Following 3D paths by a manipulator

ALICJA MAZUR, JOANNA PŁASKONKA and MIRELA KACZMAREK

In the paper a description of a manipulator relative to a desired three-dimensional path was presented. The path is parameterized orthogonally to the Serret-Frenet frame which is moving along the curve. For the path two different time parameterizations were chosen. The control law for the RTR manipulator which ensures realization of the task was specified. Theoretical considerations were illustrated by simulation results.

Key words: path following, Serret-Frenet parametrization, orthogonal projection.

1. Introduction

Contemporary robotic manipulators are used in many domains of industry or in everyday life. They can realize different tasks, such as transhipment, transportation in close area, lifting, arranging and treatment of elements using special ending, so-called end-effector etc.

From control point of view, three types of tasks for industrial manipulators can be defined: point stabilization, trajectory tracking and path tracking. In the paper only following some path, i.e. curve parameterized by curvilinear distance, has been considered.

In literature path tracking task has been discussed many times, for instance for mobile platforms [8, 9, 10], for fixed-base manipulators [3] and mobile manipulators [6, 7]. The similar task was defined also for more complex robotic objects, such as autonomic underwater vehicles [2] and flying robots [1]. However, most of papers deal only with two-dimensional paths and present control algorithms which are dedicated to robots moving on flat surfaces, consequently they cannot be simply extended into three-dimensional case. Problems with designing of control law for path tracking are a result of parametrization used in mathematical description of task geometry. Rarely, the task of three-dimensional curve tracking can be realized as a composition of two-dimensional displacements but such approach is not general in any sense and, moreover, it is more time-consuming solution. Therefore concentration of research activity on three-dimensional path tracking seems to be well-founded.

In the paper general solution to the tracking of three-dimensional curves has been proposed. For this reason orthogonal projection on given 3D curve, using Serret-Frenet frame moving along the path, has been designed. General equations of robot's motion describing its position relative the path have been established in Section 2. Formulation of control problem considered in the paper has been presented in Section 3. Main result, i.e. control algorithm solving path tracking problem for fixed-base non-redundant manipulator was designed in Section 4. All considerations were illustrated with simulations for RTR manipulator, presented in Section 5 and summarized in Section 6.

2. Equations of robot's motion relative to the path

2.1. Dynamics of holonomic manipulator

Dynamic modeling of a robot manipulator consists of finding the mapping between the forces exerted on the structure and the joint positions, velocities and accelerations. Dynamic model of holonomic robot with n degrees of freedom can be derived from Lagrange's equations and has a form

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + D(\theta) = u \qquad (1)$$

where:
$\theta, \dot{\theta}, \ddot{\theta} \in R^n$ – vectors of joint positions, velocities and accelerations,
$M(\theta) - (n \times n)$ inertia matrix of manipulator,
$C(\theta, \dot{\theta}) - (n \times n)$ matrix coming from Coriolis and centrifugal forces,
$D(\theta) \in R^n$ – vector of gravity forces,
$u \in R^n$ – vector of controls (input signals from actuators).

Inertia matrix M of any manipulator is always symmetric and positive definite. For robotic manipulators skew-symmetry holds between matrices M and C, i.e.

$$\frac{d}{dt}M = C + C^T. \qquad (2)$$

2.2. Serret-Frenet parametrization for 3D curve

In the paper motion of manipulator moving in three-dimensional space, see Fig. 1, has been considered. Point M describing position of manipulator's end-effector can be defined by cartesian coordinates $p = (x,y,z)^T$ expressed relative to basic frame $X_0Y_0Z_0$. In some distance s calculated along the path, the Serret-Frenet frame should be located

$$Q(s) = [T(s), N(s), B(s)],$$

where T is unit vector tangent to the path, N is unit vector normal and B – unit vector binormal to given curve r
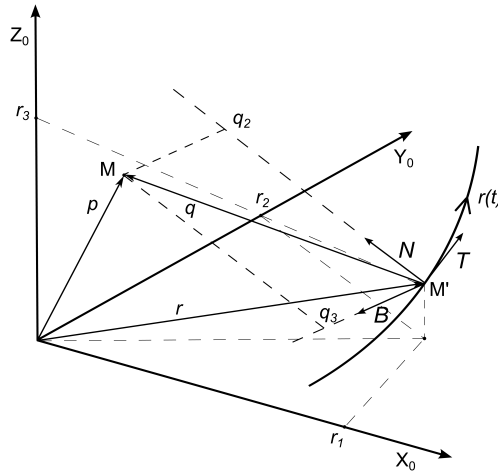
$$r(s) = (r_1(s), r_2(s), r_3(s))^T.$$

Figure 1: Illustration of path tracking problem using three-dimensional Serret-Frenet frame with orthogonal projection on a path.

To illustrate position of proper axes of Serret-Frenet frame, so-called Frenet trihedron has been proposed.
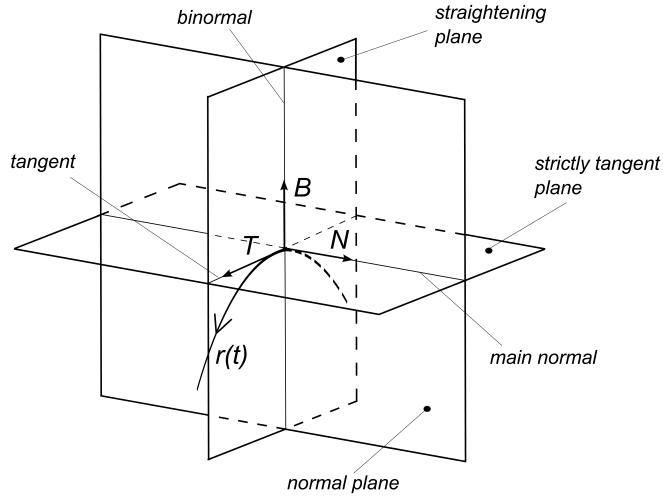

Figure 2: Frenet trihedron.

Unit vectors of Serret-Frenet frame are defined as below

$$T \quad = \quad \frac{dr}{ds} = \frac{\dot{r}}{\| \dot{r} \|}, \tag{3}$$

$$N = \frac{\frac{dT}{ds}}{\| \frac{dT}{ds} \|}, \tag{4}$$

$$B = T \times N. \tag{5}$$

Three-dimensional curve is defined also by two parameters, namely curvature $\kappa(s)$ and torsion $\tau(s)$. Curvature of plain curve in some point is equal to inversion of radius of such circle which is tangent to the curve in the same point. In turn, torsion τ defines how much the curve swerves from the plane. The curvature of the path can be calculated from definition as follows

$$\kappa(s) = \| \frac{dT}{ds} \| = \| \frac{d^2 r(s)}{ds^2} \|,$$

whereas the torsion is defined the following way

$$\tau(s) = \| \frac{dB}{ds} \| = \frac{1}{\kappa^2(s)} \left(\frac{dr}{ds} \times \frac{d^2 r}{ds^2}, \frac{d^3 r}{ds^3} \right),$$

where $(\cdot, \cdot)$ denotes dot product of two vectors.

Motion of Serret-Frenet frame defined along given path can be expressed by Serret-Frenet matrix equations (using curvilinear distance s) as follows

$$\frac{dQ}{ds} = Q(s)W(s), \tag{6}$$

where matrix $W(s)$ defining Serret-Frenet frame (6) has a form

$$W(s) = \begin{bmatrix} 0 & -\kappa(s) & 0 \\ \kappa(s) & 0 & -\tau(s) \\ 0 & \tau(s) & 0 \end{bmatrix}.$$

2.3.   Equation of robot motion relative to path

As it has been presented in Fig. 1, coordinates of point M relative the Serret-Frenet frame are equal to $q = (q_1, q_2, q_3)^T$, while in normal plane (normal plane is spanned by N and B unit vectors) position of the same point is defined by coordinates $(q_2, q_3)^T$. From this reason it is necessary to obtain $(s, q_2, q_3)^T$ coordinates of manipulator if description of robot's motion relative to moving Serret-Frenet frame is needed.

To locate point M in normal plane of the path, some condition has to be fulfilled

$$p - r \perp T \quad \Longrightarrow \quad (T, p - r) = 0. \tag{7}$$

After calculating time derivative of equation (7) and putting $\kappa(s)N$ instead of term $\frac{dT}{ds}$, we get expression for path parametrization as follows

$$\dot{s} = \frac{ds}{dt} = -\frac{(T, \dot{p} - \dot{r})}{\kappa(N, p - r)}. \tag{8}$$

This relationship can be obtained only if position error $(p-r)$ is not equal to zero. Such assumption is valid if moving object is in some distance from the curve. In other words, all further calculations can be done only if robot tracks a path but it is not on the path. In means that equations hold during asymptotic tracking desired path but after reducing position error $(p-r)$ to very small value it is necessary to switch onto control algorithm for following along the path.

Vectors p and q fulfill relationship

$$p = Qq + r, \tag{9}$$

which can be transformed to the form

$$q = Q^T(p-r) = \begin{pmatrix} (T, p-r) \\ (N, p-r) \\ (B, p-r) \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}.$$

From equation (7) it can be concluded, that first coordinate q_1 always equals to zero because $q_1 = (T, p-r) = 0$.

Taking time derivative of (9) and putting (6), we obtain following expression

$$\dot{p} = QWq\dot{s} + Q\dot{q} + \dot{r}. \tag{10}$$

The above equation can be transformed to the form

$$\dot{q} = Q^T(\dot{p}-\dot{r}) - Wq\dot{s} = Q^T(\dot{p}-\dot{r}) - WQ^T(p-r)\dot{s}. \tag{11}$$

Element $WQ^T(p-r)\dot{s}$ can be rewritten by

$$
\begin{aligned}
WQ^T(p-r)\dot{s} &= \dot{s} \begin{bmatrix} 0 & -\kappa(s) & 0 \\ \kappa(s) & 0 & -\tau(s) \\ 0 & \tau(s) & 0 \end{bmatrix} \begin{bmatrix} T^T \\ N^T \\ B^T \end{bmatrix} (p-r) \\
&= \dot{s} \begin{bmatrix} 0 & -\kappa(s) & 0 \\ \kappa(s) & 0 & -\tau(s) \\ 0 & \tau(s) & 0 \end{bmatrix} \begin{pmatrix} (T, p-r) \\ (N, p-r) \\ (B, p-r) \end{pmatrix} \\
&= \dot{s} \begin{pmatrix} -\kappa(N, p-r) \\ \kappa(T, p-r) - \tau(B, p-r) \\ \tau(N, p-r) \end{pmatrix}.
\end{aligned} \tag{12}
$$

In turn, term $Q^T(\dot{p}-\dot{r})$ is equal to

$$Q^T(\dot{p}-\dot{r}) = \begin{bmatrix} T^T \\ N^T \\ B^T \end{bmatrix}(\dot{p}-\dot{r}) = \begin{pmatrix} (T, \dot{p}-\dot{r}) \\ (N, \dot{p}-\dot{r}) \\ (B, \dot{p}-\dot{r}) \end{pmatrix}. \tag{13}$$

Putting (12) and (13), each coordinate in equation (11) can be extracted as follows

$$
\begin{align}
\dot{q}_1 &= (T, \dot{p} - \dot{r}) + \dot{s}\kappa(N, p - r), \tag{14} \\
\dot{q}_2 &= (N, \dot{p} - \dot{r}) - \dot{s}\kappa(T, p - r) + \dot{s}\tau(B, p - r), \tag{15} \\
\dot{q}_3 &= (B, \dot{p} - \dot{r}) - \dot{s}\tau(N, p - r). \tag{16}
\end{align}
$$

Using $(T, p - r) = 0$, equation (15) can be simplified to the form

$$
\dot{q}_2 = (N, \dot{p} - \dot{r}) + \dot{s}\tau(B, p - r). \tag{17}
$$

After elimination $\dot{s}$ from (15) and (16), we get following form of $\dot{q}_2$ and $\dot{q}_3$

$$
\begin{align}
\dot{q}_2 &= (N, \dot{p} - \dot{r}) + \dot{s}\tau(B, p - r) = (N, \dot{p} - \dot{r}) - \frac{\tau}{\kappa} \frac{(B, p - r)(T, \dot{p} - \dot{r})}{(N, p - r)} \notag \\
&= \left(N - \frac{\tau}{\kappa} \frac{(B, p - r)}{(N, p - r)} T, \dot{p} - \dot{r} \right), \tag{18} \\
\dot{q}_3 &= (B, \dot{p} - \dot{r}) - \dot{s}\tau(N, p - r) = (B, \dot{p} - \dot{r}) + \frac{\tau}{\kappa} \frac{(T, \dot{p} - \dot{r})}{(N, p - r)} (N, p - r) \notag \\
&= \left(B + \frac{\tau}{\kappa} T, \dot{p} - \dot{r} \right). \tag{19}
\end{align}
$$

Moreover, after putting (8) into (6) another definition of Serret-Frenet frame can be obtained

$$
\begin{align}
\dot{T} &= \kappa\dot{s}N = -\frac{(T, \dot{p} - \dot{r})}{(N, p - r)} N, \tag{20} \\
\dot{N} &= \dot{s}(-\kappa T + \tau B) = \frac{(T, \dot{p} - \dot{r})}{(N, p - r)} \left(T - \frac{\tau}{\kappa} B \right), \tag{21} \\
\dot{B} &= \dot{s}\tau N = \frac{\tau}{\kappa} \frac{(T, \dot{p} - \dot{r})}{(N, p - r)} N. \tag{22}
\end{align}
$$

Finally, we get the following equations describing robot position relative to moving Serret-Frenet frame

$$
\begin{align}
\dot{s} &= -\frac{(T, \dot{p} - \dot{r})}{\kappa(N, p - r)}, \tag{23} \\
\dot{q}_2 &= \left(N - \frac{\tau}{\kappa} \frac{(B, p - r)}{(N, p - r)} T, \dot{p} - \dot{r} \right), \tag{24} \\
\dot{q}_3 &= \left(B + \frac{\tau}{\kappa} T, \dot{p} - \dot{r} \right), \tag{25} \\
\dot{T} &= -\frac{(T, \dot{p} - \dot{r})}{(N, p - r)} N, \tag{26} \\
\dot{N} &= \frac{(T, \dot{p} - \dot{r})}{(N, p - r)} \left(T - \frac{\tau}{\kappa} B \right), \tag{27}
\end{align}
$$

$$\dot{B} = \frac{\tau}{\kappa}\frac{(T,\dot{p}-\dot{r})}{(N,p-r)}N. \tag{28}$$

Equations (23)-(28) are point of departure to design control algorithms for three-dimensional path tracking. It is necessary to remember that vector $p = (x,y,z)^T$ describes robot's cartesian position relative to fixed basic frame $X_0Y_0Z_0$, vector $r = (r_1,r_2,r_3)^T$ describes given path in R^3 relative to the same basic frame and $(s,q_2,q_3)^T$ are coordinates of the robot relative to the path.

3. Control problem statement

In the paper, we want to address the following control problem to fixed-base robotic arms:

Determine control law u such that a holonomic robotic manipulator with fully known dynamics follows the desired smooth path defined in R^3 space.

To design path tracking controller for considered manipulators, let us observe that complete mathematical equations describing manipulator relative to desired curve in R^3 space have cascaded structure consisting of two groups of equations: kinematics (23)-(28) and dynamics (1).
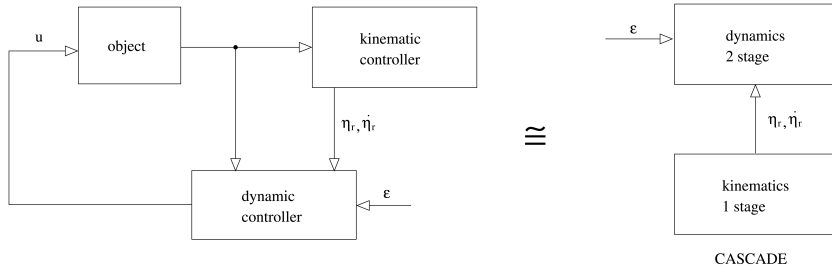

Figure 3: Structure of the proposed control algorithm: cascade with two stages.

For this reason the structure of the controller is divided into two parts, see Fig. 3, working simultaneously:

- kinematic controller θ_r – represents a vector of embedded control inputs, which ensure realization of the task for the geometric path tracking problem if the dynamics were not present. Such the controller generates 'velocity profile' which can be executed in practice to follow the desired curve in R^3.

- dynamic controller – as a consequence of cascaded structure of the system model, the system's velocities cannot be commanded directly, as it is assumed in the design of kinematic controller, and instead they must be realized as the output of the dynamics driven by u.

To solve the presented control problem for manipulators, backstepping-like algorithm [5] has to be evoked. Backstepping is well-known and often used approach to control cascaded systems, e.g. systems with nonholonomic constraints or systems with additional constraints on robot velocities, such as solution to the geometric path tracking problem.

4.   Control algorithm for non-redundant manipulator

4.1.   Description of manipulator moving along the curve

Equations (23)-(28) can be rewritten as follows

$$\dot{s} = -\frac{(T, \dot{p} - \dot{r})}{\kappa(N, p - r)} = P_1 \dot{p} + R_1 \tag{29}$$

$$\dot{q}_2 = \left(N - \frac{\tau}{\kappa}\frac{(B, p - r)}{(N, p - r)}T, \dot{p} - \dot{r}\right) = P_2 \dot{p} + R_2, \tag{30}$$

$$\dot{q}_3 = \left(B + \frac{\tau}{\kappa}T, \dot{p} - \dot{r}\right) = P_3 \dot{p} + R_3, \tag{31}$$

with selected elements equal to

$$P_1 = -\frac{T^T}{\kappa(N, p - r)},$$

$$P_2 = \left(N - \frac{\tau}{\kappa}\frac{(B, p - r)}{(N, p - r)}T\right)^T,$$

$$P_3 = \left(B + \frac{\tau}{\kappa}T\right)^T,$$

$$R_1 = \frac{(T, \dot{r})}{\kappa(N, p - r)},$$

$$R_2 = -\left(N - \frac{\tau}{\kappa}\frac{(B, p - r)}{(N, p - r)}T, \dot{r}\right),$$

$$R_3 = -\left(B + \frac{\tau}{\kappa}T, \dot{r}\right).$$

After rearranging, equations (29)-(31) can be expressed in matrix form

$$\dot{\xi} = P\dot{p} + R, \tag{32}$$

where

$$\xi = \begin{pmatrix} s \\ q_2 \\ q_3 \end{pmatrix}, \quad P = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}, \quad R = \begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix}.$$

On the other hand, cartesian coordinates of end-effector p are functions of joint variables, given by manipulator's kinematics

$$p = k(\theta), \tag{33}$$

so $\dot{p}$ depends on joint velocities in the following manner

$$\dot{p} = \frac{\partial k}{\partial \theta} \dot{\theta} = J(\theta)\dot{\theta}, \tag{34}$$

where $J(\theta)$ is Jacobi matrix for position coordinates. Substituting (34) into (32), we get expression

$$\dot{\xi} = PJ\dot{\theta} + R, \tag{35}$$

in which signal $\dot{\theta}$ plays a role of control input.

4.2. Kinematic controller

If manipulator is non-redundant, as it has been assumed earlier, then Jacobi matrix is square matrix. If desired path does not require singular configurations by manipulator, then matrix $J(\theta)$ is invertible. Next, if it is possible to invert matrix P, the following kinematic control algorithm can be proposed

$$\dot{\theta}_{ref} = J^{-1}P^{-1}(\dot{\xi}_d - K_p e_\xi - R), \quad e_\xi = \xi - \xi_d, \tag{36}$$

with positive definite regulation matrix $K_p > 0$. Vector

$$\xi_d = (s_d(t), q_{2d}, q_{3d})^T$$

describes desired behavior of path tracking errors, usually $q_{2d} = 0$ and $q_{3d} = 0$, whereas desired path parametrization (dependency on time) can be arbitrary function, depending on designer's choice. Designed velocity (36) substituted into equations (35) makes path tracking error e_ξ convergent to zero. Signal $\dot{\theta}_{ref}$ is proposed velocity of the robot's joints, i.e. 'velocity profile' coming from kinematic controller – motion planning subsystem. Such velocity has to be next realized on dynamic level.

In the kinematic control law (36) inversion of Jacobi matrix plays a special role. Its existence is crucial from control point of view and therefore assumption about passing by manipulator only through non-singular configurations has to be added. Such assumption occurs not only in control law presented in this article but it has been made in many robot's tasks, for example for motion defined in cartesian coordinates even than manipulator has only rotational degrees of freedom. In the case of motion along straight line an input-output decoupling control algorithm [4] has to be used which near singular configurations stops motion of industrial manipulators. The same approach, i.e. stopping near singularities, can be used in kinematic control law given by (36).

126   A. MAZUR, J. PŁASKONKA, M. KACZMAREK

4.3. Dynamic controller

For realization of trajectory $\xi_d(t)$ tracking, dynamic control algorithm can be proposed

$$u = M(\theta)\ddot{\theta}_{ref} + C(\theta,\dot{\theta})\dot{\theta}_{ref} + D(\theta) - K_d\dot{e}_\theta, \quad \dot{e}_\theta = \dot{\theta} - \dot{\theta}_{ref}, \quad K_d > 0, \qquad (37)$$

with positive definite regulation matrix K_d. Such control law applied to robot dynamics (1) preserves asymptotic convergence of velocity tracking error $\dot{e}_\theta$ to zero. It implies proper realization of path tracking process, because real velocities of robot joints θ reproduce velocities θ_{ref} planned by kinematic controller, i.e. motion planning control subsystem.

Proof of convergence of dynamic controller. Equations of system (1) with applied control law (37) have a form

$$M(\theta)\ddot{e}_\theta + C(\theta,\dot{\theta})\dot{e}_\theta + K_d\dot{e}_\theta = 0. \qquad (38)$$

For closed-loop system (38) following Lyapunov-like function should be considered

$$V(\dot{e}_\theta) = \frac{1}{2}\dot{e}_\theta^T M(\theta)\dot{e}_\theta. \qquad (39)$$

Due to positive definiteness of inertia matrix $M(\theta)$, such function $V(\dot{e}_\theta)$ is always non-negative.

Using skew-symmetry property (2), time derivative of V calculated along trajectories of system (38) can be expressed as

$$\begin{aligned} \dot{V} &= \dot{e}_\theta^T M(\theta)\ddot{e}_\theta + \frac{1}{2}\dot{e}_\theta^T \dot{M}(\theta)\dot{e}_\theta = \dot{e}_\theta^T(-C(\theta,\dot{\theta})\dot{e}_\theta - K_d\dot{e}_\theta) + \frac{1}{2}\dot{e}_\theta^T \dot{M}(\theta)\dot{e}_\theta \\ &= -\dot{e}_\theta^T K_d\dot{e}_\theta \leqslant 0. \end{aligned} \qquad (40)$$

From La Salle invariance principle it can be straightforwardly concluded that $\dot{e}_\theta = 0$ is asymptotic stable equilibrium point for system (38). This completes the proof.

5. Simulation study

The simulations were run with the MATLAB package and the SIMULINK toolbox. As an object of simulations we have taken RTR manipulator with three degrees of freedom, presented in Fig. 4.

Links of the RTR manipulator have been modeled as homogenous sticks with length equal to $l_2 = 0.9$ m and $l_3 = 1$ m and masses $m_2 = 20$ kg and $m_3 = 20$ kg. Dynamics of RTR manipulator are given by (1) with elements equal to:
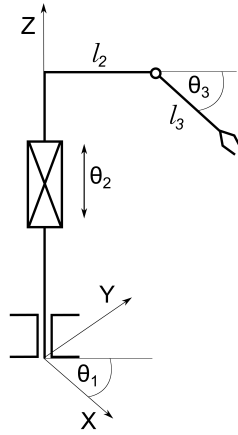
Figure 4: Manipulator RTR – object of simulation

- inertia matrix

$$M(\theta) = \begin{bmatrix} M_{11} & 0 & 0 \\ 0 & M_{22} & M_{23} \\ 0 & M_{23} & M_{33} \end{bmatrix},$$

$$
\begin{aligned}
M_{11} &= (1/3)m_2 l_2^2 + m_3 \left(l_2^2 + (1/3)l_3^2 \cos^2\theta_3 + l_2 l_3 \cos\theta_3 \right), \\
M_{22} &= m_2 + m_3, \\
M_{23} &= (1/2)m_3 l_2 l_3 \cos\theta_3, \\
M_{33} &= (1/3)m_3 l_3^2,
\end{aligned}
$$

- matrix of Coriolis and centrifugal forces

$$C(\theta,\dot{\theta}) = \begin{bmatrix} C_{11} & 0 & C_{13} \\ 0 & 0 & C_{23} \\ C_{31} & 0 & 0 \end{bmatrix},$$

$$
\begin{aligned}
C_{11} &= \dot{\theta}_3 \left(-(1/2)m_3 l_2 l_3 \sin\theta_3 - (1/3)m_3 l_3^2 \sin\theta_3 \cos\theta_3 \right), \\
C_{13} &= -\dot{\theta}_1 \left((1/2)m_3 l_2 l_3 \sin\theta_3 + (1/3)m_3 l_3^2 \sin\theta_3 \cos\theta_3 \right), \\
C_{23} &= -(1/2)\dot{\theta}_3 m_3 l_2 l_3 \sin\theta_3, \\
C_{31} &= \dot{\theta}_1 \left((1/2)m_3 l_2 l_3 \sin\theta_3 + (1/3)m_3 l_3^2 \sin\theta_3 \cos\theta_3 \right),
\end{aligned}
\qquad (41)
$$

- gravity vector

$$D(\theta) = \begin{pmatrix} 0 \\ (m_2 + m_3)g \\ \frac{1}{2}gm_3 l_3 \cos\theta_3 \end{pmatrix}.$$

The goal of the simulations was to investigate a behavior of this rigid fixed-base manipulator with the controllers (36) and (37) proposed in the paper. Simulation was conducted for two different function of path parameterizations s_d: linear and quadratic.

As a desired path a screw curve has been chosen

$$r(s) = (r_1(s), r_2(s), r_3(s))^T = \left(\cos\frac{s}{\sqrt{2}}, \sin\frac{s}{\sqrt{2}}, \frac{s}{\sqrt{2}} \right)^T. \tag{42}$$

Unit vectors T, N and B and path parameters are selected as below

$$T(s) = \frac{1}{\sqrt{2}} \begin{pmatrix} -\sin\frac{s}{\sqrt{2}} \\ \cos\frac{s}{\sqrt{2}} \\ 1 \end{pmatrix} \qquad N(s) = \begin{pmatrix} -\cos\frac{s}{\sqrt{2}} \\ -\sin\frac{s}{\sqrt{2}} \\ 0 \end{pmatrix}$$

$$B(s) = \frac{1}{\sqrt{2}} \begin{pmatrix} \sin\frac{s}{\sqrt{2}} \\ -\cos\frac{s}{\sqrt{2}} \\ 1 \end{pmatrix} \qquad \kappa(s) = \frac{1}{2}, \qquad \tau(s) = \frac{1}{2}.$$

Cartesian position of end-effector for RTR manipulator can be expressed as

$$p = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos\theta_1(l_3\cos\theta_3 + l_2) \\ \sin\theta_1(l_3\cos\theta_3 + l_2 \\ l_3\sin\theta_3 + \theta_2 \end{pmatrix},$$

then Jacobi matrix has a form

$$J(\theta) = \begin{bmatrix} -\sin\theta_1(l_3\cos\theta_3 + l_2) & 0 & -\cos\theta_1\sin\theta_3 l_3 \\ \cos\theta_1(l_3\cos\theta_3 + l_2) & 0 & -\sin\theta_1\sin\theta_3 l_3 \\ 0 & 1 & \cos\theta_3 l_3 \end{bmatrix}.$$

Inverse Jacobi matrix $J^{-1}(\theta)$ exists if the manipulator RTR does not pass through singular configurations

$$\sin\theta_3 = 0 \qquad l_3\cos\theta_3 + l_2 = 0.$$

First singular configuration means that robotic arm is stretchet in maximal range and the second one can occur only if length l_2 is bigger than l_3. It is possible to avoid all singularities in robotic joint space if manipulator can realize motion from initial configuration

to desired task without necessity to pass through singular configuration. Such a case is
presented in this simulation study.

Matrix P is non-singular in whole domain of parametrization validity because

$$\det\ P = \frac{-1}{\kappa(N, p-r)} \neq 0,$$

only if vector $p-r$ (distance between effector and given path) is different from zero.

5.1. Linear path parametrization s_d

Definition of the path with linear time dependency can be selected e.g. as

$$\xi_d(t) = (s_d, q_{2d}, q_{3d})^T(t) = \left(\frac{t}{10}, 0, 0\right)^T. \tag{43}$$

In simulations the influence of the regulation matrix K_p on behavior of manipulator's
end-effector has been tested. Matrix K_p is diagonal with the same value of parameter on
the diagonal. Matrix K_d is chosen as constant and equal to $K_d = \text{diag}\{50\}$.

Tracking of the desired path for RTR manipulator by linear time parametrization has
been presented in Figure 5. Tracking errors of cartesian coordinates in normal plane have
been presented in Figures 6-7. In turn, errors of curvilinear distance $e_s = s - s_d$ have been
plotted in Figure 8.

From plots in Figures 5-8 it can be concluded that path tracking with linear time
parametrization is realized properly and tracking errors go to zero. Moreover, real curvi-
linear parametrization $s(t)$ tends to the desired function $s_d(t)$.

It is worth to observe that distance tracking errors e_2 and e_3 have only positive values
– it means that distance $(p-r)$ is positive and does not change sign during regulation
process. In other words, matrix P is non-singular and path parametrization using orthog-
onal projection on the curve is valid.

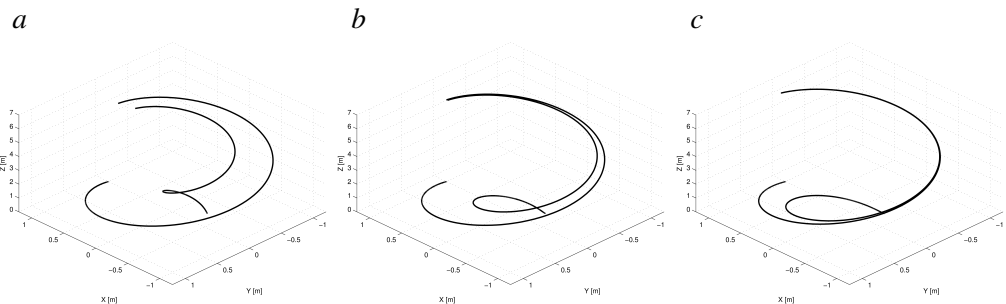*a*   *b*   *c*

Figure 5: Trajectory of manipulator for linear time parametrization:
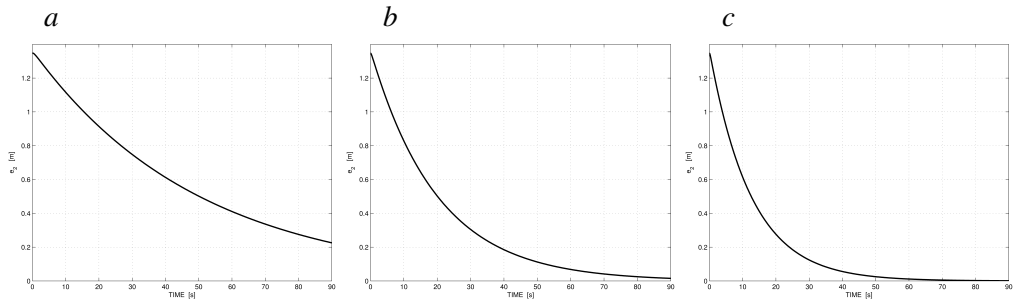a – trajectory for $K_p = 0.02$, b – trajectory for $K_p = 0.05$, c – trajectory for $K_p = 0.1$

Figure 6: Distance error $e_2 = q_2 - q_{2d}$ for linear time parametrization:
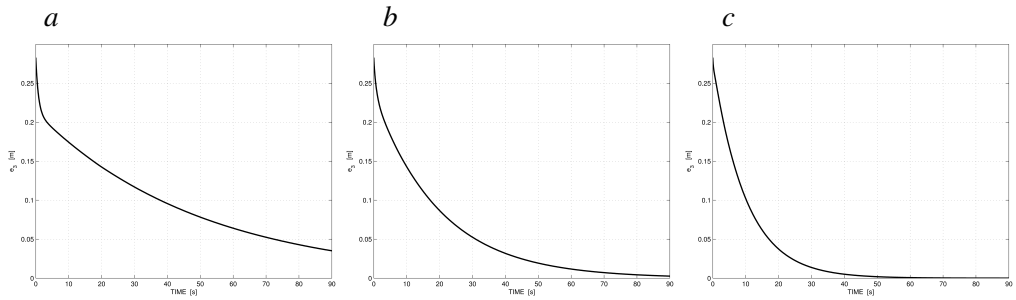a – error e_2 for $K_p = 0.02$, b – error e_2 for $K_p = 0.05$, c – error e_2 for $K_p = 0.1$

Figure 7: Distance error $e_3 = q_3 - q_{3d}$ for linear time parametrization:
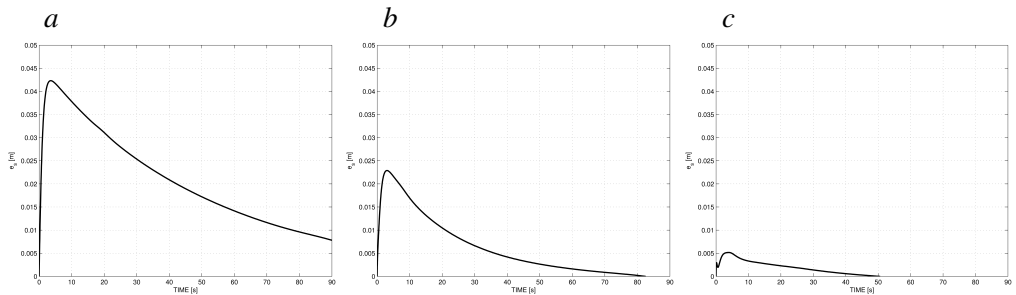a – error e_3 for $K_p = 0.02$, b – error e_3 for $K_p = 0.05$, c – error e_3 for $K_p = 0.1$

Figure 8: Curvilinear error $e_s = s - s_d$ for linear time parametrization:
a – error e_s for $K_p = 0.02$, b – error e_s for $K_p = 0.05$, c – error e_s for $K_p = 0.1$

5.2. Quadratic path parametrization s_d

Definition of the path with quadratic time dependency can be selected e.g. as

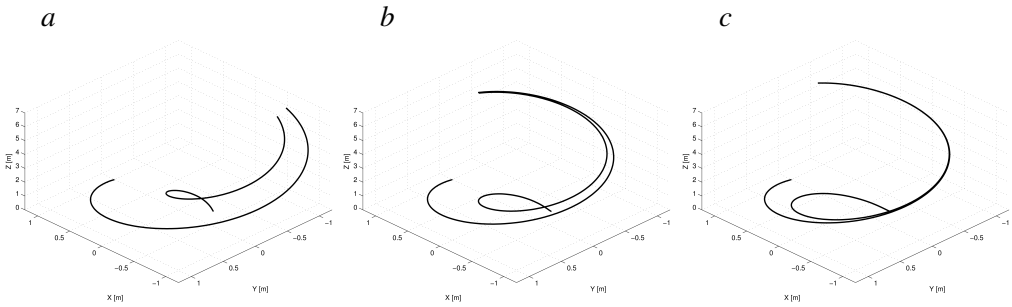$$\xi_d(t) = (s_d, q_{2d}, q_{3d})^T(t) = \left(0.1t - 0.0001t^2, 0, 0\right)^T. \tag{44}$$

Figure 9: Trajectory of manipulator for quadratic time parametrization:
a – trajectory for $K_p = 0.02$, b – trajectory for $K_p = 0.05$, c – trajectory for $K_p = 0.1$

Tracking of the desired path for RTR manipulator by quadratic time parametrization has been presented in Fig. 9. Tracking errors of cartesian coordinates in normal plane have been presented in Figures 10-11.
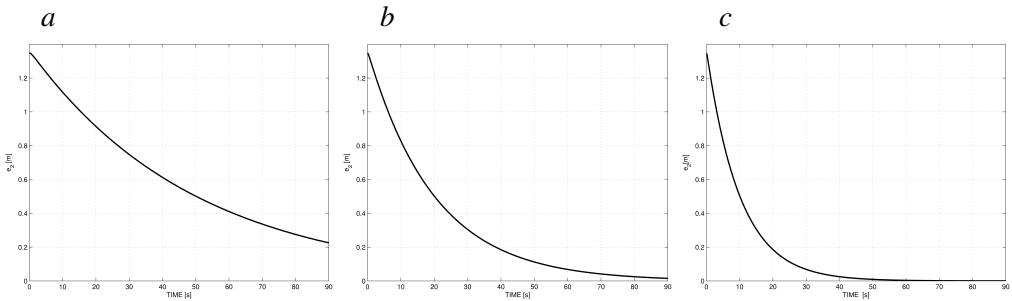

Figure 10: Distance error $e_2 = q_2 - q_{2d}$ for quadratic time parametrization:
a – error e_2 for $K_p = 0.02$, b – error e_2 for $K_p = 0.05$, c – error e_2 for $K_p = 0.1$

In turn, errors of curvilinear distance $e_s = s - s_d$ have been plotted in Figure 12.

From plots 9-12 it can be concluded that path tracking with quadratic time parametrization is realized properly and tracking errors go to zero. Moreover, real curvilinear parametrization $s(t)$ tends to the desired function $s_d(t)$.

Similarly to linear parametrization, distance tracking errors e_2 and e_3 have only positive values – it means that distance $p - r$ is positive and does not change sign during regulation process. In other words, matrix P is non-singular and path parametrization is valid in simulation.
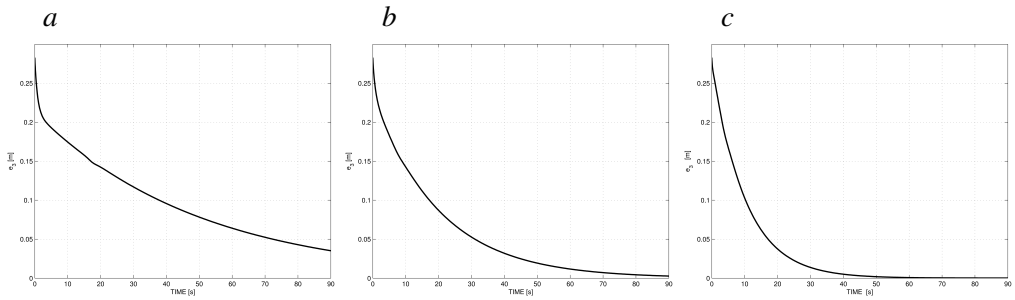
a   b   c

Figure 11: Distance error $e_3 = q_3 - q_{3d}$ for quadratic time parametrization:
a – error e_3 for $K_p = 0.02$, b – error e_3 for $K_p = 0.05$, c – error e_3 for $K_p = 0.1$

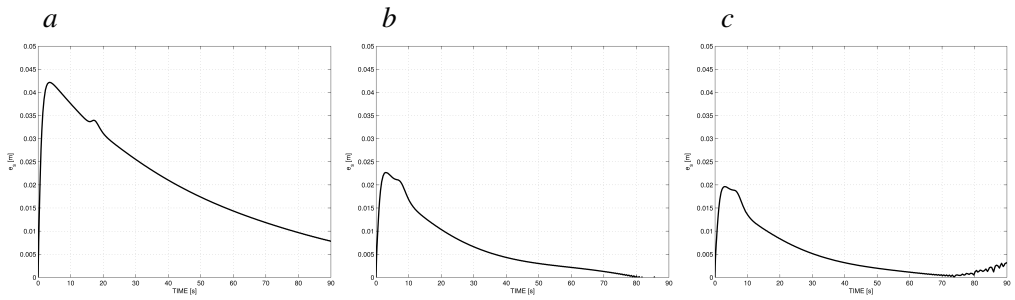a   b   c

Figure 12: Curvilinear error $e_s = s - s_d$ for quadratic time parametrization:
a – error e_s for $K_p = 0.02$, b – error e_s for $K_p = 0.05$, c – error e_s for $K_p = 0.1$

6. Conclusions

In the paper general solution to path tracking problem in three-dimensional space has been presented. To achieve robot's description relative to the curve, Serret-Frenet parametrization with orthogonal projection on given path has been used. Obtained equations are valid only in such a case, if distance between object and the path, i.e. $p - r$, does not equal to zero.

For stationary non-redundant manipulator cascaded control scheme has been proposed. Such control scheme consists of two stages in cascade, namely kinematic controller solving geometric problem of path tracking, and dynamic controller which makes possibly to realize velocities designed in kinematic controller on dynamic level.

To test, if designed control approach solves path tracking problem for any time dependency of curvilinear distance, two different time parameterizations for the same curve has been chosen: linear and quadratic function on time. Simulations have confirmed proper action of control algorithm introduced in the paper.

As trends for future works it is worth to mention that new defined control algorithm can be used to regulate other robotic objects, such as mobile manipulators or object with

parametric uncertainty of dynamics. However, most important limitation of proposed control scheme is fact that distance to the path has to stay not equal to zero. It implies that motion along the three-dimensional path has to be realized without overshoot.

References

[1] V. CICHELLA, I. KAMINER, E. XARGAY, V. DOBROKHODOV, N. HOV-AKIMYAN, A. AGUIAR, and A. PASCOAL: A Lyapunov-based approach for time-coordinated 3D path-following of multiple quadrotors. In *Proc. of the 51st Annual Conf. on Decision and Control*, Maui, Hawaii, USA, (2012), 1776-1781.

[2] P. ENCARNAÇAO and A. PASCOAL: 3D path following for autonomous underwater vehicle. In *Proc. of the 39th IEEE Conf. on Decision and Control*, **3** (2000), 2977-2982.

[3] M. GALICKI: Adaptive control of kinematically redundant manipulator. *Lecture Notes in Control and Information Sciences*, **335** (2006), 129-139.

[4] A. ISIDORI: Nonlinear Control Systems. Springer, London, 3rd edition, 1995.

[5] M. KRSTIĆ, I. KANELLAKOPOULOS and P. KOKOTOVIĆ: Nonlinear and Adaptive Control Design. J. Wiley and Sons, New York, 1995.

[6] A. MAZUR: Hybrid adaptive control laws solving a path following problem for nonholonomic mobile manipulators. *Int. J. of Control*, **77**(15), (2004), 1297-1306.

[7] A. MAZUR and D. SZAKIEL: On path following control of nonholonomic mobile manipulators. *Int. J. Applied Mathematics and Computer Sciences*, **19**(4), (2009), 561-574.

[8] A. MICAELLI and C. SAMSON: Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Technical Report No. 2097, INRIA, Sophia-Antipolis, 1993.

[9] J. PŁASKONKA: The path following control of a unicycle based on the chained form of a kinematic model derived with respect to the Serret-Frenet frame. In *Proc. of the 17th Int. Conf. on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland, (2012), 617–620a.

[10] D. SOETANTO, L. LAPIERRE and A. PASCOAL: Adaptive, non-singular path-following control of dynamic wheeled robots. In *Proc. of the 42nd IEEE Conf. on Decision and Control*, Maui, Hawaii, USA, **2** (2003), 1765-1770.