BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 64, No. 1, 2016
*DOI: 10.1515/bpasts-2016-0008*

# Trajectory reproduction and trajectory tracking problem for the nonholonomic systems

A. RATAJCZAK*

Department of Control Systems and Mechatronics, Wrocław University of Technology, 11/17 Janiszewski St., 50-372 Wroclaw, Poland

**Abstract.** This paper introduces a new algorithm of trajectory reproduction and trajectory tracking for nonholonomic systems. The endogenous configuration space approach is employed as a guideline in the algorithm derivation. The derivation uses a trajectory reproduction error, which is an integral of the difference between the resultant trajectory and the desired trajectory over the motion horizon. Such a definition of the error allows to solve both the trajectory reproduction as well as the trajectory tracking problem. Considerable attention in the paper has been paid to the implementation aspects of the algorithm. The nonparametric approach is used together with a higher order of the integration method. The algorithm efficiency is illustrated with computer simulations accomplished for two nonholonomic systems: the dynamics of the double pendulum with a passive joint, and the kinematics of the unicycle.

**Key words:** trajectory reproduction, trajectory tracking, nonholonomic system.

## 1. Introduction

The motion planning problem [1] consist in finding an inverse of the end-point map. Having this inverse one can find a control function which drives the system from defined initial point to the desired point. A slightly different kind of problem is the trajectory reproduction or trajectory tracking problem. In this case, in order to solve the problem the instantaneous map has to be inverted. Then, using this inverse we are able to compute the control function, which drives the system in such a way that the resultant trajectory is as close as possible to the desired trajectory. This kind of problem relies on finding a continuous inverse of function-to-function map. If the desired trajectory is an admissible trajectory of the system, and for the initial time the system stands on the desired trajectory then the problem is called trajectory reproduction problem. When one or both above conditions are not meet then the problem is called trajectory tracking problem.

The reproduction problem is the one of the fundamental problems in the system theory. It allows to consider such problems as a system reproducibility or finding the continuous map inverse. The problem of the linear system reproducibility is introduced in [2]. The review of the continuous map inverse in nonlinear systems can be found in [3].

The trajectory reproduction or the trajectory tracking problem for the nonholonomic systems is widespread in the literature. For example, in [4] the authors present the stabilization around the desired trajectory. This approach is based on a closed loop algorithm, however the initial control which realizes the nominal trajectory (without disturbances) is needed. The trajectory tracking with obstacles avoidance is considered in [5], where in order to solve the trajectory tracking problem the Vector Field Orientation method together with the Artificial Potential Function are employed. The trajectory reproduction with the continuation method is presented in [6]. There, the wanted control function is obtained as a solution of the second order partial differential equation. The trajectory tracking problem can be also solved using the Lie algebra together with the transverse function control approach presented in [7, 8]. The experimental verification of the solution of the trajectory tracking problem could be also find in the literature [9, 10].

This paper introduces a new approach to the trajectory reproduction and the trajectory tracking problem. In order to derive the algorithm the endogenous configuration space approach [11] is employed as a guideline. As a result of the new algorithm we obtain a control function which provides a system motion which minimizes the distance between the system resultant trajectory and the desired trajectory. The endogenous configuration space approach was firstly applied to a continuous inverse problem in [12]. In this work, the continuous inverse problem is solved in approximated way with an arbitrary error profile. The new approach presented in this paper slightly corresponds to the theoretical framework in [12]. The main novel of this work is the formulation of the instantaneous kinematics of a nonholonomic system and a Jacobian inverse algorithm based on the endogenous configuration space approach which allows us to solve the continuous inverse problem for a nonholonomic systems.

One of the advantages of the presented methodology is that the introduced algorithm does not depend on the properties of the systems. It can be applied to any nonholonomic system which could be represented by a control affine system. In the paper the approach will be enroll to two different nonholonomic systems, i.e. the underactuated manipulator and the unicycle. The former with the second order nonholonomic constraints and the latter with the first order nonholonomic constraints.

*e-mail: adam.ratajczak@pwr.edu.pl

The paper is composed in the following way. Section 2 introduces the trajectory reproduction and the trajectory tracking problem. The algorithm derivation with all necessary theoretical preliminaries contains Sec. 3. The implementation aspects together with the simulation results are collected in Sec. 4. Section 5 concludes the paper.

## 2. Trajectory reproduction/tracking problem

Let us introduce the control affine system with the output function

$$\begin{cases} \dot{q} = f(q) + G(q)u, \\ y = k(q), \end{cases} \tag{1}$$

where $q \in \mathbb{R}^n$ is the state space vector, $u \in \mathbb{R}^m$ stands for the vector of control functions, and the task space vector is denoted by $y \in \mathbb{R}^r$. Moreover, in the control affine system (1), $G(q) \in \mathbb{R}^{n \times m}$ is the control matrix and $f(q) \in \mathbb{R}^n$ is the drift term. For such system let us define the trajectory reproduction/tracking problem in the following way. Find a control function $u(t)$ which drives the system (1), over a prescribed time interval $[0, T]$, such that the system output trajectory $y(t)$ is as close as possible to the demanded trajectory $y_d(t)$.

**Remark 1.** When the initial condition $y(0)$ of the system (1) output coincides with the demanded trajectory at $t = 0$, i.e. $y(0) = y_d(0)$ and the desired trajectory is an admissible trajectory then the problem becomes a trajectory reproduction problem. When the initial points of the output and demanded trajectory are different or when the demanded trajectory is non-admissible then the problem will be called the trajectory tracking problem.

## 3. Algorithm

The algorithm derivation bases on the endogenous configuration space approach [11], which was originally introduced to solve the motion planning problem for mobile manipulators. However, the systematically exploration of that idea yields with the adaptation of the endogenous configuration space approach to more complex systems [13]. The endogenous configuration is a space $L^2_m[0,T]$ of Lebesgue square integrable functions defined on the interval $[0, T]$. The space $L^2_m[0,T]$ is a Hilbert space with inner product

$$\langle u_1(\cdot), u_2(\cdot) \rangle = \int_0^T u_1^{\mathrm{T}}(t) u_2(t) \, \mathrm{d}t$$

and the norm $\|u(\cdot)\|^2 = \int_0^T u^{\mathrm{T}}(t)u(t) \ \mathrm{d}t$. Every admissible control function $u(\cdot) \in \mathcal{U} \subset L^2_m[0,T]$ corresponds to the state trajectory $q(t) = \varphi_{q_0,t}(u(\cdot))$ and output trajectory $y(t) = k(\varphi_{q_0,t}(u(\cdot)))$, where $\varphi_{q_0,t}(u(\cdot))$ denotes the flow of the system (1) at the moment $t$, starting from $q_0$ and driven by $u(\cdot)$. Let us assume that those trajectories are well defined for every $t \in [0, T]$.

### 3.1. Preliminaries.

Let us introduce the instantaneous map which transforms the control space to task space $K_{q_0,t}\colon \mathcal{U} \to \mathbb{R}^r$. For every control function $u(t) \in \mathcal{U}$ the instantaneous map provides the output value $y(t)$ and is defined as follows

$$K_{q_0,t}(u(\cdot)) = y(t) = k(\varphi_{q_0,t}(u(\cdot))). \tag{2}$$

Formally, the map (2) resembles the instantaneous kinematics. Following that, let us define the system Jacobian as a derivative of the instantaneous map

$$\begin{aligned} J_{q_0,t}(u(\cdot))v(\cdot) &= \mathcal{D}K_{q_0,t}(u(\cdot))v(\cdot) \\ &= \left.\frac{\mathrm{d}}{\mathrm{d}\delta}\right|_{\delta=0} K_{q_0,T}(u(\cdot) + \delta v(\cdot)) \\ &= C(t) \int_0^t \Phi(t,s)B(s)v(s) \ \mathrm{d}s. \end{aligned} \tag{3}$$

To compute the system Jacobian let us introduce the linear approximation to system (1)

$$\begin{cases} \dot{\xi}(t) = A(t)\xi(t) + B(t)v(t), \\ \eta(t) = C(t)\xi(t), \end{cases} \tag{4}$$

along a control-trajectory $(u(t), q(t))$ pair where the matrices can be computed from

$$A(t) = \frac{\partial(f(q(t)) + G(q(t))u(t))}{\partial q},$$
$$B(t) = \frac{\partial(f(q(t)) + G(q(t))u(t))}{\partial u} = G(q(t)),$$
$$C(t) = \frac{\partial k(q(t))}{\partial q}$$

and

$$\xi(t) = \mathcal{D}\varphi_{q_0,t}(u(\cdot)).$$

The matrix $\Phi(t, s)$ from (3) is the fundamental matrix of the linear system (4) and fulfills the differential equation [14]

$$\frac{\partial}{\partial t}\Phi(t,s) = A(t)\Phi(t,s), \tag{5}$$

with initial condition $\Phi(s, s) = I_n$.

Enrolling the above formulas we can rewrite the definition of the trajectory reproduction/tracking problem as: find a control function $u^*(\cdot) \in \mathcal{U}$, which drives the system in such way that the difference between the output trajectory and the desired trajectory is as small as possible

$$\min_{u(\cdot)} \|y(\cdot) - y_d(\cdot)\|^2 = \min_{u(\cdot)} \|k(\varphi_{q_0,t}(u(\cdot))) - y_d(t)\|^2.$$

To start the derivation of the algorithm we shall formulate the task map $\mathbb{K}^1_{q_0,T}\colon \mathcal{U} \to \mathbb{R}$ of the trajectory reproduction/tracking problem which takes the following form

$$\begin{aligned} &\mathbb{K}^1_{q_0,T}(u(\cdot)) = \\ &\frac{1}{2}\int_0^T (y(t) - y_d(t))^{\mathrm{T}} W(t)(y(t) - y_d(t)) \, \mathrm{d}t, \end{aligned} \tag{6}$$

where $W(t) = \mathrm{diag}\{w_1(t), w_2(t), \ldots, w_r(t)\}, \ w_i(t) > 0$ is a diagonal weighting matrix. Alternatively, the task map (6) could be defined as a vector instead of a scalar, i.e. $\mathbb{K}^2_{q_0,T} \colon \mathcal{U} \to \mathbb{R}^r$, then the map takes the form

$$\mathbb{K}^2_{q_0,T}(u(\cdot)) = \frac{1}{2}\int\limits_0^T W(t) \begin{pmatrix} (y_1(t)-y_{1d}(t))^2 \\ (y_2(t)-y_{2d}(t))^2 \\ \vdots \\ (y_r(t)-y_{rd}(t))^2 \end{pmatrix} \mathrm{d}t,$$

where again

$$W(t) = \mathrm{diag}\{w_1(t), w_2(t), \dots, w_r(t)\},$$

$$w_i(t) > 0$$

is a diagonal weighting matrix. The both definitions of task maps could be rewritten in more general form as

$$\mathbb{K}_{q_0,T}(u(\cdot)) = \int\limits_0^T F(y(t))\ \mathrm{d}t = \int\limits_0^T F(k(q(t)))\ \mathrm{d}t, \qquad (7)$$

where the function $F(y(t)) \geq 0$ and it is differentiable wherever defined.

With the task map (7) we associate the Jacobian which may be obtained by differentiation of the task map

$$\begin{gathered} \mathbb{J}_{q_0,T}(u(\cdot))v(\cdot) = \mathcal{D}\mathbb{K}_{q_0,T}(u(\cdot)) \\ = \left.\frac{\mathrm{d}}{\mathrm{d}\delta}\right|_{\delta=0} \mathbb{K}_{q_0,T}(u(\cdot)+\delta v(\cdot)) = \\ \left.\frac{\mathrm{d}}{\mathrm{d}\delta}\right|_{\delta=0} \int\limits_0^T F(k(\varphi_{q_0,t}(u(\cdot)+\delta v(\cdot))))\ \mathrm{d}t \\ = \int\limits_0^T \frac{\partial F(y(t))}{\partial y} \mathcal{D}k(\varphi_{q_0,t}(u(\cdot))v(\cdot))\ \mathrm{d}t. \end{gathered} \qquad (8)$$

Substituting (3) into (8) we arrive with the general formula of the Jacobian for trajectory reproduction/tracking problem

$$\begin{gathered} \mathbb{J}_{q_0,T}(u(\cdot))v(\cdot) \\ = \int\limits_0^T \frac{\partial F(y(t))}{\partial y} J_{q_0,t}(u(\cdot))v(\cdot)\ \mathrm{d}t. \end{gathered}$$

Using the above derivation we formulate two Jacobians for task map $\mathbb{K}^1_{q_0,T}(u(\cdot))$ and $\mathbb{K}^2_{q_0,T}(u(\cdot))$ respectively as

$$\begin{gathered} \mathbb{J}^1_{q_0,T}(u(\cdot))v(\cdot) \\ = \int\limits_0^T (y(t)-y_d(t))^{\mathrm{T}} W(t) J_{q_0,t}(u(\cdot))v(\cdot)\ \mathrm{d}t \end{gathered} \qquad (9)$$

and

$$\begin{gathered} \mathbb{J}^2_{q_0,T}(u(\cdot))v(\cdot) \\ = \int\limits_0^T \mathrm{diag}\{(y_1(t)-y_{1d}(t)), \dots, \\ (y_r(t)-y_{rd}(t))\} W(t) J_{q_0,t}(u(\cdot))v(\cdot)\ \mathrm{d}t. \end{gathered} \qquad (10)$$

**3.2. Algorithm derivation.** Using the task map (7) we can define the error function of the trajectory reproduction/tracking problem as $e_T = \mathbb{K}_{q_0,T}(u(\cdot))$. In view of that error formula the trajectory reproduction/tracking algorithm finds a control function which drives the system (1) in such way that the error value is minimised. The following problem can be solved by means of a Jacobian algorithm whose derivation uses the homotopy method [15]. This being so, in the control space $\mathcal{U}$ we choose a smooth curve $u_\theta(\cdot)$ parametrized with a variable $\theta \in \mathbb{R}$, which passes a certain initial configuration $u_0(\cdot)$. Along this curve we compute the trajectory reproduction/tracking error $e_T(\theta)$ and we want it to decrease exponentially

$$\frac{\mathrm{d}e_T(\theta)}{\mathrm{d}\theta} = -\gamma e_T(\theta), \qquad (11)$$

with a decay rate $\gamma$. Substituting $e_T(\theta) = \mathbb{K}_{q_0,T}(u_\theta(\cdot))$ into (11) we arrive with the Ważewski-Davidenko equation

$$\begin{gathered} \frac{\mathrm{d}}{\mathrm{d}\theta}\mathbb{K}_{q_0,T}(u_\theta(\cdot)) \\ = \mathbb{J}_{q_0,T}(u_\theta(\cdot))\frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -\gamma e_T(\theta). \end{gathered} \qquad (12)$$

Employing a right Jacobian inverse $\mathbb{J}^{\#}_{q_0,T}(u_\theta(\cdot))$, such that $\mathbb{J}_{q_0,T}(u_\theta(\cdot))\mathbb{J}^{\#}_{q_0,T}(u_\theta(\cdot)) = I$, the Eq. (12) takes the form of the dynamic system

$$\frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -\gamma \mathbb{J}^{\#}_{q_0,T}(u_\theta(\cdot))e_T(\theta), \qquad (13)$$

which constitutes the trajectory reproduction/tracking algorithm. The solution of the trajectory tracking problem is a limit $u^*(t) = \lim_{\theta\to\infty} u_\theta(\cdot)$ of the resultant trajectory of (13).

As a right Jacobian inverse appearing in (13) we choose the Moore-Penrose inverse derived from minimizing the square norm of the control function $\min_{v(\cdot)} \|v(\cdot)\|^2$ with respecting the Jacobian equation

$$\mathbb{J}_{q_0,T}(u(\cdot))v(\cdot) = \eta, \qquad (14)$$

where if we take $\mathbb{J}^1_{q_0,T}(u(\cdot))v(\cdot)$ then the $\eta \in \mathbb{R}$ and if we take $\mathbb{J}^2_{q_0,T}(u(\cdot))v(\cdot)$ then the $\eta \in \mathbb{R}^r$. The corresponding Lagrange function takes the form

$$L(v(\cdot),\lambda) = \int\limits_0^T \left( v^{\mathrm{T}}(t)v(t) + \lambda\frac{\partial F(y(t))}{\partial y} J_{q_0,t}(u(\cdot))v(\cdot) \right)\mathrm{d}t$$

$$= \int\limits_0^T \left( v^{\mathrm{T}}(t)v(t) + \lambda\frac{\partial F(y(t))}{\partial y} C(t)\int\limits_0^t \Phi(t,s)B(s)v(s)\ \mathrm{d}s \right)\mathrm{d}t,$$

where $\lambda \in \mathbb{R}$ is a Lagrange multiplier. The differentiation of the Lagrange function with respect to $v(\cdot)$ and equating the derivative to 0

$$\mathcal{D}L(v(\cdot),\lambda)\mu(\cdot) = \left.\frac{\mathrm{d}}{\mathrm{d}\delta}\right|_{\delta=0} L(v(\cdot)+\delta\mu(\cdot),\lambda) = 0,$$

together with the identity

$$\int\limits_0^T\int\limits_0^t f(t,s)\mathrm{d}s\mathrm{d}t = \int\limits_0^T\int\limits_s^T f(t,s)\ \mathrm{d}t\ \mathrm{d}s,$$

yield

$$v(t) = -\lambda B^{\mathrm{T}}(t) \int_t^T \Phi(s,t) \left( \frac{\partial F(y(t))}{\partial y} \right)^{\mathrm{T}} \mathrm{d}s = -\lambda \alpha(t). \quad (15)$$

Now, we eliminate the Lagrange multiplier $\lambda$ by inserting (15) into (14) and we arrive with equality

$$-\lambda \|\alpha(\cdot)\|^2 = \eta,$$

which allows us to compute the $\lambda$. Finally we obtain

$$v(t) = \frac{\alpha(t)}{\|\alpha(\cdot)\|^2} \eta$$

which defines the right Jacobian pseudoinverse [16] for trajectory reproduction/tracking problem

$$\left( \mathbb{J}^{\#}_{q_0,T}(u_\theta(\cdot))\eta \right)(t) = \frac{\alpha(t)}{\|\alpha(\cdot)\|^2} \eta. \quad (16)$$

The form of the particular Jacobian pseudoinverse, corresponding with one of the Jacobians (9) or (10), could be obtained from the formula (16)

$$\begin{aligned} &\mathbb{J}^{1\#}_{q_0,T}(u(\cdot)) \colon \mathbb{R} \to \mathcal{U}, \\ &\left( \mathbb{J}^{1\#}_{q_0,T}(u(\cdot))\eta \right)(t) = \frac{\alpha_1(t)}{\|\alpha_1(\cdot)\|^2} \eta, \end{aligned} \quad (17)$$

where

$$\alpha_1(t) = B^{\mathrm{T}}(t) \int_t^T \Phi^{\mathrm{T}}(s,t) C^{\mathrm{T}}(s) W(s)(y(s) - y_d(s))\mathrm{d}s,$$

and

$$\begin{aligned} &\mathbb{J}^{2\#}_{q_0,T}(u(\cdot)) \colon \mathbb{R}^r \to \mathcal{U}, \\ &\left( \mathbb{J}^{2\#}_{q_0,T}(u(\cdot))\eta \right)(t) = \frac{\alpha_2(t)}{\|\alpha_2(\cdot)\|^2} \eta, \end{aligned} \quad (18)$$

where

$$\alpha_2(t) = B^{\mathrm{T}}(t) \int_t^T \Phi^{\mathrm{T}}(t,s) C^{\mathrm{T}}(s) W(s)$$
$$\mathrm{diag}\{(y_1(s) - y_{1d}(s)), \ldots, (y_r(s) - y_{rd}(s))\}\mathrm{d}s.$$

One of the presented Jacobian pseudoinverses (17) or (18) could be used in the algorithm Eq. (13) to solve the trajectory reproduction/tracking problem. The computed control function $u^*(t)$ generates the system (1) output trajectory which is as close as possible to the demanded trajectory.

## 4. Numerical computation

In order to obtain the solution of the Jacobian trajectory reproduction/tracking algorithm it is necessary to solve the functional differential Eq. (13) in the control space $\mathcal{U}$. The solution could be achieved using several techniques. On one side, the computation can be provided with different integrating methods, e.g. Euler scheme or higher order method along with step size optimization. On the other side, the control function can be represented with the orthogonal series or can be sampled with sufficient number of points. The first representation will be called parametric, the second one – non-parametric [17]. Combining the presented techniques the algorithm computation falls into four classes: 1. Euler and parametric, 2. Euler and non-parametric, 3. higher order and parametric and finally 4. higher order and non-parametric. The recent research shows that the first class is usually easy to implement however it contains more numerical errors. The implementation of the last class make a challenge, but the results coincide with the theoretical derivation.

The trajectory reproduction problem solved with the techniques of the class 1. is presented in [18]. In this paper the trajectory reproduction/tracking problem will be solved enrolling the 4-th class technique.

**4.1. Implementation aspects.** As it was already mentioned the trajectory reproduction/tracking algorithm will be solved directly using the non-parametric version and the *MATLAB* built-in higher order integration method. For each value of independent variable $\theta$ it is necessary to solve the control affine system Eq. (1) along with the fundamental matrix Eq. (5). The solutions of those equations can be substituted into the Jacobian inverse Eq. (16) or (18). Finally, using the Jacobian inverse we can solve the algorithm functional differential Eq. (13). Collecting all necessary equations we arrive with the following set of differential-algebraic equations (DAE)

$$\left\{ \begin{aligned} &\frac{\mathrm{d}q_\theta(t)}{\mathrm{d}t} = f(q_\theta(t)) + G(q_\theta(t))u_\theta(t), \\ &\frac{\mathrm{d}\Phi_\theta(T,t)}{\mathrm{d}t} = -\Phi_\theta(T,t)A_\theta(t), \\ &\frac{\mathrm{d}u_\theta(t)}{\mathrm{d}\theta} = -\gamma \frac{\alpha_1(t)}{\|\alpha_1(\cdot)\|^2} e_T(\theta), \\ &\alpha_1(t) = B^{\mathrm{T}}(t) \int_t^T \Phi^{\mathrm{T}}(s,t) C^{\mathrm{T}}(s) \\ &\times W(s)(y_\theta(s) - y_d(s))\mathrm{d}s, \\ &A_\theta(t) = \frac{\partial (f(q_\theta(t)) + G(q_\theta(t))u_\theta(t))}{\partial q}, \\ &B_\theta(t) = G(q_\theta(t)), \\ &C_\theta(t) = \tfrac{\partial k(q_\theta(t))}{\partial q}, \\ &e_T(\theta) = \frac{1}{2} \int_0^T (y_\theta(t) - y_d(t))^{\mathrm{T}} \\ &\times W(t)(y_\theta(t) - y_d(t))\ \mathrm{d}t, \end{aligned} \right. \quad (19)$$

with boundary conditions $q_\theta(0) = q_0$, $\Phi_\theta(T,T) = I_n$, and given initial control function $u_0(t)$. The DAE system (19) could be simply rewritten for the second task map $\mathbb{K}^2_{q_0,T}(u(\cdot))$.

**4.2. Simulations.** In this section we present the simulation results for the two models of nonholonomic robotic systems. The first model is a planar double pendulum with passive second joint, usually denoted by $R\overline{R}$ where the bar over the joint symbol means no actuation. The second model is a kinematics of a unicycle. The presented computations was done in *MATLAB* environment.

$R\overline{R}$ **manipulator.** The schematic overview of the manipulator is depicted in Fig. 1. The meaning of the symbols presented in the figure are as follows: $l_1$, $l_2$ are the length of the links and $m_1$, $m_2$ are the links masses. Under the assumption that there is no gravity force the dynamics of the $R\overline{R}$ manipulator takes the form [19]

$$M(x)\ddot{x} + N(x,\dot{x})\dot{x} = \begin{pmatrix} \tau \\ 0 \end{pmatrix}, \tag{20}$$

where $x \in \mathbb{R}^2$ is the joint space vector and $\tau$ is the control momentum in the actuated joint. The inertia matrix is denoted as $M(x)$, and the $N(x,\dot{x})$ is a matrix of Coriolis and centrifugal terms, and their definition are as follows

$$M(x) = \begin{bmatrix} a + \frac{m_2(l_1^2+l_2^2)}{3} & \frac{l_1 l_2 m_2 \cos x_2}{2} + \frac{l_2^2 m_2}{3} \\ \frac{l_1 l_2 m_2 \cos x_2}{2} + \frac{l_2^2 m_2}{3} & \frac{l_2^2 m_2}{3} \end{bmatrix},$$

$$N(x,\dot{x}) = \begin{bmatrix} -\frac{m_2 l_1 l_2 x_4 \sin x_2}{2} & -\frac{m_2 l_1 l_2 (x_3+x_4) \sin x_2}{2} \\ \frac{m_2 l_1 l_2 x_3 \sin x_2}{2} & 0 \end{bmatrix},$$

where

$$a = m_2 l_1 l_2 \cos x_2 + m_2 l_1^2.$$

The second equation in (20) is a set of the second order nonholonomic constraints.

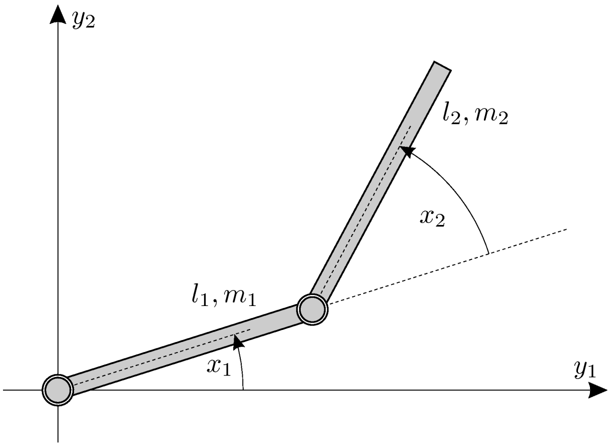

Fig. 1. $R\overline{R}$ manipulator

In order to simplify the dynamics of the manipulator we apply the partially feedback linearization [20]

$$\begin{gathered}\tau = (M_{11}(x) - M_{12}(x)M_{22}^{-1}(x)M_{12}^{\mathrm{T}}(x))u \\ -M_{12}(x)M_{22}^{-1}(x)\left(\left[N_{21}(x,\dot{x}), N_{22}(x,\dot{x})\right]\dot{x}\right) \\ + \left[N_{11}(x,\dot{x}), N_{12}(x,\dot{x})\right]\dot{x}\end{gathered}$$

and after introducing the new state space $q = (x,\dot{x}) \in \mathbb{R}^4$ we arrive with partially linearized control affine system with output function

$$\begin{cases} \dot{q} = f(q) + G(q)u \\ = \begin{pmatrix} q_3 \\ q_4 \\ 0 \\ -\begin{bmatrix} N_{21} & N_{22} \end{bmatrix} M_{22}^{-1} \begin{pmatrix} q_3 \\ q_4 \end{pmatrix} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ -M_{21}M_{22}^{-1} \end{pmatrix} u, \\ y = k(q) = \begin{pmatrix} l_1 \cos q_1 + l_2 \cos(q_1+q_2) \\ l_1 \sin q_1 + l_2 \sin(q_1+q_2) \end{pmatrix}. \end{cases} \tag{21}$$

In Eq. (21) the $u$ has the meaning of the angular acceleration of the actuated joint expressed in $[\mathrm{rad/s^2}]$, and the output function represents the coordinates of the end-effector in task space. For the sake of simplification we choose the model parameters equal $l_1 = l_2 = 0.5$, $m_1 = 1$, $m_2 = 0.5$, where the length could be expressed in meters [m] and the masses in kilograms [kg].

**Unicycle.** The second nonholonomic robotic system used as a testbed is an unicycle. The diagram of this model is presented in Fig. 2. The kinematics of the unicycle is derived under the assumption that the wheel cannot slip laterally and longitudinally and takes the form

$$\begin{cases} \dot{q} = \begin{bmatrix} \cos q_3 & 0 \\ \sin q_3 & 0 \\ 0 & 1 \end{bmatrix} u, \\ y = k(q) = q, \end{cases}$$

where $q \in \mathbb{R}^3$ is the state space vector which denotes the position and orientation of the unicycle and $u \in \mathbb{R}^2$ is the control vector whose first element is the longitudinal velocity expressed in $[\mathrm{m/s}]$, and the second element represents the angular velocity in $[\mathrm{rad/s}]$. We assume that the output function is an identity so the output space is equal to the state space.

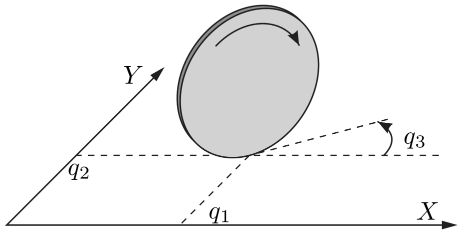

Fig. 2. Unicycle

**Simulation results.** For each of two previously introduced nonholonomic robotic systems, the following simulations scenario are provided. **Simulation 1**: output trajectory tracking problem for an $R\overline{R}$ manipulator with non-admissible desired trajectory solved using the Jacobian of the form (8). **Simulation 2**: state trajectory reproduction problem for an unicycle solved using the Jacobian of the form (10). **Simulation 3**: state trajectory tracking problem for an unicycle solved using the Jacobian of the form (8). It is assumed that the units in all plots are expressed in the SI system.

**Simulation 1**. For a desired trajectory we choose a straight line $y_d(t) = ((l_1 + l_2)(T - t)/T, 0)$ coincides with $y_1$ axis (see Fig. 4, dashed line). The simulation parameters is set as follows: time horizon $t \in [0, 5]$, the decay rate $\gamma = 3$, the initial state of the $R\overline{R}$ manipulator $q_0 = (0, 0, 0, 0)$ and the initial control $u_0(t) = 0.1 \sin(2\pi t/T)$. In the Fig. 3 the resultant control is depicted. The desired trajectory is non-admissible so it is impossible to obtain the error equal zero. This can be observed in the Fig. 4 as well as in the Fig. 6 where the convergence of the algorithm is shown. One can see that the error decrease exponentially, and then saturates on certain value, so no further improvement of the resultant control could be made. The difference between the trajectory of the output and the desired trajectory is depicted in Fig. 5.

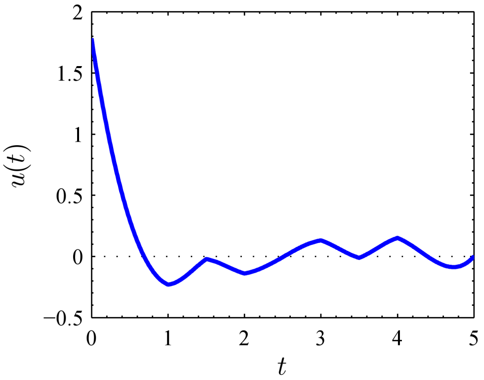

Fig. 3. Simulation 1. Control functions

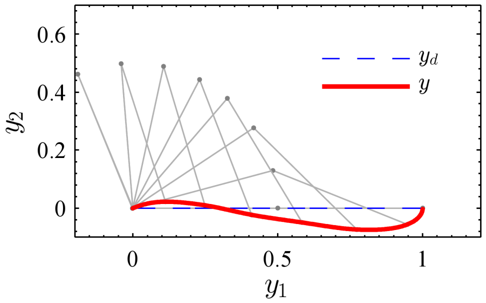

Fig. 4. Simulation 1. Motion path in task space

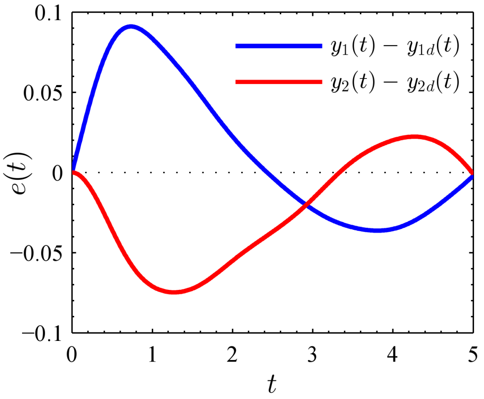

Fig. 5. Simulation 1. Output function errors

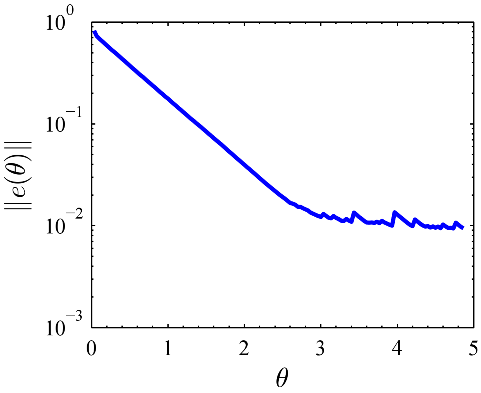

Fig. 6. Simulation 1. Algorithm convergence

**Simulation 2.** The desired trajectory is a circle trajectory with the center in the axis origin and the radius equal to 1, so $q_d = (\sin(2\pi t/T), -\cos(2\pi t/T), 2\pi t/T)$. This simulation results are obtained for the following parameter values: $T = 5$, $\gamma = 4$, $u_0(t) = (t/4, t/4)$ and $q_0 = [0, -1, 0]$ which means that at the beginning $t = 0$, the unicycle stands on the desired trajectory $q(0) = q_d(0)$. Figures 7–9 present the solution of the state trajectory reproduction for the unicycle. The resultant control functions are depicted in Fig. 7. Figure 8 shows the difference between the unicycle trajectory and the desired trajectory. The values of the errors could be even more decreased by increasing the accuracy of computations. As it can be seen in Fig. 9 the error norm decrease exponentially which coincides with (11).

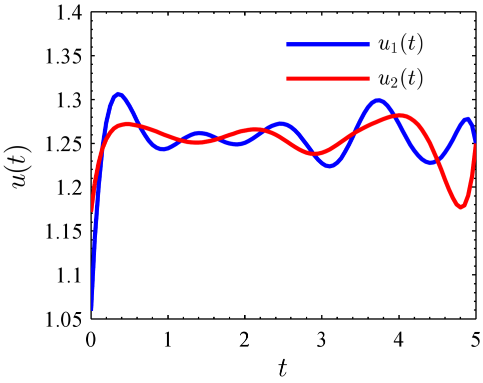

Fig. 7. Simulation 2. Control functions

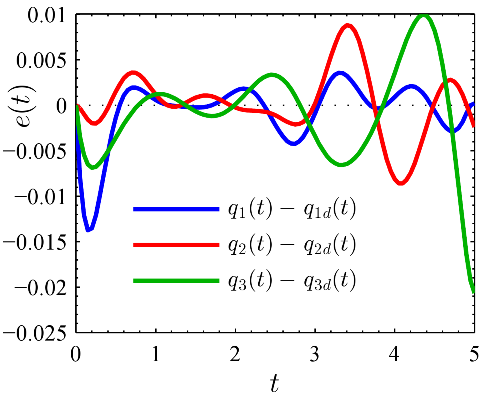

Fig. 8. Simulation 2. State errors

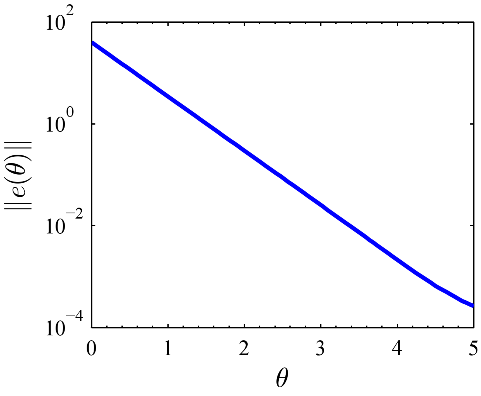

Fig. 9. Simulation 2. Algorithm convergence

**Simulation 3.** The last simulation result shows a solution of the trajectory tracking problem. The desired trajectory is equal to the previous one $q_d = (\sin(2\pi t/T), -\cos(2\pi t/T), 2\pi t/T)$. However, this time, the initial position of the unicycle is moved away from the desired trajectory $q_0 = (0, 0, 0)$. The rest of the simulation parameters are set as: $T = 10$, $\gamma = 4$, $u_0(t) = (\pi/5, \pi/5)$. The solution of the problem is presented in Figs. 10–15. The Fig. 10 shows the resultant control functions of the unicycle. The motion path of the model is depicted in Fig. 11. One can see that the distance between trajectories is decreasing during the motion. The same observations could be deduced from Fig. 12 where the differences between the real and desired trajectory are shown. The trajectories of the state vector along with the desired state trajectories present Figs. 13 and 14. As in previous simulations the algorithm converges exponentially (see Fig. 15).

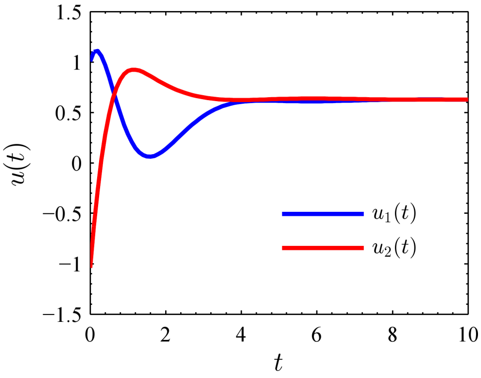

Fig. 10. Simulation 3. Control functions

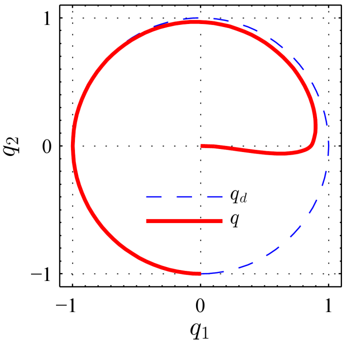

Fig. 11. Simulation 3. Motion path in $XY$ plane

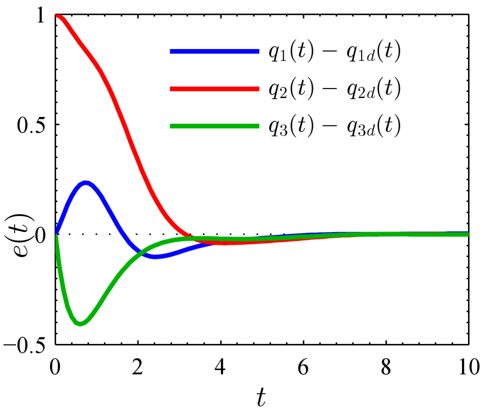

Fig. 12. Simulation 3. State errors

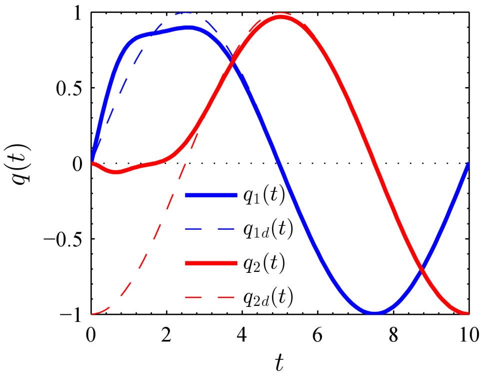

Fig. 13. Simulation 3. State trajectories (position)

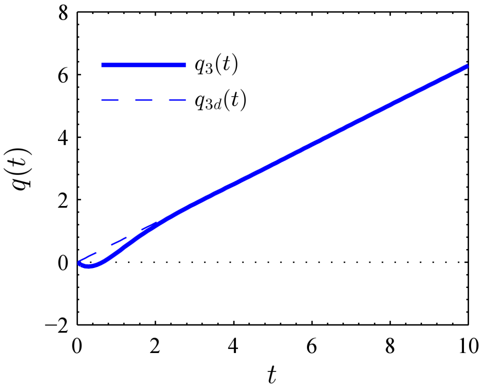

Fig. 14. Simulation 3. State trajectories (orientation)

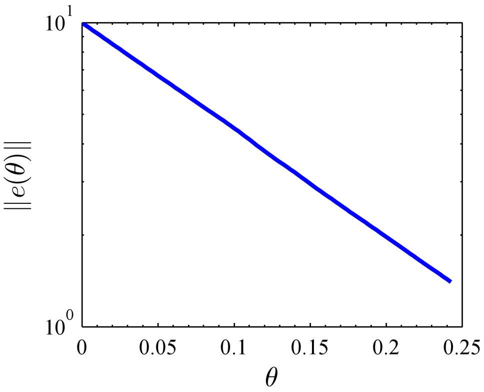

Fig. 15. Simulation 3. Algorithm convergence

## 5. Conclusions

The proposed algorithm satisfactorily solves all presented problems. The algorithm efficiency is illustrated with the simulation results of the trajectory reproduction as well as the trajectory tracking problem for two nonholonomic robotic systems.

The presented approach is a preliminary result and needs some further improvements. Currently, the proposed algorithm convergence strongly depends on the choice of the initial control $u_0(t)$. Further research will be performed to overtake this disadvantage.

The presented improvements of the implementation results in more accurate computations, contrary to the numerical method presented in [18]. Moreover, the accuracy of computation could be manipulated using a standard *MATLAB* operation. This fact also helps to obtain even smaller trajectory reproduction or tracking error. The decrease of the error value could be achieved by increasing the computation accuracy, however, it reflects on the computation time.

Up to now, the main disadvantage of this approach is the computation time. The computation of the algorithm step (i.e. the solution of system (19) for one value of $\theta$) takes about 4s (PC with a 3.2GHz processor). The number of computed steps strongly depends on the chosen accuracy. The simulations presented in the paper take around 500 steps. Those numbers disqualify the present form of the algorithm to be able to operate in real time. One of the future work direction will be focused on the computation time reduction. On the other hand, the presented algorithm is an open-loop algorithm, so the computation could be done before the real motion. One could find the control function, which should drive the system close to the desired trajectory, which for example, could be passed as an input argument to the control algorithm presented in [4].

**Acknowledgements.** This research was supported by the National Science Centre, Poland, under the grant decision No DEC-2013/09/B/ST7/02368.

### References

[1] S.M. LaValle, *Planning Algorithms*, Cambridge University, Cambridge, 2006.

[2] R.W. Brockett and M.D. Mesarewić, "The reproducibility of multivariable systems", *J. Math. Anal. Appl.* 11, 548–563 (1965).

[3] W. Respondek, "Right and left invertibility of nonlinear control systems", in ed. H. J. Sussmann, *Non-linear Controllability and Optimal Control*, Marcel Dekker, New York, 1990.

[4] G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J.P. Laumond, "Stabilization of trajectories for systems with nonholonomic constraints", *IEEE Trans. Autom. Control* 39 (1), 216–222 (1994).

[5] W. Kowalczyk, M. Michałek, and K. Kozłowski, "Trajectory tracking control with obstacle avoidance capability for unicycle-like mobile robot", *Bull. Pol. Ac.: Tech.* 60 (3), 537–546 (2012).

[6] J. Karpińska and K. Tchoń, "Continuation method approach to trajectory planning in robotic systems", *16th Int. Conf. Methods and Models in Automation and Robotics, MMAR* 2011, 51–56 (2011).

[7] P. Morin and C. Samson, "Trajectory tracking for nonholonomic vehicles", ed. K. Kozłowski, *Robot Motion and Control*, Lecture Notes in Control and Information Sciences, vol. 335, pp. 3–23, Springer, London, 2006.

[8] P. Morin and C. Samson, "Trajectory tracking for nonholonomic systems. Theoretical background and applications", *Research Report* RR-6464, CD-ROM (2008).

[9] D. Pazderski and K. Kozłowski, "Motion control of a skid-steering robot using transverse function approach – experimental evaluation", in *10th Int. Workshop on Robot Motion and Control, RoMoCo* 2015, 72–77 (2015).

[10] D. Chaos, R. Muñoz-Mansilla, D. Moreno-Salinas, and J. Aranda, "Nonlinear control for trajectory tracking of a nonholonomic rc-hovercraft with discrete inputs", *Mathematical Problems in Engineering* 2013, ID 589267 (2013).

[11] K. Tchoń and J. Jakubiak, "Endogenous configuration space approach to mobile manipulators: a derivation and performance assessment of Jacobian inverse kinematics algorithms", *Int. J. Control* 76 (14), 1387–1419 (2003).

[12] K. Tchoń and J. Jakubiak, "Simple algorithm for solving the continous inverse kinematics problem for stationary and mobile robots", ed. K. Tchoń, *Advances in Robotics*, pp. 275–282, WKŁ, Warsaw, 2005, (in Polish).

[13] D. Paszuk, K. Tchoń, and Z. Pietrowska, "Motion planning of the trident snake robot equipped with passive or active wheels", *Bull. Pol. Ac.: Tech.* 60 (3), 547–555 (2012).

[14] E.D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, Springer, New York, 1998.

[15] K. Tchoń, "Continuation method in robotics", *7th Conf. Computer Methods and Systems* 1, 17–24 (2007).

[16] A. Ratajczak and K. Tchoń, "Multiple-task motion planning of non-holonomic systems with dynamics", *Mech-Sci* 4 (1), 153–166 (2013).

[17] A. Ratajczak and K. Tchoń, "Parametric and non-parametric Jacobian motion planning for non-holonomic robotic systems", *J. Intell. Robot. Syst.* 1, 1–12 (2013).

[18] A. Ratajczak, "Output trajectory reproduction in nonholonomic systems", eds. K. Tchoń and C. Zieliński, *Advances in Robotics*, vol. 2, pp. 671–680, OWPW, Warsaw, 2014.

[19] I. Fantoni and R. Lozano, *Non-linear Control for Underactuated Mechanical Systems*, Springer, London, 2002.

[20] M.W. Spong, "Underactuated mechanical systems", ed. K.P. Valavanis, *Control Problems in Robotics and Automation*, Springer, London, 1998.