10.24425/acs.2023.146279

Archives of Control Sciences
Volume 33(LXIX), 2023
No. 2, *pages 339–370*

The non-orthogonal Serret–Frenet parametrization applied to the path following problem of a manipulator with partially known dynamics

Alicja MAZUR and Filip DYBA

In this paper an application of the Serret–Frenet parametrization of a curve to the path following task is presented. This curvilinear parametrization method is used to obtain a control object description relative to the desired curve defined in the three-dimensional space. In order to derive proper equations, the innovative approach of the non-orthogonal projection of a control object on the given path is investigated. The non-orthogonal projection allows to design a global control algorithm. The proposed solution results in a cascade structure of the control system. Thus, the backstepping integrator algorithm was applied to create a control law. Due to the partial knowledge of control object dynamic parameters, an adaptive algorithm is taken into account. Theoretical considerations are confirmed with simulation study. Conducted simulations illustrated following paths at different levels of complexity by a holonomic non-redundant manipulator with a fixed base.

Key words: backstepping integrator algorithm, holonomic manipulator, non-orthogonal projection, path following, Serret–Frenet parametrization

1. Introduction

Over the past decades, the control problem of robotic systems, especially with constraints in motion, has attracted a great deal of attention. In this work, the path following problem of a holonomic manipulator has been addressed. Realization of a robot developed motion usually can be performed by the trajectory tracking task but in such a case sometimes the robot has to achieve very high velocities: trajectories as "curves parametrized by time" have to be executed in a fast time

The authors are with Department of Cybernetics and Robotics, Faculty of Electronics, Photonics and Microsystems, Wrocław University of Science and Technology, Janiszewskiego Street 11/17, Wrocław, 50-372, Poland.

The corresponding author is F. Dyba, e-mail: filip.dyba@pwr.edu.pl

Received 08.12.2022. Revised 06.04.2023.

regime. However, in many practical situations velocities of the system are bounded by limited actuations and at the time it is better to plan the robot motion as a motion along a curvilinear path. For the latter method, the approach utilized the Serret–Frenet frame is frequently adopted to obtain in very intuitive way the error dynamics equations.

In the literature path following task has been discussed many times, for instance for mobile robots [13, 18, 23], for holonomic fixed-base manipulators [9, 10, 16], and mobile manipulators [15, 17]. The similar control problem was defined also for more complex robotic objects, such as autonomic underwater vehicles [6, 24], flying robots [14], and a group of robots [4]. However, most of the mentioned papers deal only with two-dimensional paths. Thus, the presented control algorithms are restricted to robots moving on flat surfaces. As a consequence, they cannot be easily extended into the three-dimensional case. Hence, in the paper the manipulator control in the $\mathbb{R}^3$ space is considered.

The most important part of path tracking controller designing is the usage of curvilinear parametrization, usually the Serret–Frenet approach [8, 21], or, less frequently, the Bishop approach [1]. The concept of the Serret–Frenet frame associated with a reference object moving in the three-dimensional space along the desired curve makes it possible to describe an object not relative to the inertial frame, but to the moving one. Equations expressing position and orientation errors (defined between the object and the reference frame located on the curve) play a similar role to non-integrable constraints of the first order in motion of non-holonomic robots.

There are two methods of application the Serret–Frenet description to obtain motion equation preserving path following in 3D space. The first of them is the orthogonal parametrization with the orthogonal projection of the object on the path. In such a case the location of an object relative to the curve is expressed as the shortest distance between the object and its projection. The object is located on the normal plane spanned by normal and binormal vectors [16]. This approach have been eagerly used in many control algorithms, e.g. [3, 5]. Unlike the orthogonal approach, in the non-orthogonal parametrization the reference robot can be ahead of the robot being controlled or behind in the comparison to it, since the starting point of the Serret–Frenet frame is assumed to be at the beginning of the path. As a result, the reference robot moves along the path freely, not necessarily at the shortest distance from the controlled robot.

In the orthogonal approach equations of error dynamics can be singular, therefore the parametrization is defined only locally, near the path. On the other hand, the non-orthogonal parametrization cannot be singular, but the price to pay for it is the increased number of state variables that are controlled with an unchanged number of control inputs. It is worth to pay attention to the fact that the non-orthogonal parametrization does not define the formula on speed of movement along the path $\dot{s}$. This means that the values of $\dot{s}$ can be treated

on the design stage of the control algorithm as an additional control variable. To conclude, although the usage of the non-orthogonal projection results in the increase of the dimensionality of the control problem, the designed algorithm is global, which is a significant advantage of the approach described in this paper compared to the solutions presented so far.

Hence, in this work the non-orthogonal 3D parametrization has been used to get equations of motion expressed relative to the desired smooth path in the 3D space. The whole mathematical model of the path following problem for a fixed-based non-redundant manipulator consists of two groups of equations, namely of the Serret–Frenet parametrization for the manipulator kinematics and the partially known dynamics, which are connected to each other and form a two-stage cascade system. Therefore, the backstepping integrator algorithm is used as the control method for the cascade of equations [12].

The paper structure has been defined as follows. In Section 2 a mathematical model of a holonomic manipulator with partially known dynamics and equations resulting from application of the non-orthogonal Serret–Frenet parametrization have been presented. The control problem has been formulated in Section 3. The main result of the paper, i.e. the path following algorithm based on the non-orthogonal Serret–Frenet parametrization, has been shown in Section 4. The convergence proof of the proposed control algorithm has been also given there. Section 5 consists of descriptions of considered simulation cases and derived results. All work has been summarized in Section 6.

2. Mathematical models

2.1. Holonomic manipulator

The considered control object is a holonomic manipulator, which can be described with the following equations of dynamics [25]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}(\boldsymbol{q}) = \boldsymbol{u}, \tag{1}$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ are the vectors of joint positions, velocities and accelerations, respectively, $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is the matrix of Coriolis and centrifugal forces, $\boldsymbol{D}(\boldsymbol{q})$ is the gravity forces vector, and $\boldsymbol{u}$ is the vector of applied controls.

It can be concluded directly from the definition that the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ is positive-definite and symmetric [25]. Moreover, the following relation holds

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{C}^T(\boldsymbol{q},\dot{\boldsymbol{q}}). \tag{2}$$

The location of the manipulator end-effector in the manipulator base frame is described by the forward kinematics

$$\boldsymbol{p} = k(\boldsymbol{q}). \tag{3}$$

Velocities of the end-effector frame with respect to the manipulator base reference frame may be calculated by differentiating equation (3) with time

$$\dot{\boldsymbol{p}} = \frac{\partial k(\boldsymbol{q})}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{4}$$

where $\boldsymbol{J}(\boldsymbol{q})$ is the Jacobi matrix. For the considered fixed-base manipulator it is assumed that the manipulator base frame is equal to the inertial reference frame.

2.1.1. Partially known dynamics

It is worth noticing that in many applications parameters of the dynamic model (1) are unknown [22]. Despite that fact, it is possible to control the object correctly. However, it has to be assumed that the unknown parameters are constant and the model is linearly dependent on them [22]. If there are some unidentified values, the model (1) may be expressed with the usage of regression matrix $\boldsymbol{Y}$

$$\begin{aligned}\boldsymbol{M}(\boldsymbol{q},\boldsymbol{a})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{a})\dot{\boldsymbol{q}} + \boldsymbol{D}(\boldsymbol{q},\boldsymbol{a})&\\ = \boldsymbol{M}_0(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_0(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_0(\boldsymbol{q}) + \boldsymbol{M}_{\boldsymbol{a}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_{\boldsymbol{a}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_{\boldsymbol{a}}(\boldsymbol{q})&\\ = \boldsymbol{M}_0(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_0(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_0(\boldsymbol{q}) + \boldsymbol{Y}(\ddot{\boldsymbol{q}},\dot{\boldsymbol{q}},\boldsymbol{q},\boldsymbol{q})\boldsymbol{a} = \boldsymbol{u},&\end{aligned} \tag{5}$$

where $\boldsymbol{a}$ is the vector of unknown parameters, matrices with 0 in the subscript denote the known part of the model, and matrices with $\boldsymbol{a}$ in the subscript correspond to the part of the model dependent on the unknown parameters. The model (5) is partially parametrized, but the full parametrization may be considered. In such a case matrices $\boldsymbol{M}_0$, $\boldsymbol{C}_0$, $\boldsymbol{D}_0$ are equal to zero matrices of proper size.

2.2. Serret–Frenet parametrization

One of the most popular methods of curvilinear parametrizations is the Serret–Frenet parametrization described independently by [8] and [21]. The curve geometry in the three-dimensional space is defined by three vectors: tangential to the curve $\boldsymbol{T}$, normal to the curve $\boldsymbol{N}$ and binormal to the curve $\boldsymbol{B}$, which create an orthonormal basis in $\mathbb{R}^3$ and span the Frenet trihedron [25]. Their position with respect to a certain curve $\boldsymbol{r}(u)$ is presented in Fig. 1. Versors of the Serret–Frenet frame may be defined with respect to an arbitrary parameter u with the following equations [7]

$$\boldsymbol{T}(u) = \frac{\dfrac{\mathrm{d}\boldsymbol{r}(u)}{\mathrm{d}u}}{\left\|\dfrac{\mathrm{d}\boldsymbol{r}(u)}{\mathrm{d}u}\right\|}, \tag{6a}$$

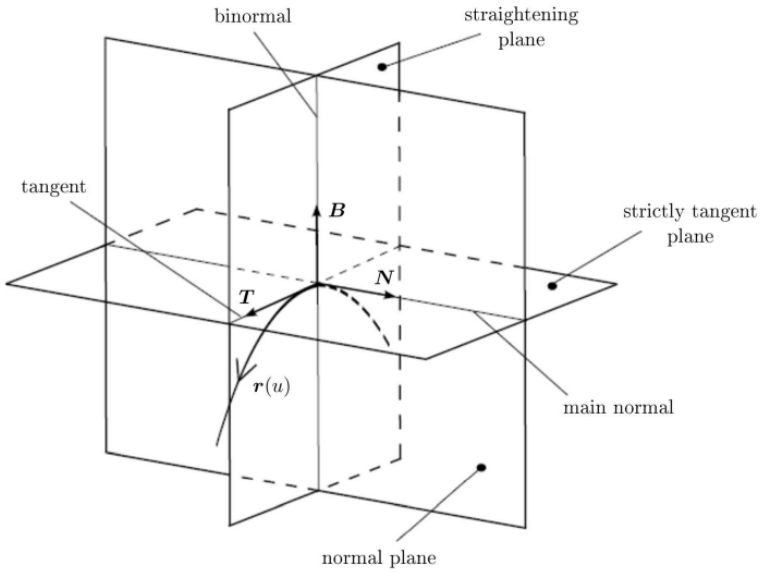


Figure 1: Frenet trihedron (based on [16])

$$\boldsymbol{B}(u) = \frac{\dfrac{\mathrm{d}\boldsymbol{r}(u)}{\mathrm{d}u} \times \dfrac{\mathrm{d}^2\boldsymbol{r}(u)}{\mathrm{d}u^2}}{\left\| \dfrac{\mathrm{d}\boldsymbol{r}(u)}{\mathrm{d}u} \times \dfrac{\mathrm{d}^2\boldsymbol{r}(u)}{\mathrm{d}u^2} \right\|}, \tag{6b}$$

$$\boldsymbol{N}(u) = \boldsymbol{B}(u) \times \boldsymbol{T}(u). \tag{6c}$$

Parameters which describe the curve geometry are named curvature κ and torsion τ [19]. Curvature defines the degree of the curve swerve from a straight line, whereas torsion expresses the curve swerve from a plain. They are defined by the following equations

$$\kappa(u) = \frac{\left\| \dfrac{\mathrm{d}\boldsymbol{r}(u)}{\mathrm{d}d} \times \dfrac{\mathrm{d}^2\boldsymbol{r}(u)}{\mathrm{d}u^2} \right\|}{\left\| \dfrac{\mathrm{d}\boldsymbol{r}(u)}{\mathrm{d}u} \right\|^3}, \tag{7a}$$

$$\tau(u) = \frac{\left\langle \dfrac{\mathrm{d}\boldsymbol{r}(u)}{\mathrm{d}u} \times \dfrac{\mathrm{d}^2\boldsymbol{r}(u)}{\mathrm{d}u^2}, \dfrac{\mathrm{d}^3\boldsymbol{r}(u)}{\mathrm{d}u^3} \right\rangle}{\left\| \dfrac{\mathrm{d}\boldsymbol{r}(u)}{\mathrm{d}u} \times \dfrac{\mathrm{d}^2\boldsymbol{r}(u)}{\mathrm{d}u^2} \right\|^2}, \tag{7b}$$

where $\langle \cdot, \cdot \rangle$ denotes a scalar product of vectors.

However, it is noteworthy that it is often more desirable to define a curve with respect to the parameter s, which is called *arclength* or *curvilinear distance* and describes how far from the beginning of a curve a point is located. This kind of parametrization is named normalized parametrization [19]. The curvilinear distance s is related to the parameter u by the equation

$$s(u) = \int_0^u \left\| \frac{\mathrm{d}\boldsymbol{r}(v)}{\mathrm{d}v} \right\| \mathrm{d}v. \tag{8}$$

Equation (8) leads to the definition of the curvilinear velocity

$$\dot{s} = \dot{u} \left\| \frac{\mathrm{d}\boldsymbol{r}(u)}{\mathrm{d}u} \right\|. \tag{9}$$

Considering these relations, versors defined by equations (6) can be expressed with respect to the arclength s [19]

$$\boldsymbol{T}(s) = \frac{\mathrm{d}\boldsymbol{r}(s)}{\mathrm{d}s}, \tag{10a}$$

$$\boldsymbol{N}(s) = \frac{\dfrac{\mathrm{d}\boldsymbol{T}(s)}{\mathrm{d}s}}{\left\| \dfrac{\mathrm{d}\boldsymbol{T}(s)}{\mathrm{d}s} \right\|}, \tag{10b}$$

$$\boldsymbol{B}(s) = \boldsymbol{T}(s) \times \boldsymbol{N}(s). \tag{10c}$$

Also, curvature and torsion may be described with the usage of the parameter s. Relations (7) are respectively transformed to equations [20]

$$\kappa(s) = \left\| \frac{\mathrm{d}\boldsymbol{T}(s)}{\mathrm{d}s} \right\|, \tag{11a}$$

$$\tau(s) = \left\| \frac{\mathrm{d}\boldsymbol{B}(s)}{\mathrm{d}s} \right\| = \frac{1}{\kappa^2(s)} \left\langle \frac{\mathrm{d}\boldsymbol{r}(s)}{\mathrm{d}s} \times \frac{\mathrm{d}^2\boldsymbol{r}(s)}{\mathrm{d}s^2}, \frac{\mathrm{d}^3\boldsymbol{r}(s)}{\mathrm{d}s^3} \right\rangle. \tag{11b}$$

It shows that normalized parametrization simplifies the curve definition. However, it is usually difficult to define an analytical model with respect to the arclength due to the non-linear relation defined by equation (8).

Derivatives of versors creating Frenet trihedron along the curve are crucial relations for changes of the curve geometry. They are defined by equations [19]

$$\frac{\mathrm{d}\boldsymbol{T}(s)}{\mathrm{d}s} = \kappa(s)\boldsymbol{N}(s), \tag{12a}$$

$$\frac{\mathrm{d}\boldsymbol{N}(s)}{\mathrm{d}s} = -\kappa(s)\boldsymbol{T}(s) + \tau(s)\boldsymbol{B}(s), \tag{12b}$$

$$\frac{\mathrm{d}\boldsymbol{B}(s)}{\mathrm{d}s} = -\tau(s)\boldsymbol{N}(s). \tag{12c}$$

The aforementioned equations (12) can be rewritten in a matrix form

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{S}(s)}{\mathrm{d}s} &= \left[\frac{\mathrm{d}\boldsymbol{T}(s)}{\mathrm{d}s} \;\; \frac{\mathrm{d}\boldsymbol{N}(s)}{\mathrm{d}s} \;\; \frac{\mathrm{d}\boldsymbol{B}(s)}{\mathrm{d}s}\right] \\ &= \left[\boldsymbol{T}(s) \;\; \boldsymbol{N}(s) \;\; \boldsymbol{B}(s)\right] \begin{bmatrix} 0 & -\kappa(s) & 0 \\ \kappa(s) & 0 & -\tau(s) \\ 0 & \tau(s) & 0 \end{bmatrix} = \boldsymbol{S}(s)\boldsymbol{W}(s),\end{aligned} \tag{13}$$

where $\boldsymbol{S}(s)$ is the rotation matrix describing orientation of the Serret–Frenet frame with respect to the inertial frame. The matrix $\boldsymbol{S}(s)$ is an orthogonal matrix and $\boldsymbol{S} \in \mathbb{SO}(3)$ [25], so $\boldsymbol{S}^T = \boldsymbol{S}^{-1}$. It is also worth noticing that matrix $\boldsymbol{W}(s)$ is skew-symmetric, i.e. the relation $\boldsymbol{W}^T(s) = -\boldsymbol{W}(s)$ is true. Derivative of the matrix $\boldsymbol{S}(s)$ with respect to the time parameter may be calculated using the chain formula

$$\dot{\boldsymbol{S}}(s) = \frac{\mathrm{d}\boldsymbol{S}(s)}{\mathrm{d}s}\dot{s} = \dot{s}\boldsymbol{S}(s)\boldsymbol{W}(s). \tag{14}$$

2.3. Non-orthogonal parametrization in $\mathbb{R}^3$ space

The local Serret–Frenet frame moving along a certain curve may be associated with a reference object, whose motion has to be imitated by a real object. In order to utilize information about the curve geometry included in the Serret–Frenet frame evolution, the control object model needs to be expressed with respect to the defined reference frame. The non-orthogonal projection method may be harnessed to project the control object local frame on the desired curve. Such approach does not implies any constraints of relative location of virtual and real objects. Thus, it is possible for the reference object to be situated ahead or behind the control object. Furthermore, it indicates that curvilinear velocity $\dot{s}$ of the reference frame may de defined arbitrarily. Illustration of the non-orthogonal projection on a certain curve is presented in Fig. 2. Coordinates of the point P, which describes the real robot location in the inertial frame, are equal to $\boldsymbol{p} = \begin{pmatrix} x & y & z \end{pmatrix}^T$. The vector $\boldsymbol{d} = \begin{pmatrix} d_1 & d_2 & d_3 \end{pmatrix}^T$ defines location of the same point in the local Serret–Frenet frame.

The controlled position of a robot in the inertial frame $\boldsymbol{p}$ is connected with its location in the local Serret–Frenet frame $\boldsymbol{d}$ with the following relation

$$\boldsymbol{p} = \boldsymbol{S}\boldsymbol{d} + \boldsymbol{r}, \tag{15}$$

where $\boldsymbol{r}$ is the vector describing the virtual robot location on the curve with respect to the inertial frame. Equation (15) may be transformed in order to derive

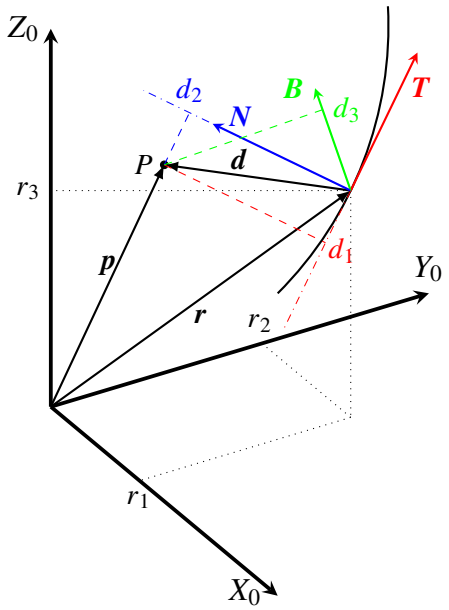


Figure 2: Non-orthogonal projection in $\mathbb{R}^3$ space

relations describing position errors of the control object with respect to the local Serret–Frenet frame moving along a path

$$\boldsymbol{d} = \boldsymbol{S}^T(\boldsymbol{p} - \boldsymbol{r}) = \begin{pmatrix} \langle \boldsymbol{T}, \boldsymbol{p} - \boldsymbol{r} \rangle \\ \langle \boldsymbol{N}, \boldsymbol{p} - \boldsymbol{r} \rangle \\ \langle \boldsymbol{B}, \boldsymbol{p} - \boldsymbol{r} \rangle \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}. \tag{16}$$

In fact, the vector $\boldsymbol{d}$ may be treated as a set of path following errors. If they tend toward zero, the desired motion of the robot is performed. Their dynamics may be obtained directly from equation (16) by differentiating it with time

$$\dot{\boldsymbol{d}} = \boldsymbol{S}^T(\dot{\boldsymbol{p}} - \dot{\boldsymbol{r}}) + \dot{\boldsymbol{S}}^T(\boldsymbol{p} - \boldsymbol{r}). \tag{17}$$

The element $\boldsymbol{S}^T\dot{\boldsymbol{r}}$ describes linear velocities in the body frame of the virtual robot [25], and its definition is consistent with the equation

$$\boldsymbol{S}^T\dot{\boldsymbol{r}} = \begin{pmatrix} \left\langle \boldsymbol{T}, \dfrac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}u} \right\rangle \\ \left\langle \boldsymbol{N}, \dfrac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}u} \right\rangle \\ \left\langle \boldsymbol{B}, \dfrac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}u} \right\rangle \end{pmatrix} \dot{u}. \tag{18}$$

Consideration of equation (6a) in the above relation leads to

$$\boldsymbol{S}^T\dot{\boldsymbol{r}} = \begin{pmatrix} \langle \boldsymbol{T}, \boldsymbol{T} \rangle \\ \langle \boldsymbol{N}, \boldsymbol{T} \rangle \\ \langle \boldsymbol{B}, \boldsymbol{T} \rangle \end{pmatrix} \dot{u} \left\| \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}u} \right\|. \tag{19}$$

Taking into account equation (9) and the fact that $\{\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{B}\}$ is the orthonormal basis in $\mathbb{R}^3$, equation (19) can be rewritten as

$$\boldsymbol{S}^T\dot{\boldsymbol{r}} = \begin{pmatrix} \dot{s} & 0 & 0 \end{pmatrix}^T. \tag{20}$$

By implementing equations (4), (14), (16) and (20) into equation (17) the following form of tracking error dynamics may be achieved

$$\dot{\boldsymbol{d}} = \boldsymbol{S}^T\boldsymbol{J}\dot{\boldsymbol{q}} - \begin{pmatrix} \dot{s} \\ 0 \\ 0 \end{pmatrix} - \dot{s}\boldsymbol{W}\boldsymbol{S}^T(\boldsymbol{p} - \boldsymbol{r}) = \boldsymbol{S}^T\boldsymbol{J}\dot{\boldsymbol{q}} - \begin{pmatrix} \dot{s} \\ 0 \\ 0 \end{pmatrix} - \dot{s}\boldsymbol{W}\boldsymbol{d}. \tag{21}$$

Considering definition of the matrix $\boldsymbol{W}$ from equation (13), the final form of the tracking error dynamics may be derived

$$\dot{\boldsymbol{d}} = \boldsymbol{S}^T\boldsymbol{J}\dot{\boldsymbol{q}} - \begin{pmatrix} \dot{s} \\ 0 \\ 0 \end{pmatrix} - \dot{s}\begin{pmatrix} -\kappa d_2 \\ \kappa d_1 - \tau d_3 \\ \tau d_2 \end{pmatrix} = \boldsymbol{G}\dot{\boldsymbol{q}} + \boldsymbol{F}. \tag{22}$$

3. Control problem formulation

The considered control problem is following a smooth path in $\mathbb{R}^3$ space by a holonomic non-redundant manipulator with partially known dynamics. The desired path has to be located in the workspace of the considered fixed-base manipulator. Thus, a certain geometric curve is given in $\mathbb{R}^3$ space and it has to be tracked by the control object.

The main impact of the work consists in showing that the task defined by path following requires the introduction of additional equations to the description of the object which constrain the control object motion and enforce path following.

The main aim of the control algorithm is therefore to bring the real robot to the origin of the local Serret–Frenet frame, which moves along the given curve. Such approach results in the necessity of considering some additional equations. These relations are described by equation (22), which may be treated similarly to non-holonomic constraints

$$\dot{\boldsymbol{d}} = \boldsymbol{G}\dot{\boldsymbol{q}} + \boldsymbol{F}. \tag{23}$$

Moreover, the robot motion is controlled by the partially known dynamics

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}(\boldsymbol{q}) = \boldsymbol{M}_0(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_0(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{D}_0(\boldsymbol{q}) + \boldsymbol{Y}(\ddot{\boldsymbol{q}},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}},\boldsymbol{q})\boldsymbol{a} = \boldsymbol{u}. \tag{24}$$

Equations (23) and (24) form a cascaded system with two stages: the first stage is created with constraint equations and the second one consists of the dynamics. Hence, a cascade structure is adopted by the considered system.

Control signals are generated so that the manipulator motion is performed under the constraints of the curve geometry. A method considered for solving this kind of problems is called the backstepping integrator algorithm [12].

The backstepping integrator algorithm may be applied to control a dynamic system with a cascade structure. The essential feature of such a system is the fact that its subsystems may be treated independently. As a result, the design process of the control law is recursive, starting from the most inner subsystem.

In the considered problem such approach results in division of the control system into two parallel performed parts: kinematic controller and dynamic controller. Equations (23) and (24) define the path following error dynamics and the object dynamics, respectively. They form a cascaded structure of the model, which is presented in Fig. 3.

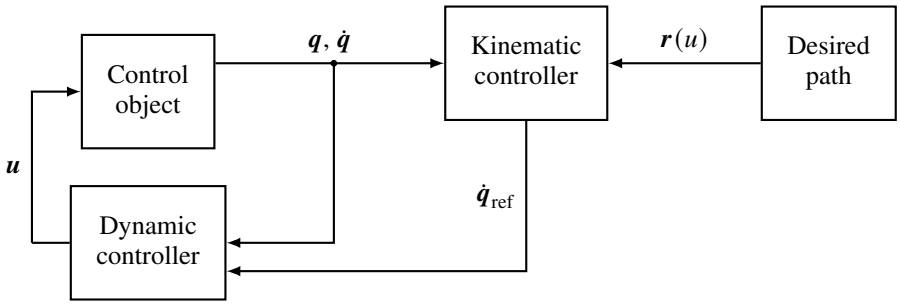


Figure 3: Control system structure

The error dynamics (23) defines the first stage of the cascade. The input of this subsystems is joint velocity $\dot{\boldsymbol{q}}$. The kinematic controller is responsible for generating velocity profiles $\dot{\boldsymbol{q}}_{\mathrm{ref}}$, which allow to move from the current manipulator configuration to the desired configuration, so that constraints resulting from the path geometry are preserved all the time. However, direct control of the manipulator position in the Serret–Frenet frame is impossible due to the cascade structure of the considered dynamic system. The dynamics (24) forms the second stage of the cascade. Hence, the dynamic controller needs to be designed. It allows to perform the desired reference velocities by generating proper control torques $\boldsymbol{u}$.

4. Control law

In the following section two stages of the control system cascade are presented.

4.1. Kinematic controller

Equations of tracking error dynamics obtained with the non-orthogonal projection of a control object on the desired path may be expressed in an affine form

$$\dot{\boldsymbol{d}} = \boldsymbol{G}\dot{\boldsymbol{q}} + \boldsymbol{F}. \tag{25}$$

Based on equation (25) a kinematic controller is proposed. Its main aim is to generate proper joint velocity profiles

$$\dot{\boldsymbol{q}}_{\text{ref}} = \boldsymbol{G}^{-1}(\boldsymbol{\chi} - \boldsymbol{F}). \tag{26}$$

The feedback loop is closed by putting equation (26) into equation (25). As a result, the integrator is defined

$$\dot{\boldsymbol{d}} = \boldsymbol{\chi}, \tag{27}$$

where $\boldsymbol{\chi}$ plays a role of a new control input. For the system (27) the control law is defined as

$$\boldsymbol{\chi} = \dot{\boldsymbol{d}}_d - \boldsymbol{K}_k \boldsymbol{e}_{\boldsymbol{d}}, \tag{28}$$

where $\boldsymbol{d}_d(t)$ is the time-dependent desired trajectory of path following errors, $\boldsymbol{e}_{\boldsymbol{d}} = \boldsymbol{d} - \boldsymbol{d}_d$ is the vector of errors of performing the desired values $\boldsymbol{d}_d$, and $\boldsymbol{K}_k = \text{diag}\{k_k\}$ is the gain matrix. The control defined by equation (28) provides error $\boldsymbol{e}_{\boldsymbol{d}}$ convergence to zero. Thus, the control object performs the desired motion along the given curve.

Proof. Taking into account the control law (28), the system (27) in the closed feedback loop may be rewritten as

$$\dot{\boldsymbol{e}}_{\boldsymbol{d}} + \boldsymbol{K}_k \boldsymbol{e}_{\boldsymbol{d}} = \boldsymbol{0}, \tag{29}$$

where $\boldsymbol{K}_k = \boldsymbol{K}_k^T > 0$. For the system (29) the following Lyapunov-like function is proposed

$$V_1(\boldsymbol{e}_{\boldsymbol{d}}) = \frac{1}{2}\boldsymbol{e}_{\boldsymbol{d}}^T \boldsymbol{e}_{\boldsymbol{d}}\,. \tag{30}$$

Its time derivative calculated along solutions of the system (29) is given as

$$\dot{V}_1(\boldsymbol{e}_{\boldsymbol{d}}) = \boldsymbol{e}_{\boldsymbol{d}}^T \dot{\boldsymbol{e}}_{\boldsymbol{d}} = \boldsymbol{e}_{\boldsymbol{d}}^T(-\boldsymbol{K}_k \boldsymbol{e}_{\boldsymbol{d}}) = -\boldsymbol{e}_{\boldsymbol{d}}^T \boldsymbol{K}_k \boldsymbol{e}_{\boldsymbol{d}} = -W(\boldsymbol{e}_{\boldsymbol{d}}) \leqslant 0. \tag{31}$$

From the LaSalle invariance principle [2], convergence of the errors $\boldsymbol{e}_{\boldsymbol{d}}$ to the set

$$W(\boldsymbol{e}_{\boldsymbol{d}}) = 0 \tag{32}$$

can be concluded. The system is asymptotically stable with zero equilibrium point for the positive-definite gain matrix $\boldsymbol{K}_k$. Hence, the choice of positive regulation gains guarantees convergence $\boldsymbol{d} \rightarrow \boldsymbol{d}_d$, which results in correct path following. This completes the proof for the first stage of cascade. □

4.2. Dynamic controller

The dynamic controller is the second stage of the control system cascade. Its main aim is to follow the desired velocity profiles $\dot{\boldsymbol{q}}_{\text{ref}}$. The following control algorithm, which is modification of the control law given by [22], based on the partially known dynamic model is proposed

$$\boldsymbol{u} = \boldsymbol{M}_0(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\text{ref}} + \boldsymbol{C}_0(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\text{ref}} + \boldsymbol{D}_0(\boldsymbol{q}) + \boldsymbol{Y}_r(\ddot{\boldsymbol{q}}_{\text{ref}},\dot{\boldsymbol{q}}_{\text{ref}},\dot{\boldsymbol{q}},\boldsymbol{q})\hat{\boldsymbol{a}}(t) - \boldsymbol{K}_d\boldsymbol{e}_{\dot{\boldsymbol{q}}}\,, \quad (33)$$

where $\dot{\boldsymbol{q}}_{\text{ref}}$ is the vector of reference joint velocities defined according to equation (26), $\boldsymbol{e}_{\dot{\boldsymbol{q}}} = \dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_{\text{ref}}$ is the vector of errors of performing reference joint velocities, $\boldsymbol{K}_d = \text{diag}\{k_d\}$ is the positive-definite matrix of regulation coefficients, $\boldsymbol{Y}_r$ is the regression matrix based on the reference joint velocities, and $\hat{\boldsymbol{a}}(t)$ is the estimation of the unknown parameters derived with the usage of the parameter adaptation law

$$\dot{\widetilde{\boldsymbol{a}}}(t) = \dot{\hat{\boldsymbol{a}}}(t) = -\boldsymbol{\Gamma}\boldsymbol{Y}_r^T(\ddot{\boldsymbol{q}}_{\text{ref}},\dot{\boldsymbol{q}}_{\text{ref}},\dot{\boldsymbol{q}},\boldsymbol{q})\boldsymbol{e}_{\dot{\boldsymbol{q}}}\,, \quad (34)$$

where $\widetilde{\boldsymbol{a}}(t) = \hat{\boldsymbol{a}}(t) - \boldsymbol{a}$ is the difference between the estimated values and the unknown constant real values of parameters, and $\boldsymbol{\Gamma} = \text{diag}\{\gamma\}$ is a positive-definite matrix of estimator coefficients. This control law guarantees error convergence to zero. The proof is presented below.

Proof. The control law (33) is applied to equation describing dynamics of a manipulator (24). The equation defining the system in the closed feedback loop is derived

$$\boldsymbol{M}_0(\boldsymbol{q})\dot{\boldsymbol{e}}_{\dot{\boldsymbol{q}}} + \boldsymbol{C}_0(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{e}_{\dot{\boldsymbol{q}}} + \boldsymbol{Y}(\ddot{\boldsymbol{q}},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}},\boldsymbol{q})\boldsymbol{a} - \boldsymbol{Y}_r(\ddot{\boldsymbol{q}}_{\text{ref}},\dot{\boldsymbol{q}}_{\text{ref}},\dot{\boldsymbol{q}},\boldsymbol{q})\hat{\boldsymbol{a}} + \boldsymbol{K}_d\boldsymbol{e}_{\dot{\boldsymbol{q}}} = \boldsymbol{0}. \quad (35)$$

Equation (35) may be extended to the form (for the transparency of the notation the matrix arguments are neglected)

$$\begin{aligned}
&\boldsymbol{M}_0\dot{\boldsymbol{e}}_{\dot{\boldsymbol{q}}} + \boldsymbol{C}_0\boldsymbol{e}_{\dot{\boldsymbol{q}}} + \boldsymbol{Y}\boldsymbol{a} - \boldsymbol{Y}_r\boldsymbol{a} + \boldsymbol{Y}_r\boldsymbol{a} - \boldsymbol{Y}_r\hat{\boldsymbol{a}} + \boldsymbol{K}_d\boldsymbol{e}_{\dot{\boldsymbol{q}}} \\
&\quad = \boldsymbol{M}_0\dot{\boldsymbol{e}}_{\dot{\boldsymbol{q}}} + \boldsymbol{C}_0\boldsymbol{e}_{\dot{\boldsymbol{q}}} + \boldsymbol{M}_a\dot{\boldsymbol{e}}_{\dot{\boldsymbol{q}}} + \boldsymbol{C}_a\boldsymbol{e}_{\dot{\boldsymbol{q}}} - \boldsymbol{Y}_r\widetilde{\boldsymbol{a}} + \boldsymbol{K}_d\boldsymbol{e}_{\dot{\boldsymbol{q}}} \\
&\quad = \boldsymbol{M}(\boldsymbol{q},\boldsymbol{a})\dot{\boldsymbol{e}}_{\dot{\boldsymbol{q}}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{a})\boldsymbol{e}_{\dot{\boldsymbol{q}}} - \boldsymbol{Y}_r(\ddot{\boldsymbol{q}}_{\text{ref}},\dot{\boldsymbol{q}}_{\text{ref}},\dot{\boldsymbol{q}},\boldsymbol{q})\widetilde{\boldsymbol{a}} + \boldsymbol{K}_d\boldsymbol{e}_{\dot{\boldsymbol{q}}} = \boldsymbol{0}.
\end{aligned} \quad (36)$$

For the systems (29) and (36) — closed-loop systems for each stage of the cascade — the Lyapunov function is defined as follows

$$V_2(\boldsymbol{e}_d,\boldsymbol{e}_{\dot{\boldsymbol{q}}},\widetilde{\boldsymbol{a}}) = V_1(\boldsymbol{e}_d) + \frac{1}{2}\boldsymbol{e}_{\dot{\boldsymbol{q}}}^T\boldsymbol{M}(\boldsymbol{q},\boldsymbol{a})\boldsymbol{e}_{\dot{\boldsymbol{q}}} + \frac{1}{2}\widetilde{\boldsymbol{a}}^T\boldsymbol{\Gamma}^{-1}\widetilde{\boldsymbol{a}}. \quad (37)$$

Due to the properties of function (30) and the considered matrices, function (37) fulfils conditions

$$V_2(\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}, \widetilde{\boldsymbol{a}}) > 0, \quad \text{for} \quad (\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}, \widetilde{\boldsymbol{a}}) \neq \boldsymbol{0},$$
$$V_2(\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}, \widetilde{\boldsymbol{a}}) = 0, \quad \text{for} \quad (\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}, \widetilde{\boldsymbol{a}}) = \boldsymbol{0}.$$

It is noteworthy that the Lyapunov function V_2 directly depends upon the error $\boldsymbol{e_d}$, which was proved to converge to zero in the previous section. Subsequently, derivative of the Lyapunov function (37) may be calculated

$$\dot{V}_2(\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}, \widetilde{\boldsymbol{a}}) = \dot{V}_1(\boldsymbol{e_d}) + \boldsymbol{e}_{\dot{\boldsymbol{q}}}^T \boldsymbol{M}(\boldsymbol{q}, \boldsymbol{a}) \dot{\boldsymbol{e}}_{\dot{\boldsymbol{q}}} + \frac{1}{2} \boldsymbol{e}_{\dot{\boldsymbol{q}}}^T \dot{\boldsymbol{M}}(\boldsymbol{q}, \boldsymbol{a}) \boldsymbol{e}_{\dot{\boldsymbol{q}}} + \widetilde{\boldsymbol{a}}^T \boldsymbol{\Gamma}^{-1} \dot{\widetilde{\boldsymbol{a}}}. \tag{38}$$

Taking into account relations (2), (34) and (36) in equation (38) results in

$$\dot{V}_2(\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}) = -\boldsymbol{e}_{\boldsymbol{d}}^T \boldsymbol{K}_k \boldsymbol{e_d} - \boldsymbol{e}_{\dot{\boldsymbol{q}}}^T \boldsymbol{K}_d \boldsymbol{e}_{\dot{\boldsymbol{q}}} \,. \tag{39}$$

Due to the properties of the function $\dot{V}_1(\boldsymbol{e_d})$, defined with equation (31), for the positive-definite matrix $\boldsymbol{K}_d$ the following relations hold

$$\dot{V}_2(\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}) < 0, \quad \text{for} \quad (\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}) \neq \boldsymbol{0},$$
$$\dot{V}_2(\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}) = 0, \quad \text{for} \quad (\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}) = \boldsymbol{0}.$$

By virtue of the LaSalle invariant principle [2], $(\boldsymbol{e_d}, \boldsymbol{e_{\dot{q}}}) = (\boldsymbol{0}, \boldsymbol{0})$ defines the asymptotically stable equilibrium point of the whole cascaded system. In other words, system moves along the desired path because $\boldsymbol{e_d} = \boldsymbol{0}$ and velocities applied to the manipulator joints converge to the desired reference profiles, $\dot{\boldsymbol{q}} \rightarrow \dot{\boldsymbol{q}}_{\text{ref}}$.

However, the derivative of the Lyapunov function (37) is independent from the parameter estimation error $\widetilde{\boldsymbol{a}}$. It means that the estimation $\hat{\boldsymbol{a}}$ may not converge to the real values, although the proposed controller allows to follow the desired path correctly. □

5. Simulation study

5.1. Control object

In the simulation study two different robots are considered: RTR manipulator with 3 degrees of freedom and Staubli RX-60 manipulator with 6 degrees of freedom. They are described in the following section.

5.1.1. RTR manipulator

In the considered example of path following in $\mathbb{R}^3$ space a stationary manipulator of three degrees of freedom (rotational, translational and rotational) is

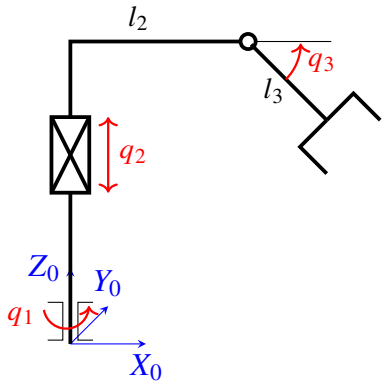


Figure 4: RTR manipulator structure

chosen as a control object. It is a holonomic and non-redundant robotic system. A scheme of the RTR manipulator structure is presented in Fig. 4. The dynamics of the robot is given by equation (1), where the following matrices are equal to [16]:

- the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} M_{11} & 0 & 0 \\ 0 & M_{22} & M_{23} \\ 0 & M_{23} & M_{33} \end{bmatrix}, \tag{40}$$

where

$$\begin{aligned} M_{11} &= \frac{1}{3}m_2 l_2^2 + m_3 l_2^2 + m_3 l_3 c_3^2 + m_3 l_2 l_3 c_3 \,, \\ M_{22} &= m_2 + m_3 \,, \\ M_{23} &= \frac{1}{2}m_3 l_2 l_3 c_3 \,, \\ M_{33} &= \frac{1}{3}m_3 l_3^2 \,, \end{aligned}$$

- the Coriolis matrix $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$

$$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} C_{11} & 0 & C_{13} \\ 0 & 0 & C_{23} \\ -C_{13} & 0 & 0 \end{bmatrix}, \tag{41}$$

where

$$\begin{aligned} C_{11} &= -\dot{q}_3 \left(\frac{1}{2}m_3 l_2 l_3 s_3 + \frac{1}{3}m_3 l_3^2 s_3 c_3 \right), \\ C_{13} &= -\dot{q}_1 \left(\frac{1}{2}m_3 l_2 l_3 s_3 + \frac{1}{3}m_3 l_3^2 s_3 c_3 \right), \end{aligned}$$

$$C_{23} = -\dot{q}_3 \frac{1}{2} m_3 l_2 l_3 s_3 \,,$$

- the gravity forces vector $\boldsymbol{D}(\boldsymbol{q})$

$$\boldsymbol{D}(\boldsymbol{q}) = \begin{pmatrix} 0 \\ (m_2 + m_3) g \\ \frac{1}{2} g m_3 l_3 c_3 \end{pmatrix}. \tag{42}$$

In equations above the notation convention is used: $s_i = \sin(q_i)$, $c_i = \cos(q_i)$. Moreover, g is the gravity acceleration, $\boldsymbol{q} = \begin{pmatrix} q_1 & q_2 & q_3 \end{pmatrix}^T$ is the vector of configuration variables defining position or angular position in particular joints, and m_i, l_i denote mass and length of the i-th manipulator link, respectively. It is assumed that two parameters of the dynamic model are unknown. Thus, the unknown parameter vector is defined as

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} m_3 l_3 \\ m_3 l_2 l_3 \end{pmatrix}. \tag{43}$$

Hence, the proper elements of the model (5) are defined as follows:

- the inertia matrix $\boldsymbol{M}_0(\boldsymbol{q})$

$$\boldsymbol{M}_0(\boldsymbol{q}) = \begin{bmatrix} \frac{1}{3} m_2 l_2^2 + m_3 l_2^2 & 0 & 0 \\ 0 & m_2 + m_3 & 0 \\ 0 & 0 & \frac{1}{3} m_3 l_3^2 \end{bmatrix}, \tag{44}$$

- the Coriolis matrix $\boldsymbol{C}_0(\boldsymbol{q}, \dot{\boldsymbol{q}})$

$$\boldsymbol{C}_0(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} -\dot{q}_3 \left(\frac{1}{3} m_3 l_3^2 s_3 c_3 \right) & 0 & -\dot{q}_1 \left(\frac{1}{3} m_3 l_3^2 s_3 c_3 \right) \\ 0 & 0 & 0 \\ \dot{q}_1 \left(\frac{1}{3} m_3 l_3^2 s_3 c_3 \right) & 0 & 0 \end{bmatrix}, \tag{45}$$

- the gravity forces vector $\boldsymbol{D}_0(\boldsymbol{q})$

$$\boldsymbol{D}_0(\boldsymbol{q}) = \begin{pmatrix} 0 \\ (m_2 + m_3) g \\ 0 \end{pmatrix}, \tag{46}$$

- the regression matrix $\boldsymbol{Y}(\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_x, \dot{\boldsymbol{q}}, \boldsymbol{q})$

$$\boldsymbol{Y}(\ddot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_x, \dot{\boldsymbol{q}}, \boldsymbol{q}) = \begin{bmatrix} \ddot{q}_1 c_3^2 & \ddot{q}_1 c_3 - \frac{1}{2}\dot{q}_{1x}\dot{q}_3 s_3 - \frac{1}{2}\dot{q}_{3x}\dot{q}_1 s_3 \\ 0 & \frac{1}{2}c_3\ddot{q}_3 - \frac{1}{2}\dot{q}_{3x}\dot{q}_3 s_3 \\ \frac{1}{2}g c_3 & \frac{1}{2}\ddot{q}_2 c_3 + \frac{1}{2}\dot{q}_{1x}\dot{q}_1 s_3 \end{bmatrix}. \tag{47}$$

Forward kinematics of the considered manipulator is defined with equation

$$\boldsymbol{p} = k(\boldsymbol{q}) = \begin{pmatrix} c_1(l_3 c_3 + l_2) \\ s_1(l_3 c_3 + l_2) \\ l_3 s_3 + q_2 \end{pmatrix}, \tag{48}$$

while the Jacobi matrix equals to

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} -s_1(l_3 c_3 + l_2) & 0 & -l_3 c_1 s_3 \\ c_1(l_3 c_3 + l_2) & 0 & -l_3 s_1 s_3 \\ 0 & 1 & l_3 c_3 \end{bmatrix}. \tag{49}$$

For simulation experiments it is assumed that $l_2 = 1.5$ m, $l_3 = 1$ m, $m_2 = 20$ kg, $m_3 = 20$ kg. The initial configuration of the manipulator is chosen

$$\boldsymbol{q}_0 = \left(\frac{\pi}{4}\ \text{rad} \quad 2\ \text{m} \quad -\frac{\pi}{2}\ \text{rad}\right)^T. \tag{50}$$

Zero initial joint velocities are also assumed.

5.1.2. Staubli RX-60 manipulator

Another robot considered in the simulation study is the Staubli RX-60 manipulator. It has six rotational degrees of freedom. Due to that fact, the vector $\boldsymbol{q}$ consists of six joint angular positions

$$\boldsymbol{q} = \begin{pmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{pmatrix}^T.$$

A structure of the manipulator is presented in Fig. 5. The local frames presented in the figure result from the Denavit–Hartenberg parameters, which are presented in Table 1.

Considered in Table 1 geometrical parameters are equal to: $l_1 = 0.237$ m, $l_2 = 0.29$ m, $l_3 = 0.237$ m, and $l_4 = 0.31$ m. They denote lengths of respective manipulator links. Based on D-H parameters the end-effector position with respect to the base frame may be derived. It is equal to

$$\boldsymbol{p} = k(\boldsymbol{q}) = \begin{pmatrix} -l_3 s_1 + c_1(l_2 c_2 + l_4 s_{23}) \\ l_3 c_1 + s_1(l_2 c_2 + l_4 s_{23}) \\ l_1 + l_4 c_{23} - l_2 s_2 \end{pmatrix}, \tag{51}$$

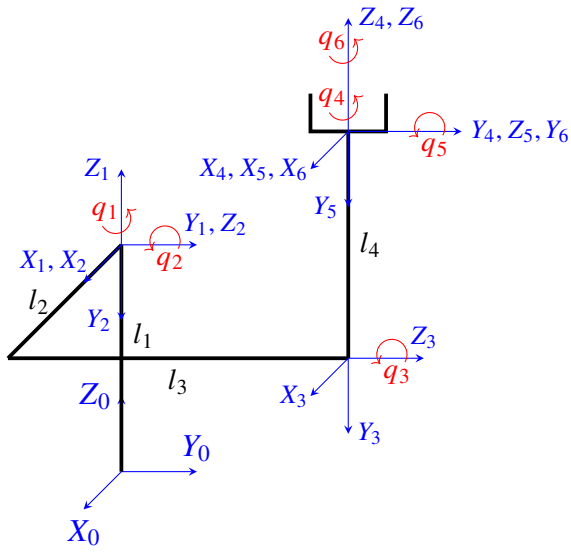


Figure 5: Staubli RX-60 manipulator structure

Table 1: D-H parameters of the Staubli RX-60 manipulator

Frame i	θ_i	d_i	a_i	α_i
1	q_1	l_1	0	$-\pi/2$
2	q_2	0	l_2	0
3	q_3	l_3	0	$\pi/2$
4	q_4	l_4	0	$-\pi/2$
5	q_5	0	0	$\pi/2$
6	q_6	0	0	0

where $s_{ij} = \sin(q_i + q_j)$ and $c_{ij} = \cos(q_i + q_j)$. It may be observed that three last rotational joints do not affect the position of the end-effector. As only position control is taken into account in the simulation study, we may assume without the generality loss that the rotational position in the three last joints are constant and equal to zero, i.e. $q_4 = q_5 = q_6 = 0$ rad.

The dynamic model and identified dynamic parameters of the manipulator are provided in [11]. The mean values of the identified parameters, provided in that paper, are used for the considered manipulator model in the simulation study.

It is assumed that for that case there are two unknown dynamic parameters. In [11] they are denoted as α_{15} and α_{16}. Thus, the unknown parameter vector $\boldsymbol{a}$

for the RX-60 manipulator takes the form

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} gm_2 l_{2_c} + g_0 m_3 l_2 \\ gm_3 l_{4_c} \end{pmatrix}, \tag{52}$$

where m_i denotes mass of the i-th link.

For the conducted simulations the initial configuration of the manipulator is chosen

$$\boldsymbol{q}_0 = \begin{pmatrix} 0 \text{ rad} & \frac{\pi}{6} \text{ rad} & \frac{3\pi}{8} \text{ rad} & 0 \text{ rad} & 0 \text{ rad} & 0 \text{ rad} \end{pmatrix}^T. \tag{53}$$

Again, joint velocities in the initial state are assumed to be equal to zero.

5.2. Considered curves

In the presented simulation study two types of curves were investigated. The first one is a helix which is described with the following equation [19]

$$\boldsymbol{\alpha}(s) = \begin{pmatrix} a\cos\frac{s}{c} & a\sin\frac{s}{c} & \frac{bs}{c} \end{pmatrix}^T, \tag{54}$$

where a, b, c are certain positive constants and $c = \sqrt{a^2 + b^2}$. The equation of the helix (54) is expressed with respect to the arclength s, so its parametrization is normalized. It is noteworthy that both curvature and torsion of the helix are constant values.

Another considered curve is a tapering helix. Its equation may be derived with a proper modification of the general helix equation. The first two elements are multiplied by an exponential function which leads to the equation

$$\boldsymbol{\beta}(u) = \begin{pmatrix} \mathrm{e}^{-uw} a\cos u & \mathrm{e}^{-uw} a\sin u & bu \end{pmatrix}^T, \tag{55}$$

where $w > 0$ is a certain positive value. The explicit relation $u(s)$, which is essential to define the normalized parametrization, is quite difficult to obtain from equation (8). Thus, the tapering helix (55) is defined with respect to a general parameter u. It is also worth noticing that curvature and torsion do not preserve constant values during the local frame evolution along the curve.

5.3. Simulation results

In the following section results of numerical simulations for the considered robots are presented.

5.3.1. RTR manipulator

In the conducted experiments the following parameters of the desired paths were assumed: $a = 1$, $b = 1$, $w = 1/300$. Moreover, two pairs of the regulation coefficients were chosen: $\{k_k = 10,\ k_d = 30\}$ and $\{k_k = 100,\ k_d = 100\}$. The estimator coefficient was chosen from the set $\gamma \in \{10,\ 100\}$. For all simulation cases the desired trajectory of path following errors is equal to

$$\boldsymbol{d}_d(t) = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T, \tag{56}$$

so the value $\boldsymbol{d}$ is equal to the path tracking error $\boldsymbol{e_d}$.

Firstly, tracking the virtual robot moving along the curve $\boldsymbol{\alpha}(s)$ defined by equation (54) is considered. The curvilinear velocity is arbitrarily chosen as

$$\dot{s} = 2\ \frac{\mathrm{m}}{\mathrm{s}}. \tag{57}$$

Figure 6 presents the desired path $\boldsymbol{\alpha}(s)$ and the path followed by the manipulator for the regulation coefficients equal to $\{k_k = 10,\ k_d = 30,\ \gamma = 100\}$. Results achieved for other simulations are analogical and are omitted in order not to reduce figure readability. Figs. 7, 8 and 9 show all trajectories of the path following errors e_{d_1}, e_{d_2}, e_{d_3}, respectively. The figures marked with (a) present the behaviour of the system during the first 10 s of the simulation. In turn, the figures marked with (b) show the simulation results for the whole time (100 s for this case), but the error trajectories are zoomed to very low order of magnitude.

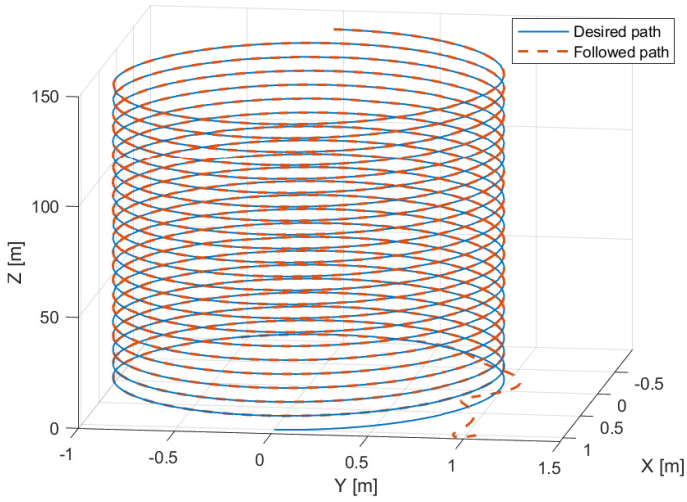
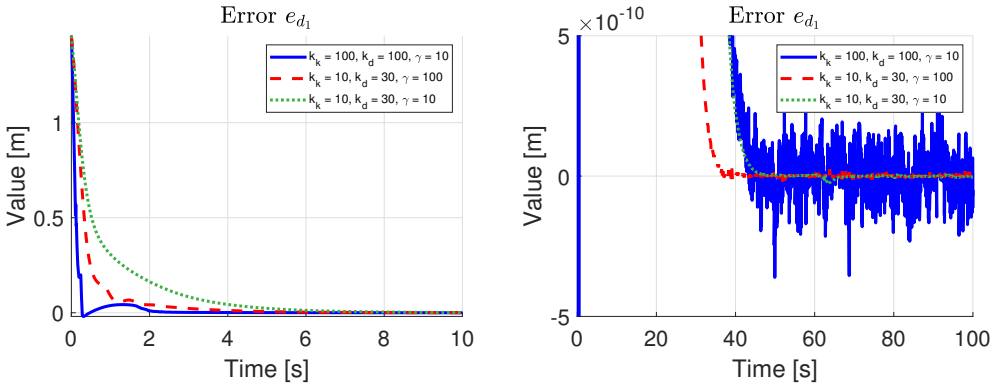
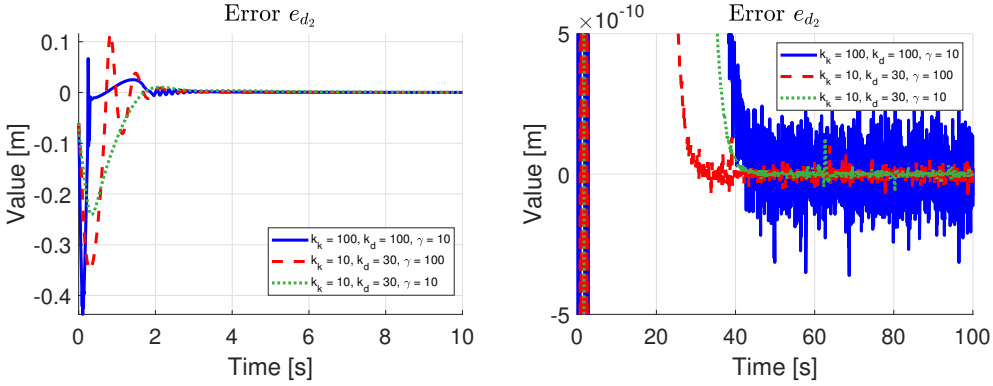


Figure 6: The desired path $\boldsymbol{\alpha}(s)$ and the path followed by the RTR manipulator for $k_k = 10$, $k_d = 30$, $\gamma = 100$



(a) Trajectory of e_{d_1} error (b) Zoom of the trajectory of e_{d_1} error

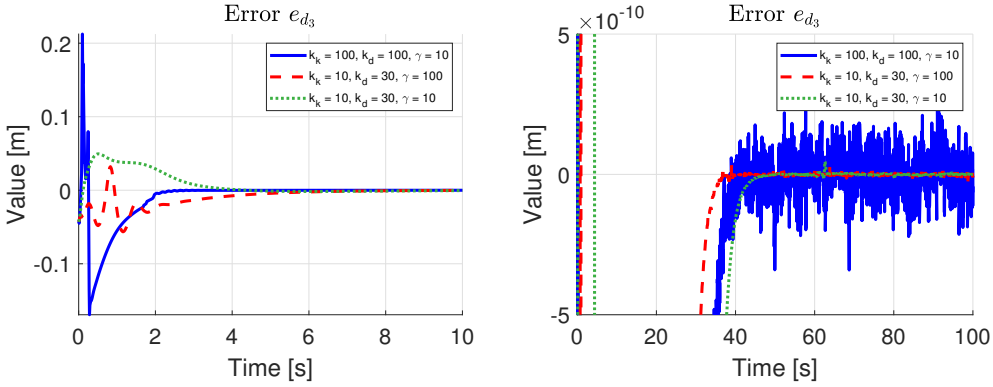
Figure 7: Path following error e_{d_1} for the helix $\alpha(s)$ – RTR manipulator (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 10$; Dashed line: $k_k = 10$, $k_d = 30$, $\gamma = 100$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 10$)



(a) Trajectory of e_{d_2} error (b) Zoom of the trajectory of e_{d_2} error

Figure 8: Path following error e_{d_2} for the helix $\alpha(s)$ – RTR manipulator (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 10$; Dashed line: $k_k = 10$, $k_d = 30$, $\gamma = 100$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 10$)

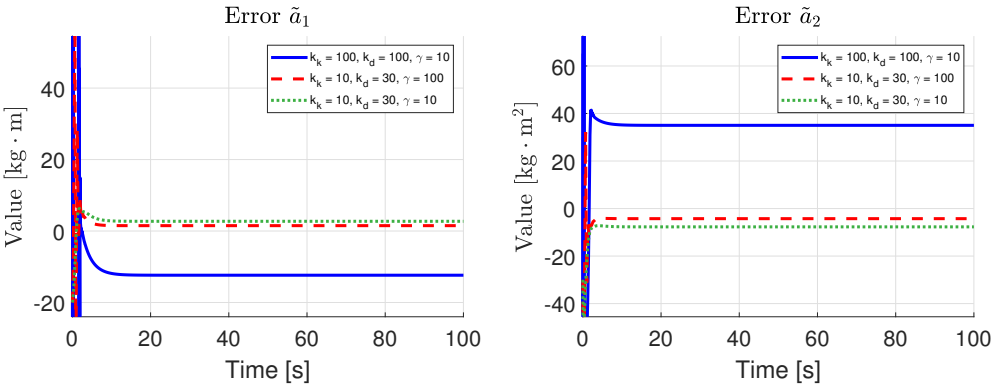
Presented graphs prove that the proposed algorithm guarantees asymptotic convergence of errors to zero. The manipulator correctly followed the path defined in the normalized parametrization. It is noteworthy that higher value of estimator coefficient guaranteed faster convergence of path tracking errors. On the other hand, higher values of kinematic regulator k_k and dynamic regulator k_d coefficients led to higher oscillations at the computational precision level. Moreover, they resulted in more rapid reaction of the system at the beginning of the manipulator motion.



(a) Trajectory of e_{d_3} error (b) Zoom of the trajectory of e_{d_3} error

Figure 9: Path following error e_{d_3} for the helix $\boldsymbol{\alpha}(s)$ – RTR manipulator (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 10$; Dashed line: $k_k = 10$, $k_d = 30$, $\gamma = 100$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 10$)

In Fig. 10 errors of estimation of the unknown parameters are presented. It may be observed that for all considered simulation cases estimated values do not converge to the real values. Different values are generated, depending on the choice of regulation gains. Despite that fact, the path is followed correctly. It is noteworthy that higher gains may lead to faster changes of the system state at the beginning of the manipulator motion.



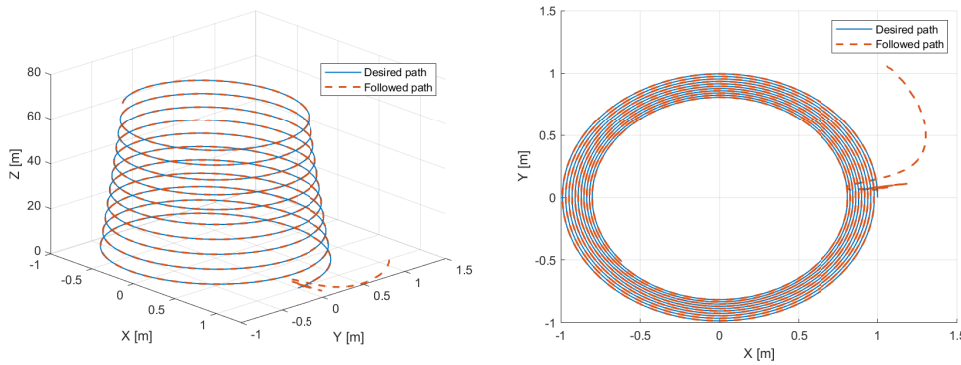
(a) Trajectory of $\widetilde{a}_1$ error (b) Trajectory of $\widetilde{a}_2$ error

Figure 10: Errors of estimation of the unknown parameters $\boldsymbol{a}$ for the helix $\boldsymbol{\alpha}(s)$ – RTR manipulator (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 10$; Dashed line: $k_k = 10$, $k_d = 30$, $\gamma = 100$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 10$)

Secondly, following the path $\boldsymbol{\beta}(u)$ defined by equation (55) was investigated. This curve is not expressed in the normalized Serret–Frenet parametrization. Due to that fact, the local frame velocity along the curve is defined as evolution of an independent parameter u. It is defined as

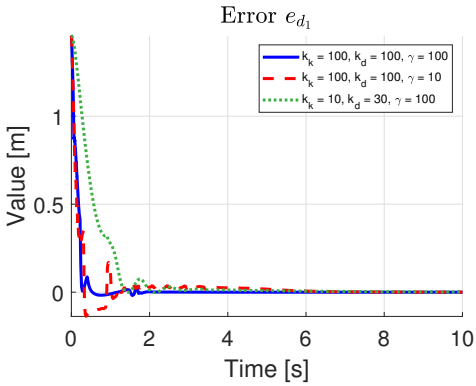
$$\dot{u} = \frac{1}{150} \frac{\text{m}}{\text{s}}. \tag{58}$$

Figure 11 presents the desired path $\boldsymbol{\beta}(u)$ and the path followed by the manipulator for the regulation coefficients $\{k_k = 10,\ k_d = 30,\ \gamma = 100\}$. The view on the paths from the top is also shown. Again, for other regulation parameters analogical results were achieved. It is worth noticing that the path is similar to the helix but it becomes narrower and narrower with the increase of the parameter u. Due to constraints of the manipulator workspace and control object geometrical properties, the curve cannot get too narrow. If the path tapers too much, it may be unobtainable by the manipulator. However, it does not result from the algorithm deficiencies. The control task may not be performed correctly as a consequence of the system constraints, i.e. the manipulator workspace limitations. Figures 12, 13 and 14 show trajectories of the path following errors e_{d_1}, e_{d_2}, e_{d_3}, respectively. Again, error convergence to zero is achieved even though the local Serret–Frenet frame moves at different rate along the curve and geometrical parameters of the curve κ and τ change in time. In the presented figures it may be observed that error values converge to zero much slower than for the helix. Path tracking errors are satisfactorily low even after 10 s of the simulation, which is presented in the figures marked with (a). However, the convergence to the precision level of the numerical computations is reached even after thousands of seconds. We extended simulation time to even 10^4 s and the results are shown in the figures marked

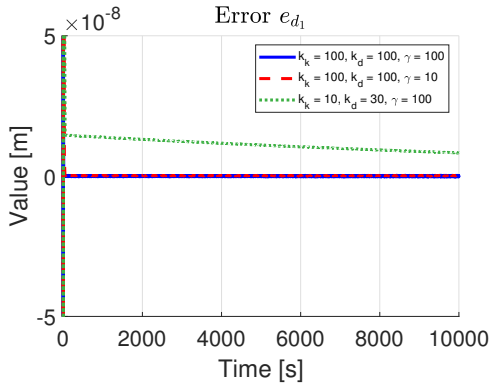


(a) General view (b) View from the top

Figure 11: The desired path $\boldsymbol{\beta}(u)$ and the path followed by the RTR manipulator for $k_k = 10$, $k_d = 30$, $\gamma = 100$



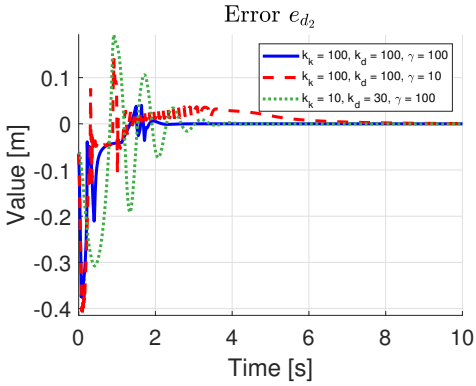
(a) Trajectory of e_{d_1} error



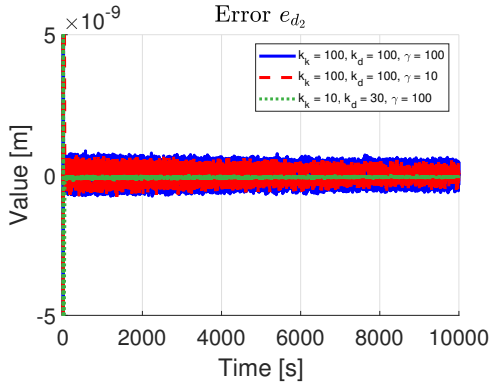
(b) Zoom of the trajectory of e_{d_1} error

Figure 12: Path following error e_{d_1} for the tapering helix $\boldsymbol{\beta}(u)$ (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 100$; Dashed line: $k_k = 100$, $k_d = 100$, $\gamma = 10$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 100$)

with (b). It is worth noticing that low controller gains emphasize the asymptotic characteristics of the proposed algorithm. Moreover, higher regulation coefficients improve error convergence, although they lead to higher oscillations.



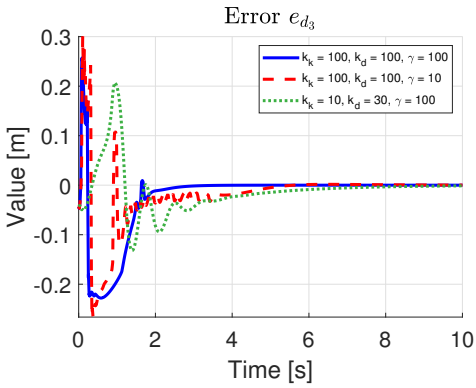
(a) Trajectory of e_{d_2} error



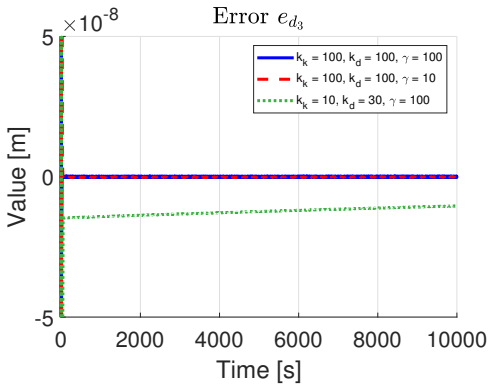
(b) Zoom of the trajectory of e_{d_2} error

Figure 13: Path following error e_{d_2} for the tapering helix $\boldsymbol{\beta}(u)$ (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 100$; Dashed line: $k_k = 100$, $k_d = 100$, $\gamma = 10$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 100$)

In Figs. 15 and 16 estimation errors $\widetilde{\boldsymbol{a}}$ are presented. Again, no convergence to the true values is obtained. It does not influence the correctness of path tracking. For higher regulation coefficients the estimated values are closer to the real ones. It is worth noticing that for the tapering helix error $\widetilde{a}_1$ does not



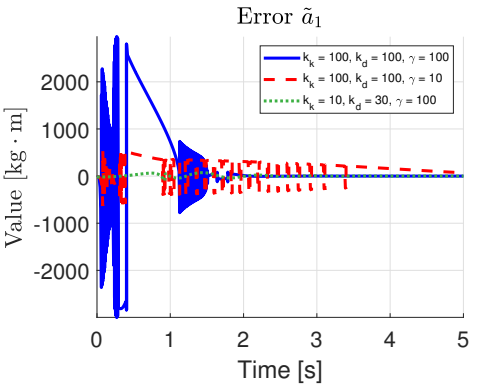
(a) Trajectory of e_{d_3} error



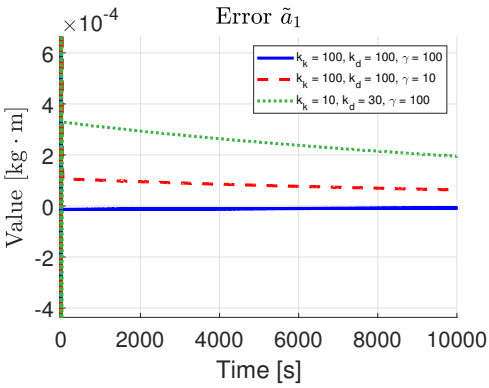
(b) Zoom of the trajectory of e_{d_3} error

Figure 14: Path following error e_{d_3} for the tapering helix $\boldsymbol{\beta}(u)$ (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 100$; Dashed line: $k_k = 100$, $k_d = 100$, $\gamma = 10$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 100$)

stabilize even after 10^4 s, which may be seen in Fig. 15b. It may be the reason of the long time needed for convergence of path tracking errors and results from low controller gains. This example shows that the proposed algorithm is asymptotically stable.



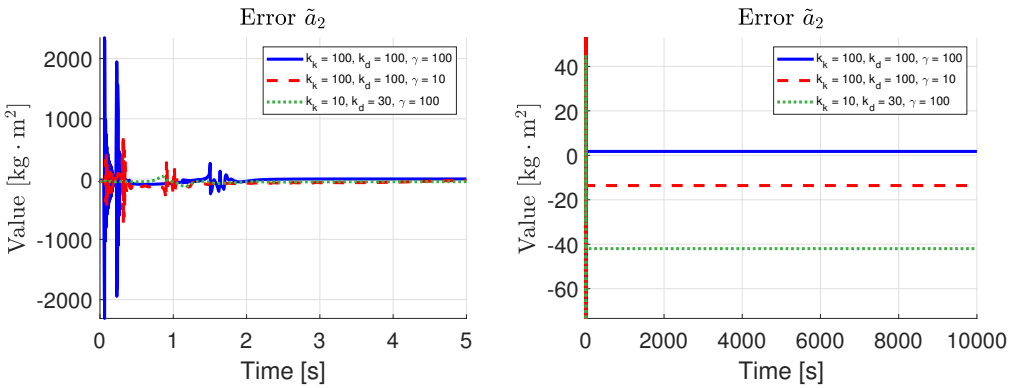
(a) Trajectory of $\widetilde{a}_1$ error



(b) Zoom of the trajectory of $\widetilde{a}_1$ error

Figure 15: Estimation error $\widetilde{a}_1$ for the tapering helix $\boldsymbol{\beta}(u)$ (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 100$; Dashed line: $k_k = 100$, $k_d = 100$, $\gamma = 10$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 100$)

For both considered curves different sets of the regulation coefficients were considered. It may be observed in the aforementioned figures that pace of the error convergence depends on the coefficient values of the gain matrices $\boldsymbol{K}_k$ and



(a) Trajectory of $\widetilde{a}_2$ error (b) Zoom of the trajectory of $\widetilde{a}_2$ error

Figure 16: Estimation error $\widetilde{a}_2$ for the tapering helix $\boldsymbol{\beta}(u)$ (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 100$; Dashed line: $k_k = 100$, $k_d = 100$, $\gamma = 10$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 100$)

$\boldsymbol{K}_d$. Also, the matrix $\boldsymbol{\Gamma}$ influences the rate of unknown parameter stabilization. The higher values, the faster rate of the error convergence to zero. Thus, the time of reaching the desired path may be regulated by change of proper parameters.

5.3.2. Staubli RX-60 manipulator

The path following simulation was also conducted for the Staubli RX-60 manipulator. Considering another control object is valuable as motion along a cylindrical helix is natural for the RTR manipulator. It is a possibility to verify proposed algorithms for a manipulator with a kinematic structure like the RX-60 manipulator.

In order to satisfy the requirement that the desired path is located in the manipulator workspace, the helix parameters must be changed in comparison to the path considered in the previous section. Hence, the following values of the given path were assumed in the simulation study: $a = 0.5$ and $b = 0.05$. Moreover, the controller gains were taken from the sets: $k_k \in \{10, 100\}$, $k_d \in \{30, 100\}$ and $\gamma \in \{30, 100\}$. The values were mixed and 3 cases were considered. Moreover, the velocity profile along the path was defined as

$$\dot{s} = 0.04 \, \frac{\mathrm{m}}{\mathrm{s}} \, . \tag{59}$$

The desired position of the end-effector with respect to the local Serret–Frenet frame is again defined as zero vector, i.e. $\boldsymbol{d}_d = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$, as we want the robot end-effector to move precisely along the given path.

In Fig. 17 the desired path and the path followed by the manipulator is presented. Also, the robot is visualized in a certain state while performing the task. It may be concluded that the path is tracked correctly.

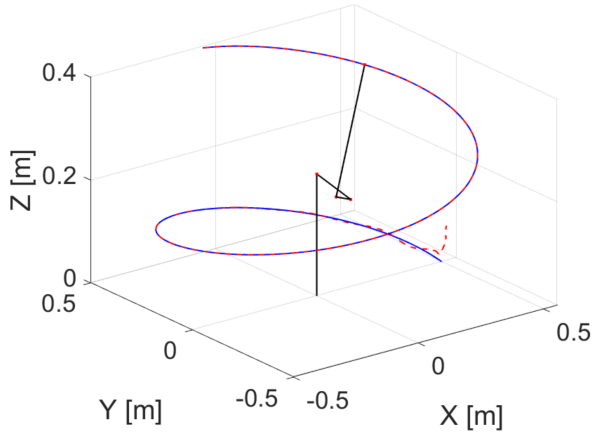
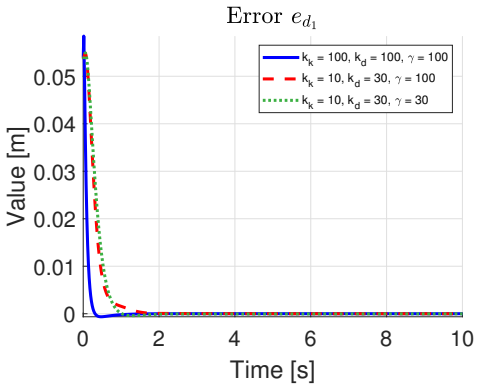
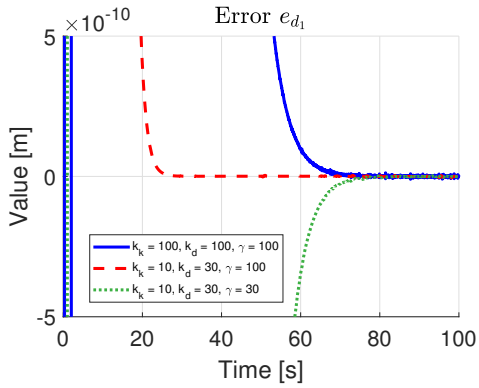


Figure 17: The desired path $\alpha(s)$ (solid line) and the path followed by the Staubli RX-60 manipulator (dashed line) for $k_k = 10$, $k_d = 30$, $\gamma = 100$

The next figures, Figs. 18–20, present the path following errors. The figures marked with (a) present error graphs during the first 10 s of the simulation. In turn, the figures marked with (b) show the error trajectories at low order of magnitude, but for the whole simulation which lasted 100 s. The presented figures emphasized asymptotic convergence of errors to zero. They confirm that



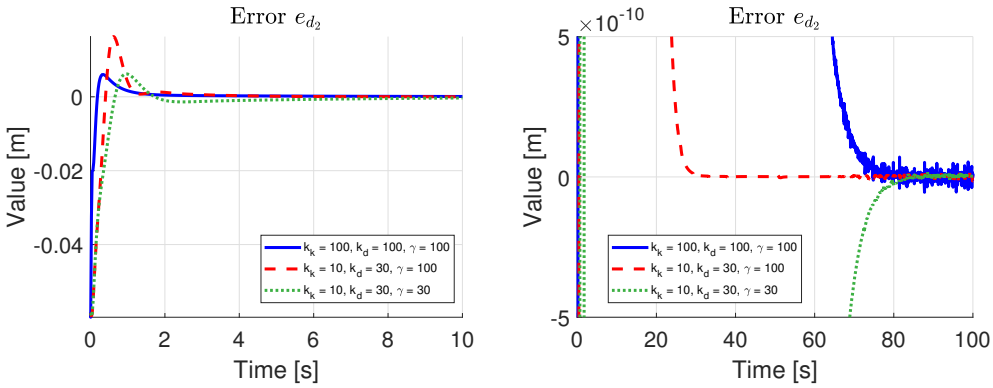
(a) Trajectory of e_{d_1} error



(b) Zoom of the trajectory of e_{d_1} error

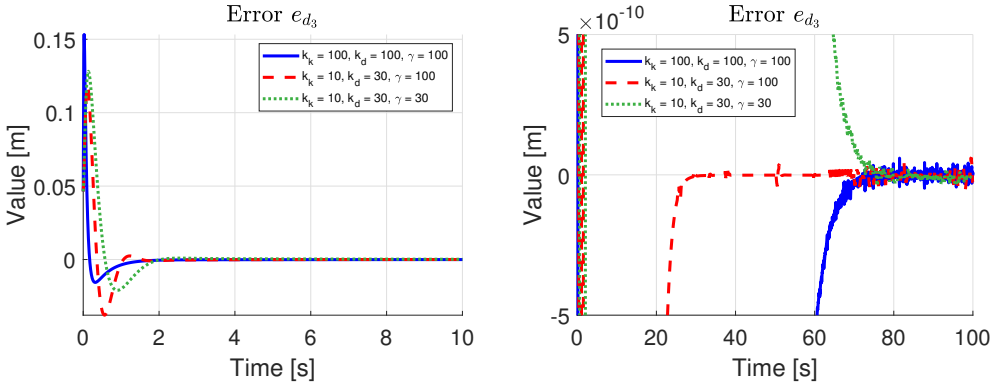
Figure 18: Path following error e_{d_1} for the helix $\alpha(s)$ – RX-60 manipulator (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 100$; Dashed line: $k_k = 10$, $k_d = 30$, $\gamma = 100$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 30$)

the proposed algorithm is asymptotically stable. Furthermore, the choice of the controller gains impacts on the pace of the convergence.



(a) Trajectory of e_{d_2} error (b) Zoom of the trajectory of e_{d_2} error

Figure 19: Path following error e_{d_2} for the helix $\boldsymbol{\alpha}(s)$ – RX-60 manipulator (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 100$; Dashed line: $k_k = 10$, $k_d = 30$, $\gamma = 100$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 30$)

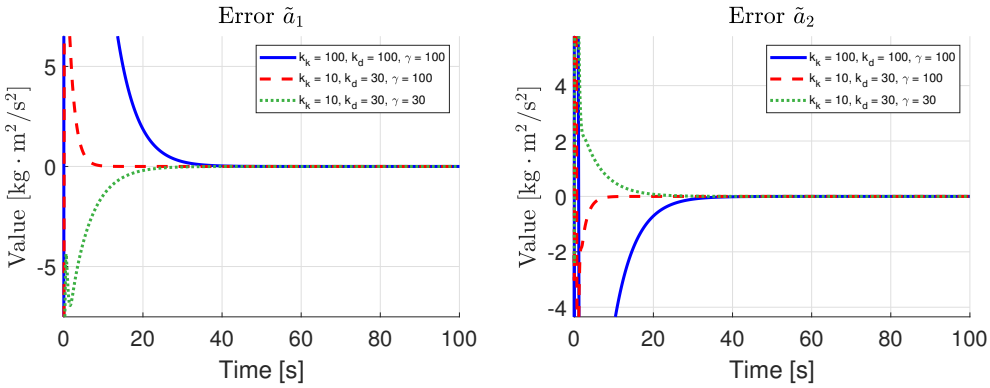


(a) Trajectory of e_{d_3} error (b) Zoom of the trajectory of e_{d_3} error

Figure 20: Path following error e_{d_3} for the helix $\boldsymbol{\alpha}(s)$ – RX-60 manipulator (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 100$; Dashed line: $k_k = 10$, $k_d = 30$, $\gamma = 100$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 30$)

In Fig. 21 errors of dynamic parameters estimation are presented. In this case all unknown parameters converge to values close to the real values. However, this feature is not crucial for the correct path following. The proposed control algorithm guarantees that the estimated values are only limited. As proved the

case with the RTR manipulator, the estimated parameters may converge to any values and the path following task is performed successfully.



(a) Trajectory of $\widetilde{a}_1$ error

(b) Trajectory of $\widetilde{a}_2$ error

Figure 21: Errors of estimation of the unknown parameters $\boldsymbol{a}$ for the helix $\boldsymbol{\alpha}(s)$ – RX-60 manipulator (Solid line: $k_k = 100$, $k_d = 100$, $\gamma = 100$; Dashed line: $k_k = 10$, $k_d = 30$, $\gamma = 100$; Dotted line: $k_k = 10$, $k_d = 30$, $\gamma = 30$)

The presented simulation case shows that the proposed control algorithm may be successfully applied to a manipulator with any kinematic structure. The adaptive algorithm on the dynamic level may be applied in order to deal with the unknown dynamic parameters.

6. Conclusions

In the paper the adaptive control algorithm for the path following task in the three-dimensional space has been presented. It allows to follow a virtual object moving along the desired path. The virtual object motion results from the considered curvilinear parametrization method. In the article the non-orthogonal Serret–Frenet parametrization has been investigated.

The Serret–Frenet parametrization allows to define the evolution of the local frame associated with a curve. It has been shown that the normalization of the parametrization does not influence effectiveness of the method application to the problem. Both normalized and non-normalized equations result in the control object description with respect to the given curve. This original approach to the control with respect to a moving frame leads to the definition of path following error dynamics. In the paper it has been shown that they take the form of non-integrable constraints of the first order. Hence, the control system has a cascade structure.

As compared to the other path following algorithms based on the Serret–Frenet parametrization, the method presented in this article introduces the non-orthogonal projection of a control object on the given curve. It is a substantial improvement as the non-orthogonal projection does not impose any additional constraints on the robot motion. Thus, there are no singularities resulting from the projection method in the control system. As a consequence, the proposed algorithm may be used globally.

Due to the cascade structure of the control system, the backstepping integrator algorithm has been applied to solve the control problem. In the paper the asymptotic convergence of the presented control algorithm for a holonomic manipulator has been presented. The kinematic control law, which is the main result of the work, guarantees that the path following errors asymptotically converge to zero. It results in correct path following. Reference velocities generated on the kinematic level are followed on the dynamic level due to the control signals generated by the dynamic controller.

The choice of the dynamic controller is arbitrary. Any dynamic controller, which guarantees velocity profiles following errors convergence to zero, allows to follow the desired path. In the article the adaptive approach has been investigated. It has been assumed that the unknown parameters of the manipulator dynamics are constant and the robot model may be expressed in linear dependency of these parameters. However, it is noteworthy that the proposed control algorithm has not guaranteed the convergence of estimated parameters to the true values as it has not been the part of control task.

The theoretical considerations have been proven with simulation study. Two different types of paths have been taken into account. One of them has been expressed in the normalized Serret–Frenet parametrization with respect to the arclength parameter, whereas for the another one it has been difficult to obtain analytical equations of the normalized parametrization. As a result, the general approach has been harnessed. Furthermore, different rates of the Serret–Frenet frame motion along the desired curve have been considered. The proposed algorithm has been applied to control the holonomic non-redundant manipulators with a fixed base. Two different kinematic structures have been taken into account. Results of the simulations have confirmed correctness of the control algorithm proposed in this paper.

To conclude, the non-orthogonal parametrization allows one to implement the global control algorithm, which guarantees that the control object reaches the desired path and moves along it. Thus, in the future the algorithm may be used for various applications, e.g. it may be implemented to control more sophisticated objects important for industrial tasks, such as mobile manipulators or redundant manipulators. Workspace of such objects is less constrained, so much wider range of curves may be taken into account. Moreover, the task of

following moving objects is crucial for space robotics, e.g. in missions of the on-orbit debris removal. The approach of control with respect to a moving frame may be investigated for such space applications.

References

[1] R.L. Bishop: There is more than one way to frame a curve. *The American Mathematical Monthly*, **82**(3), (1975), 246–251. DOI: 10.2307/2319846.

[2] C. Canudas de Wit, G. Bastin and B. Siciliano: *Theory of Robot Control*. Springer-Verlag, London, 1st edition, 1996.

[3] M. Cholewiński and A. Mazur: Path tracking by the nonholonomic mobile manipulator. In *2019 12th International Workshop on Robot Motion and Control (RoMoCo)*, (2019), 203–208. DOI: 10.1109/RoMoCo.2019.8787375.

[4] V. Cichella, I. Kaminer, E. Xargay, V. Dobrokhodov, N. Hovakimyan, A.P. Aguiar and A.M. Pascoal: A Lyapunov-based approach for time-coordinated 3D path-following of multiple quadrotors. In *Proceedings of the 51st Annual IEEE Conference on Decision and Control*, Maui, Hawaii, USA, IEEE, (2012), 1776–1781.

[5] W. Domski and A. Mazur: Path tracking with orthogonal parametrization for a satellite with partial state information. In *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018)*, **2** (2018), 252–257. DOI: 10.5220/0006835102520257.

[6] P. Encarnação and A. Pascoal: 3D path following for autonomous underwater vehicle. In *Proceedings of the 39th IEEE Conference on Decision and Control*, **3** (2000), 2977–2982. DOI: 10.1109/CDC.2000.914272.

[7] E. Erkan and S. Yüce: Serret-Frenet frame and curvatures of Bézier curves. *Mathematics*, **6**(12), (2018). DOI: 10.3390/math6120321.

[8] F. Frenet: Sur les courbes à double courbure. *Journal de Mathématiques Pures et Appliquées*, pages 437–447, 1852.

[9] M. Galicki: Path tracking by the end-effector of a redundant manipulator. *Archives of Control Sciences*, **11**(3-4), (2001), 245–261.

[10] M. Galicki: Adaptive control of kinematically redundant manipulator along a prescribed geometric path. In K. Kozłowski, editor, *Robot Motion and Control*, **335** 129–139, London, 2006. Springer London.

[11] O. Karahan and Z. Bingul: Modelling and Identification of STAUBLI RX-60 Robot. In *2008 IEEE Conference on Robotics, Automation and Mechatronics*, Chengdu, China, IEEE, (2008), 78–83. DOI: 10.1109/RAMECH.2008.4681356.

[12] M. Krstić, I. Kanellakopoulos and P.V. Kokotović: *Nonlinear and Adaptive Control Design*. John Wiley & Sons, Inc., New York, USA, 1995.

[13] Y.-L. Liao, M.-J. Zhang and L. Wan: Serret–Frenet frame based on path following control for underactuated unmanned surface vehicles with dynamic uncertainties. *Journal of Central South University*, **22**(1), (2015), 214–223. DOI: 10.1007/s11771-015-2512-z.

[14] I. Lugo-Cárdenas, S. Salazar and R. Lozano: Lyapunov based 3D path following kinematic controller for a fixed wing UAV. *IFAC-Papers OnLine, 20th IFAC World Congress*, **50**(1), (2017), 15946–15951. DOI: 10.1016/j.ifacol.2017.08.1747.

[15] A. Mazur: Hybrid adaptive control laws solving a path following problem for non-holonomic mobile manipulators. *International Journal of Control*, **77**(15), (2004), 1297–1306. DOI: 10.1080/00207170420002971 62.

[16] A. Mazur, J. Płaskonka and M. Kaczmarek: Following 3D paths by a manipulator. *Archives of Control Sciences*, **25**(1), (2015), 117–133.

[17] A. Mazur and D. Szakiel: On path following control of nonholonomic mobile manipulators. *International Journal of Applied Mathematics and Computer Science*, **19**(4), (2009), 561–574.

[18] A. Micaelli and C. Samson: *Trajectory Tracking for Unicycle-Type and Two-Steering-Wheels Mobile Robots*. Technical Report No. 2097, Sophia-Antipolis, INRIA, 1993.

[19] J. Oprea: *Differential Geometry and its Applications*. Upper Saddle River: Pearson Prentice Hall, 2nd edition, 2004.

[20] J. Płaskonka: *Path following algorithms for non-holonomic mobile manipulators*. Doctoral dissertation, Wroclaw University of Science and Technology, Poland, 2014, (in Polish).

[21] J.-A. Serret: Sur quelques formules relatives à la théorie des courbes à double courbure. *Journal de Mathématiques Pures et Appliquées*, (1851), 193–207, (in French).

[22] J. Slotine and W. Li: Adaptive manipulator control: a case study. *IEEE Transactions on Automatic Control*, **33**(11), (1988), 995–1003. DOI: 10.1109/9.14411.

[23] D. Soetanto, L. Lapierre and A. Pascoal: Adaptive, non-singular path-following control of dynamic wheeled robots. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, **2** (2003), 1765–1770. DOI: 10.1109/CDC.2003.1272868.

[24] B. Subudhi and D. Atta: Design of a path following controller for an underactuated AUV. *Archives of Control Sciences*, **19**(4), (2009), 437–450.

[25] K. Tchoń, A. Mazur, I. Dulęba, R. Hossa and R. Muszyński: *Manipulators and Mobile Robots: Models, Motion Planning, Control*. Academic Publishing House PLJ, Warsaw, 2000, (in Polish).