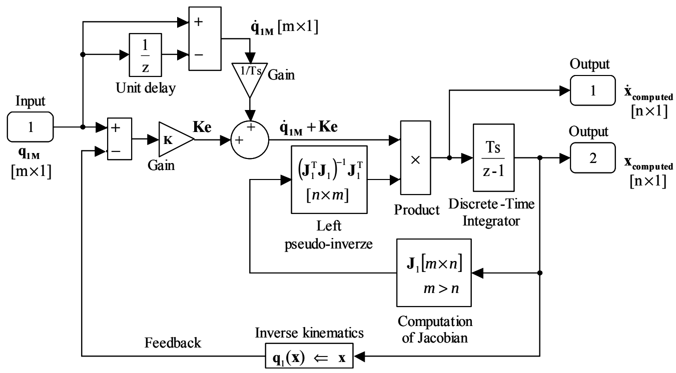

Fig. 6 Feedback direct kinematics.

5 Evaluation of presented approaches

In this section we focus on the last approach to feedback direct kinematics algorithm. The former approaches are suitable for simulation and moreover they need knowledge of the desired trajectory. Mainly in feedforward direct kinematics, this knowledge is cardinal and when there is more difference between desired and measured values this algorithm can't be used. The Newton's method is not bad, but it is slower. It is caused by its iteration character of algorithm.

For introduction, the simple trajectory, composed of spiral and arc segment, was chosen Fig.7.

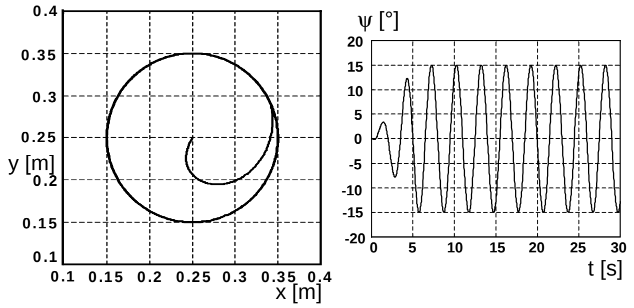

Fig.7. Desired values of operational coordinates Cartesian coordinates: x, y, ψ.

The robot (Fig.1.) begins from center of the workspace and tracking the trajectory with slow increase of sin trend of angle of winding of movable platform.

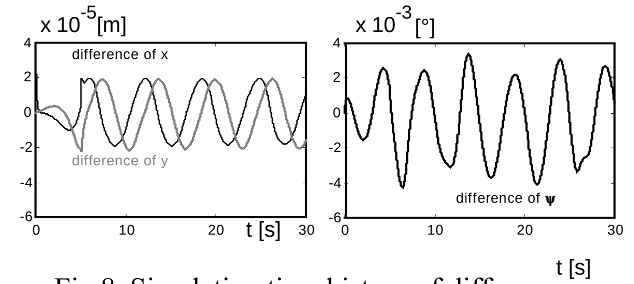

Fig.8. Simulation time history of differences.

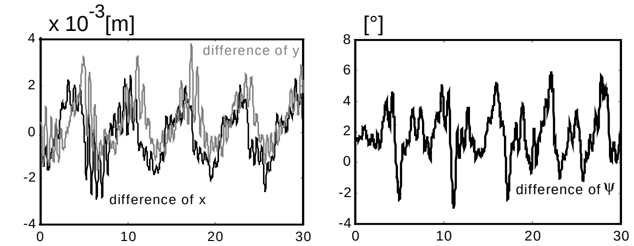

Fig.9. Real time history of differences.

The Fig.8 and Fig.9 show the time history of differences (errors) between desired values $\mathbf{x}_d$ and computed (estimated) values $\mathbf{x}$. For real time process, the sampling Ts equaled 0.002s and for simulation sampling Ts was 0.02s. The positive diagonal matrix was chosen with diagonal element $k_{ij} = 10$.

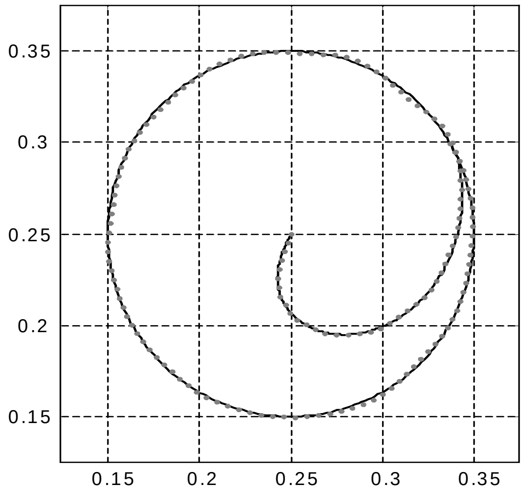

Fig.10 Real-time comparison of desired (solid) and computed (dotted) trajectory.

For real time test the simple proportional controller was used and moreover the robot was not ideally calibrated. That is why the difference between desired $\mathbf{x}_d$ and computed $\mathbf{x}$ operational coordinates is greater than in simulation case, where the direct kinematic algorithms were tested directly without control.

6 Conclusion

Presented approaches to the direct kinematics are suitable for simulation (Newton's method, feedforward algorithm) and mainly for real time using (Newton's method, feedback algorithm). They were successfully tested and shown in this paper.

References:

[1] L. Sciavicco, B. Siciliano, *Modeling and control of robot manipulators*, McGraw-Hill, 1996.

[2] V. Stejskal, M. Valášek, *Kinematics and dynamics of machinery*, Marcel Dekker, Inc.,1996.

[3] L.W. Tsai, Robot analysis the mechanics of serial and parallel manipulators, John Wiley & Sons, Inc., 1999

[4] M. Valášek, P. Steinbauer, Nonlinear control of multibody systems, *Euromech 99*, pp. 437-444.

[5] M. Valášek, Dynamic time parametrization of manipulator trajectories, Kybernetika, vol. 23, 1987.

Acknowledgments

This research is supported by
GA ČR 101/99/0729
"Redundant parallel robots and their control"
and
GA ČR 102/99/1564
"Research and education center of adaptive systems".