BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 60, No. 3, 2012
DOI: 10.2478/v10175-012-0067-9

Motion planning of the trident snake robot equipped with passive or active wheels

D. PASZUK* , K. TCHOŃ, and Z. PIETROWSKA

Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology,
11/17 Janiszewskiego St., 50-372 Wrocław, Poland

Abstract. We study the kinematics of the trident snake robot equipped with either active joints and passive wheels or passive joints and active wheels. A control system representation of the kinematics is derived, and control singularities examined. Two motion planning problems are addressed, corresponding to diverse ways of controlling the robot, and solved by means of the endogenous configuration space approach. The constraints imposed by the presence of control singularities are handled using the imbalanced Jacobian algorithm assisted by an auxiliary feedback. Performance of the motion planning algorithms is demonstrated by computer simulations.

Key words: trident snake, motion planning, singularities, endogenous configuration space approach.

1. Introduction

The trident snake robot has been designed as a sort of testbed for examining motion planning algorithms of nonholonomic systems [1]. The robot can be regarded as a wheel-legged robot [2] whose wheels and leggs are used simultaneously instead of alternatively. The original robot has had the form of a triangular body equipped with three articulated branches or arms fastened to its vertexes. Each branch consists of a number of links connected by rotational joints: active or passive. Each link is supported by a wheel moving without the lateral slip. The kinematics of the trident snake are represented by a driftless control system with three controls that can be related either to the position and orientation of the body or to the positions of the active joints. The challenge of the motion planning problem for this kind of robot results primarily from the presence of very complex singular configurations. The robot recently built in our laboratory is shown in Fig. 1. The robot's design has been described in [3].

Fig. 1. Trident snake robot

For the trident snake with a single link in every branch a Lie algebraic approach to the motion planning problem, illustrated with experimental results, has been presented in [4]. The paper [5] offers a practical trajectory tracking algorithm, based on the transverse function approach, and coping with control singularities by forcing the joint angles to stay close to the zero position. Besides the transverse function approach, the motion planning of trident snake can be done by means of the Lie algebraic approach [6]. In [7] the trident snake with a double link in every branch has been studied, and a periodic control algorithm is proposed in a subspace spanned by the second order Lie brackets of the control vector fields. A systematic derivation of kinematics representations of the trident snake with multi-link branches, along with a motion planning algorithm based on the endogenous configuration space approach has been developed in [8].

The purpose of this paper is to further advance the motion planning strategies for the trident snake robot, based on selected results obtained by the first author of this paper in [9]. Specifically, we allow the trident snake to be equipped with either passive or active wheels, that corresponds to the motion without either solely the lateral or both the lateral and the longitudinal slip. Two motion planning problems are addressed: the trident snake equipped with active joints and passive wheels will be controlled by its joint angles, while in the case of passive joints and active wheels the control will involve the rolling angles of the wheels. In the case of the joint and rolling angle control the control singularities of the trident snake need to be avoided. The motion planning problems are solved by means of the endogenous configuration space approach [10]. The singularities have been handled using the imbalanced Jacobian motion planning algorithm [11] combined with an auxiliary feedback strategy. Performance of

*e-mail: dorota.paszuk@pwr.wroc.pl

the designed motion planning algorithms has been illustrated by numeric computations. The main contribution of this paper is threefold: the derivation of the trident snake kinematics with active wheels, the introduction of the auxiliary feedback motion planning strategy, and the application of the imbalanced Jacobian algorithm of motion planning. It should be borne in mind that the motion planning results in defining an admissible robot trajectory that further will serve as a reference trajectory for the robot controller.

The paper is composed in the following way. Section 2 contains the derivation of the control system representations of kinematics of the trident snake. The motion planning problems, singularities, and the motion planning strategies are presented in Sec. 3. Section 4 is devoted to computer simulations. The paper is concluded with Sec. 5.

2. Models

In this section we provide kinematics models of the trident snake robot moving without either the lateral or both the lateral and the longitudinal slip. First, we shall derive the constraints in the Pfaffian form $\mathcal{A}(q)\dot{q} = 0$, then a control system representation of the kinematics will be given in the form of a driftless control system $\dot{q} = G(q)u$. A geometric schematic of the trident snake has been displayed in Fig. 2. The generalized coordinates $q = (\xi^T, \phi^T)^T \in \mathbb{R}^6$, where $\xi = (x, y, \theta)^T$ and $\phi = (\phi_1, \phi_2, \phi_3)^T$ describe, respectively, the position and orientation of the body relative to an external coordinate frame, and the angular positions of the branches. Assuming that a body coordinate frame is placed at the body center, and the robot's orientation is measured between the X axes of these two frames, the angles $\alpha_1 = -\frac{2\pi}{3}$, $\alpha_2 = 0$, $\alpha_3 = \frac{2\pi}{3}$ describe the position of the triangle's vertices with respect to the X axis of the body frame. Each branch is supported by a wheel, either passive or active.

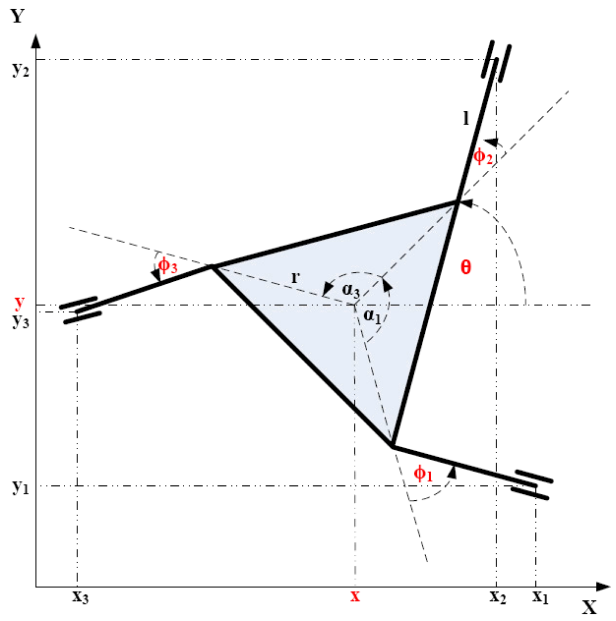


Fig. 2. Trident snake robot: schematic

2.1. Motion without lateral slip. On condition that the wheels are passive and cannot slip laterally, the kinematic constraints can be given the Pfaffian form

$$\begin{bmatrix} \sin(\alpha_1+\phi_1) & -\cos(\alpha_1+\phi_1) & -l-r\cos\phi_1 \\ \sin(\alpha_2+\phi_2) & -\cos(\alpha_2+\phi_2) & -l-r\cos\phi_2 \\ \sin(\alpha_3+\phi_3) & -\cos(\alpha_3+\phi_3) & -l-r\cos\phi_3 \end{bmatrix} \cdot Rot^T(Z,\theta)\dot{\xi} - l\dot{\phi} = \mathcal{A}(q)\dot{q} = 0, \tag{1}$$

where $Rot(Z,\theta)$ denotes the rotation with respect to the Z axis. The Pfaffian constraints yield a control system representation of the trident snake kinematics in the form of a driftless control system

$$\dot{q} = G(q)u,$$

$u = (u_1, u_2, u_3)^T \in \mathbb{R}^3$ denoting the control, defined by the matrix $G(q)$ whose columns span the null space of $\mathcal{A}(q)$, i.e. $\mathcal{A}(q)G(q) = 0$. The following matrix $G(q)$ has been found in [1]

$$G(q) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \\ \frac{1}{l}\sin(\phi_1+\alpha_1) & -\frac{1}{l}\cos(\phi_1+\alpha_1) & -\frac{1}{l}(l+r\cos\phi_1) \\ \frac{1}{l}\sin(\phi_2+\alpha_2) & -\frac{1}{l}\cos(\phi_2+\alpha_2) & -\frac{1}{l}(l+r\cos\phi_2) \\ \frac{1}{l}\sin(\phi_3+\alpha_3) & -\frac{1}{l}\cos(\phi_3+\alpha_3) & -\frac{1}{l}(l+r\cos\phi_3) \end{bmatrix} = \begin{bmatrix} G_1(\theta) \\ G_2(\phi) \end{bmatrix}. \tag{2}$$

The matrix (2) corresponds to the trident snake whose controls have the meaning of rotated position and orientation velocities $\dot{\xi}$. This control strategy will be referred to as the position and orientation control. An alternative strategy is to control the branch joint angle velocities $\dot{\phi}$, that will be called the joint angle control. To obtain a kinematics representation suitable to the joint angle control strategy, we need to employ a feedback $v = G_2(\phi)u$ arriving at

$$G_\phi(q) = \begin{bmatrix} G_1(\theta)G_2(\phi)^{-1} \\ I_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\det G_2} \begin{bmatrix} f_1 & f_2 & f_3 \\ g_1 & g_2 & g_3 \\ h_1 & h_2 & h_3 \end{bmatrix} \\ I_3 \end{bmatrix}, \tag{3}$$

where I_3 denotes the 3×3 unit matrix. Entries of the matrix (3) are defined in Appendix. The feedback $v = G_2(\phi)u$ is well defined outside certain control singularities that will be examined in more detail in the next section.

2.2. Motion without lateral and longitudinal slip. The robot's geometry remains the same as in Fig. 2, but now it is assumed that the wheels are active and not permitted to slip either laterally or longitudinally. The vector of generalized coordinates $q = (\xi^T, \phi^T, \beta^T)^T \in \mathbb{R}^9$, where $\beta = (\beta_1, \beta_2, \beta_3)^T$

collects the rolling angles of the wheels. The lack of lateral slip requires that for each wheel its velocity components obey the identity

$$\dot{x}_i \sin(\theta+\alpha_i+\phi_i) - \dot{y}_i \cos(\theta+\alpha_i+\phi_i) = 0, \quad i = 1,2,3. \tag{4}$$

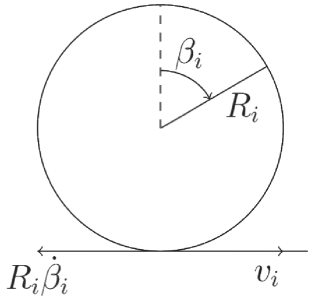


Fig. 3. The i-th wheel of the robot

The contact point coordinates (x_i, y_i) of the ith wheel should be expressed in the generalized coordinates q of the robot, so

$$\begin{aligned} x_i &= x + r\cos(\theta+\alpha_i) + l\cos(\theta+\alpha_i+\phi_i), \\ y_i &= y + r\sin(\theta+\alpha_i) + l\sin(\theta+\alpha_i+\phi_i). \end{aligned} \tag{5}$$

The velocities are obtained by differentiation of (5). The insertion into (4) yields

$$\begin{aligned} &(\dot{x} - \dot{\theta} r\sin(\theta+\alpha_i) - l\sin(\theta+\alpha_i+\phi_i) \\ &\cdot(\dot{\theta}+\dot{\phi}))\sin(\theta+\alpha_i+\phi_i) - (\dot{y} + \dot{\theta} r\cos(\theta+\alpha_i) \\ &+ l\cos(\theta+\alpha_i+\phi_i)(\dot{\theta}+\dot{\phi}))\cos(\theta+\alpha_i+\phi_i) = 0, \end{aligned}$$

leading finally to the Pfaffian form corresponding to (1). In order to exclude the longitudinal slip, the wheels will have to satisfy a condition

$$v_i - R_i\dot{\beta}_i = 0, \tag{6}$$

where v_i stands for the linear velocity of the i-th wheel, and R_i is the wheel's radius. After computing the linear velocity as

$$v_i = \dot{x}_i\cos(\alpha_i+\phi_i+\theta) + \dot{y}_i\sin(\alpha_i+\phi_i+\theta) \tag{7}$$

and invoking (5), we obtain

$$\begin{aligned} &\dot{x}\cos(\phi_i+\theta+\alpha_i) + \dot{y}\sin(\phi_i+\theta+\alpha_i) \\ &+\dot{\theta} r\sin\phi_i - R_i\dot{\beta}_i = 0. \end{aligned} \tag{8}$$

Assuming $R_1 = R_2 = R_3 = R$, the condition that all the wheels will not slip longitudinally can be written in the Pfaffian form as follows

$$\begin{aligned} &\begin{bmatrix} \cos(\phi_1+\alpha_1+\theta) & \sin(\phi_1+\alpha_1+\theta) & r\sin\phi_1 \\ \cos(\phi_2+\alpha_2+\theta) & \sin(\phi_2+\alpha_2+\theta) & r\sin\phi_2 \\ \cos(\phi_3+\alpha_3+\theta) & \sin(\phi_3+\alpha_3+\theta) & r\sin\phi_3 \end{bmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \\ &-R\begin{pmatrix} \dot{\beta}_1 \\ \dot{\beta}_2 \\ \dot{\beta}_3 \end{pmatrix} = \mathcal{A}(q)\dot{q}. \end{aligned} \tag{9}$$

The control matrix $G(q)$ has been found analogously to the matrix (2)

$$\begin{aligned} G(q) = &\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \\ \frac{1}{l}\sin(\phi_1+\alpha_1) & -\frac{1}{l}\cos(\phi_1+\alpha_1) & -\frac{1}{l}(l+r\cos\phi_1) \\ \frac{1}{l}\sin(\phi_2+\alpha_2) & -\frac{1}{l}\cos(\phi_2+\alpha_2) & -\frac{1}{l}(l+r\cos\phi_2) \\ \frac{1}{l}\sin(\phi_3+\alpha_3) & -\frac{1}{l}\cos(\phi_3+\alpha_3) & -\frac{1}{l}(l+r\cos\phi_3) \\ \frac{1}{R}\cos(\phi_1+\alpha_1) & \frac{1}{R}\sin(\phi_1+\alpha_1) & \frac{r}{R}\sin\phi_1 \\ \frac{1}{R}\cos(\phi_2+\alpha_2) & \frac{1}{R}\sin(\phi_2+\alpha_2) & \frac{r}{R}\sin\phi_2 \\ \frac{1}{R}\cos(\phi_3+\alpha_3) & \frac{1}{R}\sin(\phi_3+\alpha_3) & \frac{r}{R}\sin\phi_3 \end{bmatrix} \\ = &\begin{bmatrix} G_1(\theta) \\ G_2(\phi) \\ G_3(\phi) \end{bmatrix}. \end{aligned} \tag{10}$$

The control system based on matrix (10) is tailored to the position and orientation control strategy. If the system is to be controlled by the rolling angles velocities $\dot{\beta}$ of the wheels, referred to as the rolling angle control, we need to apply a feedback $v = G_3(\phi)u$ resulting in the following control matrix

$$G_\beta(q) = \begin{bmatrix} G_1(\theta)G_3^{-1}(\phi) \\ G_2(\phi)G_3^{-1}(\phi) \\ I_3 \end{bmatrix} = \begin{bmatrix} \frac{f_{11}}{2\Delta} & \frac{f_{12}}{2\Delta} & \frac{f_{13}}{2\Delta} \\ \frac{f_{21}}{2\Delta} & \frac{f_{22}}{2\Delta} & \frac{f_{23}}{2\Delta} \\ \frac{f_{31}}{r\Delta} & \frac{f_{32}}{r\Delta} & \frac{f_{33}}{r\Delta} \\ \frac{f_{41}}{rl\Delta} & \frac{f_{42}}{rl\Delta} & \frac{f_{43}}{rl\Delta} \\ \frac{f_{51}}{rl\Delta} & \frac{f_{52}}{rl\Delta} & \frac{f_{53}}{rl\Delta} \\ \frac{f_{61}}{rl\Delta} & \frac{f_{62}}{rl\Delta} & \frac{f_{63}}{rl\Delta} \\ & I_3 & \end{bmatrix}. \tag{11}$$

Entries of matrix (11) have been defined in Appendix. Again, the feedback is well defined provided that matrix $G_3(\phi)$ has full rank. The corresponding control singularities will be characterized in more detail in the next section.

3. Motion planning

This section is devoted to the statement of the motion planning problems for the trident snake robot, and to the presentation of the motion planning strategies. Also, a characterization of singularities is provided.

3.1. Problems. In the previous section we have derived the control system representations of the trident snake kinematics in the form of a driftless control system. By adding an output function we obtain the control system with output

$$\begin{cases} \dot{q} = G(q)u, \\ y = k(q), \end{cases} \tag{12}$$

where $q \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^s$ denote, respectively, the configuration, control, and output variables. The control matrix $G(q)$ assumes either of the forms (2), (3), (10) or (11). The control function $u(t)$ determines the state trajectory $q(t) = \varphi_{q_0,t}(u(\cdot))$ and the output trajectory $y(t) = k(q(t))$. Given the system (12), we shall address the following motion planning problem: given a task space point $y_d \in \mathbb{R}^s$ and a time horizon $T > 0$, find a control function $u(t)$, such that $y(T) = y_d$. As we have already noticed, the motion planning problem for the position and orientation control does not involve singularities, whereas adopting either the joint or the rolling angle control strategy requires to take into account the control singularities.

3.2. Singularities.

Joint angle control. In this case the control singularities occur when the matrix

$$G_2(\phi) = \begin{bmatrix} \frac{1}{l}\sin(\phi_1+\alpha_1) & -\frac{1}{l}\cos(\phi_1+\alpha_1) & -\frac{1}{l}(l+r\cos\phi_1) \\ \frac{1}{l}\sin(\phi_2+\alpha_2) & -\frac{1}{l}\cos(\phi_2+\alpha_2) & -\frac{1}{l}(l+r\cos\phi_2) \\ \frac{1}{l}\sin(\phi_3+\alpha_3) & -\frac{1}{l}\cos(\phi_3+\alpha_3) & -\frac{1}{l}(l+r\cos\phi_3) \end{bmatrix} \tag{13}$$

becomes singular. This is tantamount to the condition

$$(l+r\cos\phi_1)\sin\left(\phi_3-\phi_2+\frac{2}{3}\pi\right)$$

$$+(l+r\cos\phi_2)\sin\left(\phi_1-\phi_3+\frac{2}{3}\pi\right)$$

$$+(l+r\cos\phi_3)\sin\left(\phi_2-\phi_1+\frac{2}{3}\pi\right)=0.$$

The Fig. 4 illustrates the dependence of $\det G_2(\phi) = 0$ on the joint angles ϕ_i, $i = 1, 2, 3$, varying over the interval $[-\pi, \pi]$, for the unit lengths $l, r = 1$. The surface $\det G_2(\phi) = 0$ divides the cube $[-\pi, \pi]^3$ into two regions: $\det G_2(\phi) < 0$ and $\det G_2(\phi) > 0$, presented in Fig. 5. Apparently, the former region contains the cube $\left[-\frac{\pi}{3}, \frac{\pi}{3}\right]^3$ identified in [1, 4] as a singularity–free area. This provides a rationale for keeping the robot's joints within the region $\det G_2(\phi) < 0$.

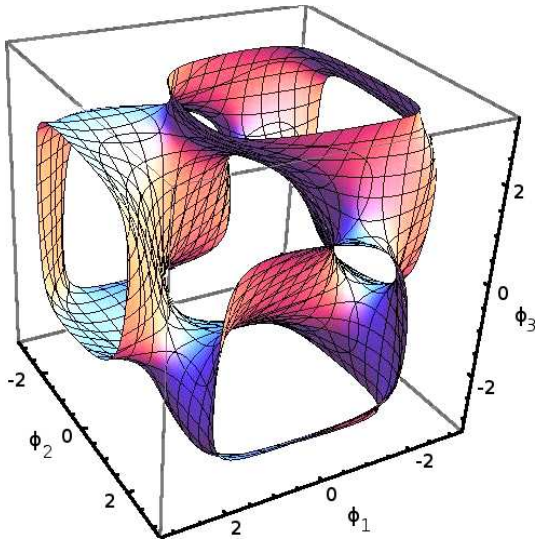
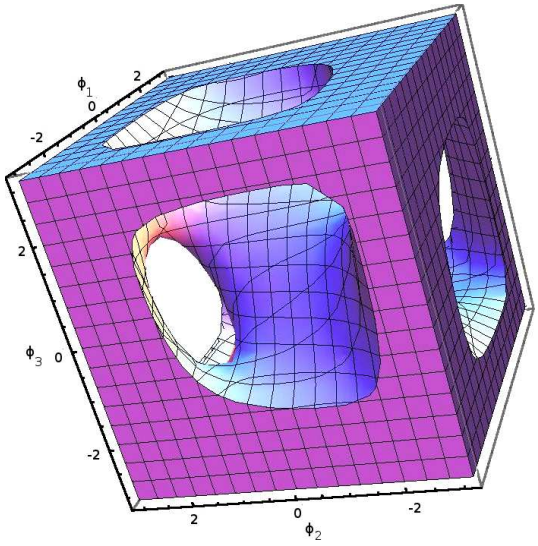


Fig. 4. Joint angle control: the surface $\det G_2(\phi) = 0$

a)



b)

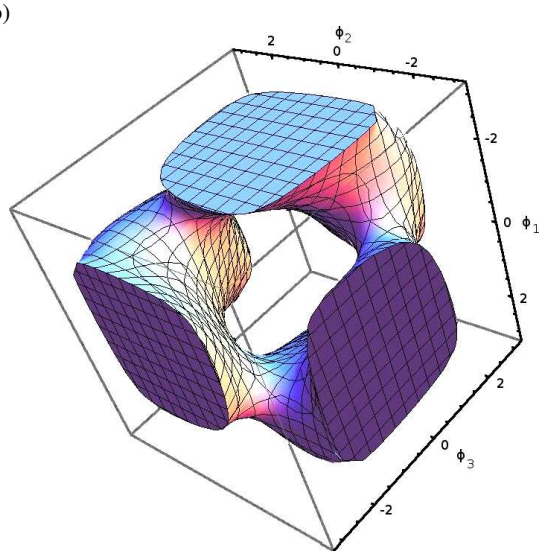


Fig. 5. Joint angle control: a) the region of $\det G_2(\phi) < 0$, b) the region of $\det G_2(\phi) > 0$

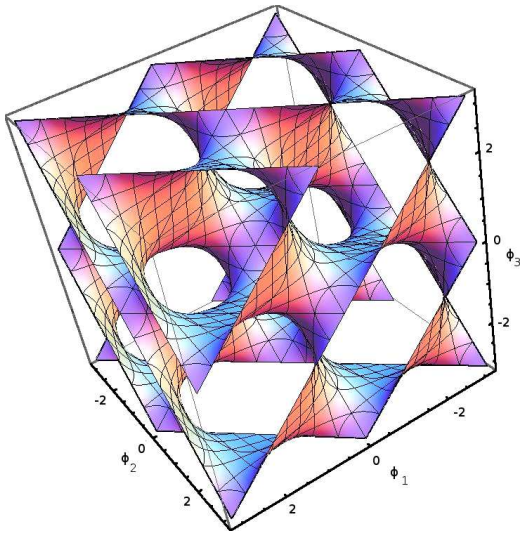
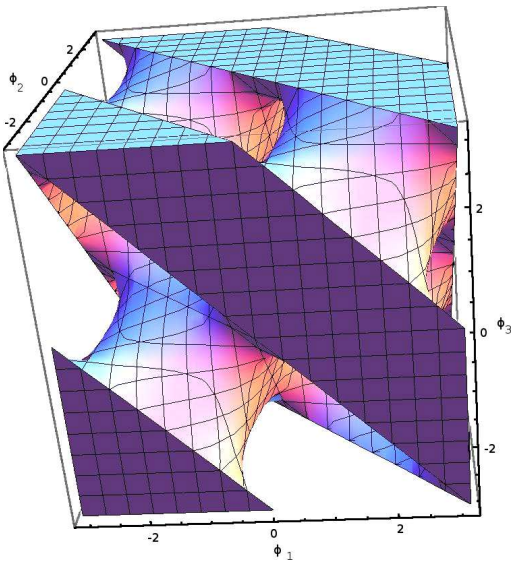


Fig. 6. Rolling angle control: the singularity locus $\det G_3(\phi) = 0$

a)



b)

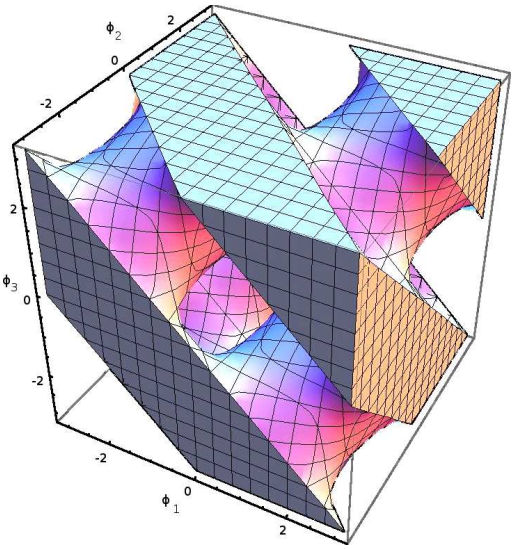


Fig. 7. Rolling angle control: a) the region of $\det G_3(\phi) < 0$, b) the region of $\det G_3(\phi) > 0$

Rolling angle control. The singularity locus is now defined by the identity $\det G_3(\phi) = 0$, where

$$G_3(\phi) = \begin{bmatrix} \frac{1}{R}\cos(\phi_1+\alpha_1) & \frac{1}{R}\sin(\phi_1+\alpha_1) & \frac{r}{R}\sin\phi_1 \\ \frac{1}{R}\cos(\phi_2+\alpha_2) & \frac{1}{R}\sin(\phi_2+\alpha_2) & \frac{r}{R}\sin\phi_2 \\ \frac{1}{R}\cos(\phi_3+\alpha_3) & \frac{1}{R}\sin(\phi_3+\alpha_3) & \frac{r}{R}\sin\phi_3 \end{bmatrix}. \tag{14}$$

The surface

$$\sin\phi_3 \sin\left(\phi_2 - \phi_1 + \frac{2}{3}\pi\right) + \sin\phi_1 \sin\left(\phi_3 - \phi_2 + \frac{2}{3}\pi\right) + \sin\phi_2 \sin\left(\phi_1 - \phi_3 + \frac{2}{3}\pi\right) = 0$$

corresponding to $\det G_3(\phi) = 0$ has been displayed in Fig. 6 for the joint angles ϕ_i ranging within the interval $[-\pi, \pi]$. As before, this surface defines two regions: $\det G_3(\phi) < 0$ and $\det G_3(\phi) > 0$, shown in Fig. 7. For a further study of motion planning we have chosen the region $\det G_3(\phi) < 0$ encompassing the origin.

3.3. Motion planning algorithms. Since the motion planning problem for the position and orientation control of the trident snake does not involve control singularities, it may be solved with the help of a Jacobian inverse kinematics algorithm provided by the endogenous configuration space approach [10]. The situation is quite different, when the joint or rolling angle control is used. In this case, the presence of singularities imposes severe constraints on the motion planning algorithm. To cope with these constraints, we shall use the imbalanced Jacobian motion planning algorithm, based on the Jacobian pseudo inverse [11]. The main idea of this algorithm will be sketched below. We begin with the system (12). To guarantee that its trajectory stays within the set of regular configurations, an additional variable will be added to the system, whose evolution is governed by the violation of constraints. Let the regular region be defined as $c(q) \leq -\epsilon$, for a positive number ϵ, where either $c(q) = \det G_2(q)$ or $c(q) = \det G_3(q)$. Then, the extended system associated with (12) will assume the following form

$$\begin{cases} \dot{q}_e = G_e(q_e)u = \begin{bmatrix} G(q) \\ p(\epsilon + c(q), \alpha) \end{bmatrix} u, \\ y_e = k_e(q_e) = (k(q), q_{n+1}), \end{cases} \tag{15}$$

where $q_e = (q^T, q_{n+1})^T$, $y_e = (y^T, q_{n+1})^T$, $u \in \mathbb{R}^m$ is the same as in (12), and the function $p(x, \alpha) = x + \frac{1}{\alpha}\ln(1 + \exp(-\alpha x))$, $\alpha > 0$, smoothly approximates the plus function $max\{x, 0\}$.

The motion planning problem in the extended system takes the following formulation: given a task space point y_d and a time $T > 0$, find a control function $u(t)$, such that $y_e(T) = y_{ed} = (y_d, 0)^T$. This is tantamount to requiring that $u(t)$ solves the original motion planning problem and simultaneously makes $c(q) \leq -\epsilon$. Assuming that the control functions of (15) belong to the Hilbert space of Lebesgue square

integrable functions on $[0,T]$, we define the endogenous configuration space as $\mathcal{X} = L^2_m[0,T]$. For the fixed initial state q_{e0} and the control time T, the end point map of the extended system

$$K_e : \mathcal{X} \to \mathbb{R}^s$$

assigns to every control function a task space point, $K_e(u(\cdot)) = y_e(T) = k_e(\varphi_{q_{e0},T}(u(\cdot)))$. The motion planning problem in the extended system consists of defining a control function $u(t)$, such that $K_e(u(\cdot)) = y_{ed}$. This problem can be solved by means of the Jacobian pseudo inverse algorithm, whose associated dynamic system takes the form

$$\begin{aligned}\frac{du_\vartheta(t)}{d\vartheta} &= -\gamma C^T_{e\vartheta}(T)B^T_{e\vartheta}(t)\Phi^T_{e\vartheta}(T,t)\\ &\cdot\mathcal{M}_e^{-1}(u_\vartheta(\cdot))(K_e(u_\vartheta(\cdot)) - y_{ed}).\end{aligned} \tag{16}$$

In (16) $u_\vartheta(\cdot)$, $\vartheta \in \mathbb{R}$ is a smooth curve of control functions. Given a control–trajectory pair $(u(t), q_e(t))$ of the extended system, we define the matrices

$$B_e(t) = G_e(q_e(t)), \qquad A_e(t) = \frac{\partial G_e(q_e(t))u(t)}{\partial q_e},$$

$$C_e(t) = \frac{\partial k_e(q_e(t))}{\partial q_e},$$

and find $\Phi_e(t,s)$ by solving the evolution equation

$$\frac{d\Phi(t,s)}{dt} = A_e(t)\Phi_e(t,s)$$

with initial condition $\Phi_e(s,s) = I_{n+1}$. Finally, the mobility matrix is equal to

$$\mathcal{M}_e(u(\cdot)) = C_e(T)\int_0^T B_e(t)\Phi_e(T,t)\Phi_e^T(T,t)B_e^T(t)dtC_e^T(T). \tag{17}$$

The dynamic system (16) evolves in the endogenous configuration space, starting from an initial control function $u_0(t)$. A solution to the motion planning problem is obtained by passing to the limit with the trajectory of (16), $u_d(t) = \lim_{\vartheta\to+\infty} u_\vartheta(t)$. However, it has been observed that when the constraints are satisfied, the mobility matrix of the extended system becomes singular. To overcome this problem, in [12] the extended system has been subject to regularization. In the case of (15) the regularization will be achieved by adding to the last differential equation the quadratic function $\rho(q) = \frac{1}{2}\phi^T\phi$, $\phi = (\phi_1,\phi_2,\phi_3)^T$ denoting the joint angles. The regularized system

$$\begin{cases}\dot q_r = G_r(q_r)u = \begin{bmatrix} G(q) \\ p(\epsilon + c(q),\alpha) + \rho(q)\end{bmatrix} u,\\ y_r = k_r(q) = (k(q), q_{n+1}).\end{cases} \tag{18}$$

The essential idea of the imbalanced Jacobian algorithm is to combine in the associated dynamic system corresponding to the Jacobian pseudo inverse algorithm for (18) the dynamic data coming from the regularized system with the error provided by the extended system. The resulting associated system takes the following form

$$\begin{aligned}\frac{du_\vartheta(t)}{d\vartheta} &= -\gamma C^T_{r\vartheta}(T)B^T_{r\vartheta}(t)\Phi^T_{r\vartheta}(T,t)\\ &\cdot\mathcal{M}_r^{-1}(u_\vartheta(\cdot))(K_e(u_\vartheta(\cdot)) - y_{ed}).\end{aligned} \tag{19}$$

All the data appearing in (19) need to be computed along the pair $(u(t), q_r(t))$, analogously as for the extended system. Again, a solution of the motion planning problem is the limit $u_d(t) = \lim_{\vartheta\to+\infty} u_\vartheta(t)$. Sufficient conditions under which the imbalanced Jacobian algorithm solves the original motion planning problem have been provided in [11].

Usually, in practical computations the control functions are represented by a finite dimensional space of coefficients of truncated expansions with respect to an orthogonal basis in the endogenous configuration space [13]. In this paper we have chosen the Fourier basis, so that

$$u(\lambda,t) = \lambda_0 + \sum_{j=1}^{w}\lambda_{2j-1}\sin j\omega t + \lambda_{2j}\cos j\omega t = P(t)\lambda, \tag{20}$$

where

$$u(\lambda,t), \lambda_j \in \mathbb{R}^m,$$

$$\lambda = (\lambda_0,\lambda_1,\ldots,\lambda_{2w})^T \in \mathbb{R}^{(2w+1)m}, \qquad \omega = \frac{2\pi}{T},$$

and the matrix $P(t)$ collects the basis functions. After the substitution of (20) into (19) we derive a finite dimensional and discrete version of the imbalanced Jacobian algorithm

$$\lambda(i+1) = \lambda(i) - \gamma J_r^{\#P}(\lambda(i))e(\lambda(i)), \tag{21}$$

where $i \geq 0$ denotes the step number, the error $e(\lambda) = K_e(u(\lambda,\cdot)) - y_{ed}$, and

$$J_r^{\#P}(\lambda) = J_r^T(\lambda)\left(J_r(\lambda)J_r^T(\lambda)\right)^{-1}. \tag{22}$$

Observe that here above $J_r(\lambda)J_r^T(\lambda)$ represents the mobility matrix (17). The Jacobian matrix in (22) is defined as

$$J_r(\lambda) = C_{r\lambda}(T)S_{r\lambda}(T), \tag{23}$$

whereas the matrix $S_{r\lambda}(t)$ satisfies the matrix differential equation

$$\frac{dS_{r\lambda}(t)}{dt} = A_{r\lambda}(t)S_{r\lambda}(t) + B_{r\lambda}(t)P(t), \tag{24}$$

with initial condition $S_{r\lambda}(0) = 0$. The matrices $A_{r\lambda}(t)$, $B_{r\lambda}(t)$ and $C_{r\lambda}(t)$ refer to the regularized system (18) steered by the control function (20).

3.4. Auxiliary feedback strategy. As we have already said, the motion planning problem for the trident snake controlled by the joint or the rolling angles is difficult due to the presence of singularities. On the other hand, the motion planning in the case of position and orientation control is non-singular. This being so, we have proposed the following motion planning strategy: we are solving the problem for the position and orientation control, simultaneously preserving non-singularity of the corresponding feedback matrix $G_2(\phi)$ or $G_3(\phi)$, see (13) or (14). It turns out that this problem is solvable by means of the imbalanced Jacobian algorithm. Then, the motion planning problem for the joint angles or the rolling angles can be solved using an auxiliary feedback. Suppose that we have solved the motion planning problem for the position and orientation control in the system

$$\begin{cases} \dot{q} = G(q)u, \\ y = k(q), \end{cases} \tag{25}$$

where $G(q)$ is given by either (2) or (10), resulting in a control $u(t)$ that makes non-singular the matrix $G_2(\phi)$ or $G_3(\phi)$. Then, the motion planning problem for the joint angles

$$\begin{cases} \dot{q} = G_\phi(q)v = \begin{bmatrix} G_1(\vartheta)G_2^{-1}(\phi) \\ I_3 \end{bmatrix} v, \\ y = k(q) = (\xi^T, \phi^T)^T, \end{cases} \tag{26}$$

or for the rolling angles

$$\begin{cases} \dot{q} = G_\beta(q)v = \begin{bmatrix} G_1(\vartheta)G_3^{-1}(\phi) \\ G_2(\phi)G_3^{-1}(\phi) \\ I_3 \end{bmatrix} v, \\ y = k(q) = (\xi^T, \phi^T)^T, \end{cases} \tag{27}$$

has the solution

$$v(t) = G_2(\phi(t))u(t) \quad \text{or} \quad v(t) = G_3(\phi(t))u(t). \tag{28}$$

The efficiency of this approach will be illustrated in the next section by numerical computations.

4. Computer simulations

As an illustration of the motion planning algorithms presented in the previous section, we shall solve the same example motion planning problem, first for the trident snake with active joints and passive wheels, and then for the robot with passive joints and active wheels. For the joint and the rolling angle control the kinematics representations (26) and (27) will be employed, and the motion planning problem will be solved in accordance with the auxiliary feedback strategy described in the Subsec. 3.4. This means that we shall first find the position and orientation control for the representation (25) respecting the constraints $\det G_2(q) \leq -\epsilon$ or $\det G_3(q) \leq -\epsilon$ and then apply the feedback (28).

In computations the control function of the form (20) has been employed, over the control horizon $T = 2$, depending on control coefficients $\lambda \in \mathbb{R}^{15}$ ($m = 3$, $w = 2$). Geometric parameters of the robot are taken as $l = r = 1$, $R = 0.1$. The update of λ will proceed in accordance with the imbalanced Jacobian pseudo inverse algorithm defined by (21), starting from the initial

$$\lambda(0) = (0.5, 0.3, 0.3, 0.3, 0.3, -0.5, 0.3, 0.3, 0.3, 0.3, -0.5, 0.3, 0.3, 0.3, 0.3)^T.$$

The starting points are

$$q_0 = (-\sqrt{2}/2, \sqrt{2}/2, 0, -\pi/6, -\pi/6, -\pi/6)^T$$

for the joint angle control and

$$q_0 = (-\sqrt{2}/2, \sqrt{2}/2, 0, -\pi/6, -\pi/6, -\pi/6, 0, 0, 0)^T$$

for the rolling angle control.

$$y_d = (0, 0, 0, -\pi/6, -\pi/6, -\pi/6)^T$$

is chosen as the final task space point. The simulations terminate, when the task space error drops below 0.01. To prevent the ill-conditioning of the algorithm close to the singularities of the regularized system, the singularity robust Jacobian pseudo inverse is used [10], whenever necessary, relying on the addition to the mobility matrix appearing in (22) a term κI_{s+1} with $\kappa = 0.01$. The convergence coefficient $\gamma = 0.5$, and the constraint coefficient $\epsilon = 0.1$. The results of computations have been summarized in Figs. 8 and 9. The requested accuracy has been achieved after, respectively, 9 and 40 iterations. It is concluded that both the control strategies have provided a correct solution to the problem.

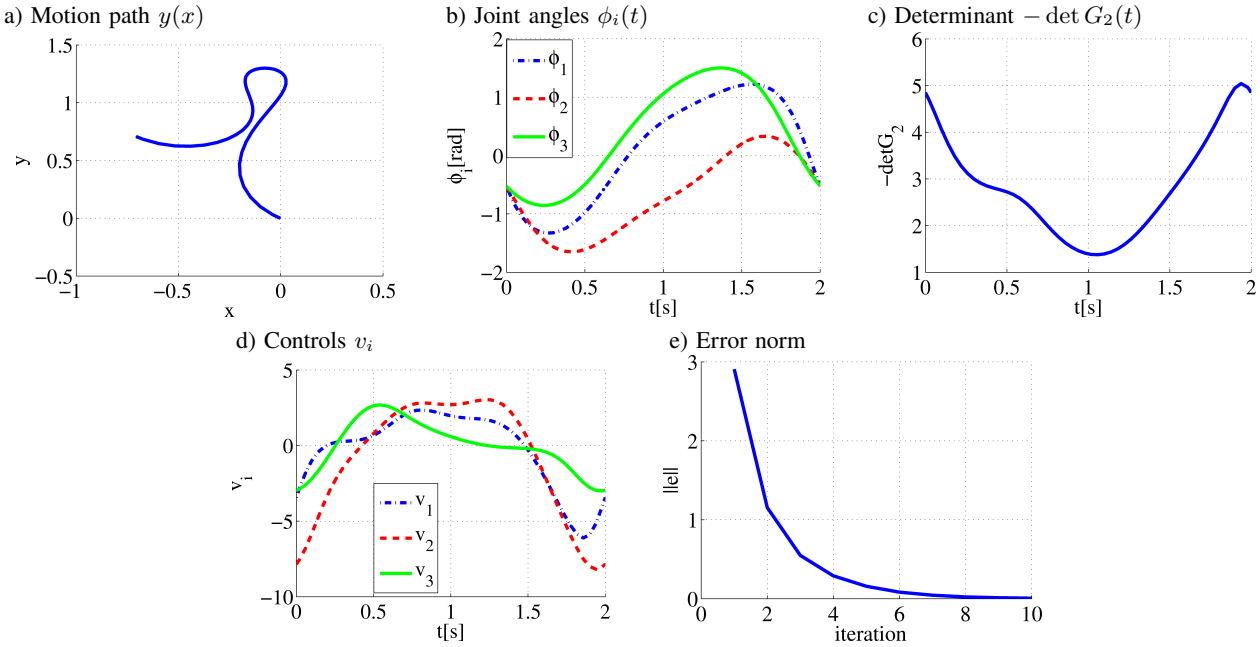


Fig. 8. No lateral slip, joint angle control

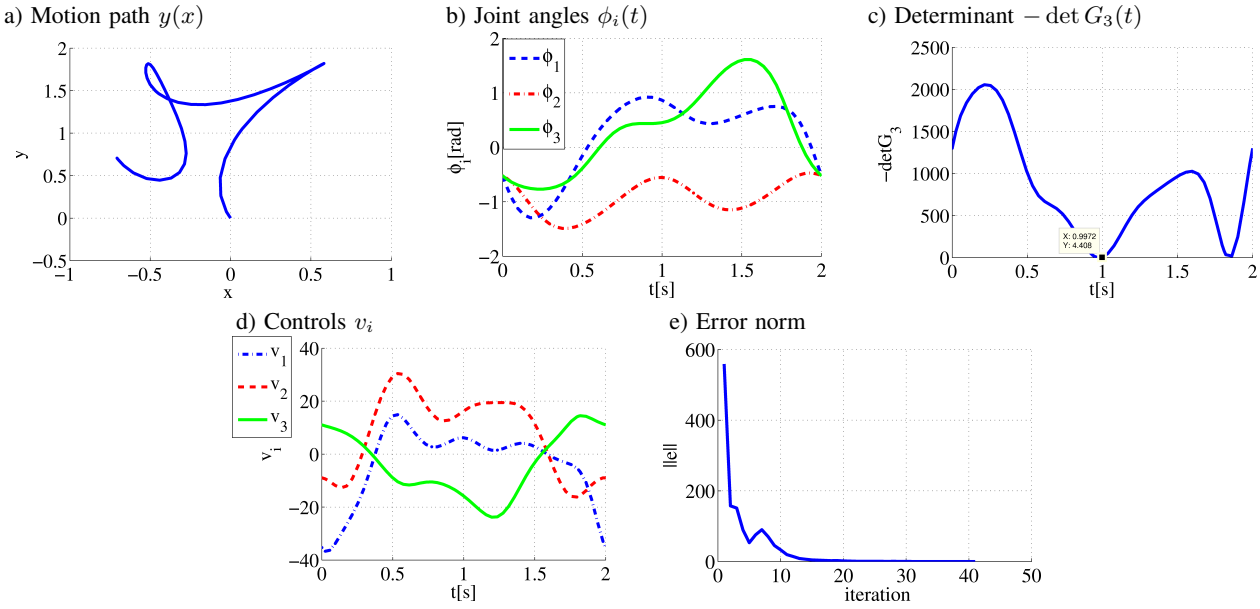


Fig. 9. No lateral and longitudinal slip, rolling angle control

5. Conclusions

For the trident snake robot with active joints and passive wheels or passive joints and active wheels, controlled by either the joint or the rolling angles, we have devised the motion planning algorithms capable of respecting the control singularities. Advantages of the feedback strategy have been demonstrated. Presented results extend the range of applicability of the endogenous configuration space approach. Future research will focus on the dynamics model of the trident snake and on the implementation of the motion planning algorithms on the physical model of the trident snake robot presented in Fig. 1.

Appendix

Entries of the matrix (3), for $i = 1, 2, 3$ read modulo 3:

$$f_i = \frac{1}{l^2}[(l + r\cos\phi_{i+2})\cos(\phi_{i+1} + \alpha_{i+1} + \theta) - (l + r\cos\phi_{i+1})\cos(\phi_{i+2} + \alpha_{i+2} + \theta)],$$

$$g_i = -\frac{1}{l^2}[(l + r\cos\phi_{i+1})\sin(\phi_{i+2} + \alpha_{i+2} + \theta) - (l + r\cos\phi_{i+2})\sin(\phi_{i+1} + \alpha_{i+1} + \theta)],$$

$$h_i = -\frac{1}{l^2}\sin(\alpha_{i+1} - \alpha_{i+2} + \phi_{i+1} - \phi_{i+2}),$$

$$\det G_2 = -\frac{1}{l^3}\left[(l + r\cos\phi_1)\sin\left(\phi_3 - \phi_2 + \frac{2}{3}\pi\right) + (l + r\cos\phi_2)\sin\left(\phi_1 - \phi_3 + \frac{2}{3}\pi\right) + (l + r\cos\phi_3)\sin\left(\phi_2 - \phi_1 + \frac{2}{3}\pi\right)\right].$$

Entries of the matrix (11):

$$f_{11} = R(\cos(\alpha_2 + \phi_2 - \phi_3 + \theta) - \cos(\alpha_3 - \phi_2 + \phi_3 + \theta) - \cos(\alpha_2 + \phi_2 + \phi_3 + \theta) + \cos(\alpha_3 + \phi_2 + \phi_3 + \theta)),$$

$$f_{12} = R(-\cos(\alpha_1 + \phi_1 - \phi_3 + \theta) + \cos(\alpha_3 - \phi_1 + \phi_3 + \theta) + \cos(\alpha_1 + \phi_1 + \phi_3 + \theta) - \cos(\alpha_3 + \phi_1 + \phi_3 + \theta)),$$

$$f_{13} = R(\cos(\alpha_1 + \phi_1 - \phi_2 + \theta) - \cos(\alpha_2 - \phi_1 + \phi_2 + \theta) - \cos(\alpha_1 + \phi_1 + \phi_2 + \theta) + \cos(\alpha_2 + \phi_1 + \phi_2 + \theta)),$$

$$f_{21} = R(\sin(\alpha_2 + \phi_2 - \phi_3 + \theta) - \sin(\alpha_3 - \phi_2 + \phi_3 + \theta) - \sin(\alpha_2 + \phi_2 + \phi_3 + \theta) + \sin(\alpha_3 + \phi_2 + \phi_3 + \theta)),$$

$$f_{22} = R(-\sin(\alpha_1 + \phi_1 - \phi_3 + \theta) + \sin(\alpha_3 - \phi_1 + \phi_3 + \theta) + \sin(\alpha_1 + \phi_1 + \phi_3 + \theta) - \sin(\alpha_3 + \phi_1 + \phi_3 + \theta)),$$

$$f_{23} = R(\sin(\alpha_1 + \phi_1 - \phi_2 + \theta) - \sin(\alpha_2 - \phi_1 + \phi_2 + \theta) - \sin(\alpha_1 + \phi_1 + \phi_2 + \theta) + \sin(\alpha_2 + \phi_1 + \phi_2 + \theta)),$$

$$f_{31} = -R\sin(\alpha_2 - \alpha_3 + \phi_2 - \phi_3),$$

$$f_{32} = R\sin(\alpha_1 - \alpha_3 + \phi_1 - \phi_3),$$

$$f_{33} = -R\sin(\alpha_1 - \alpha_2 + \phi_1 - \phi_2),$$

$$f_{41} = R((l + r\cos\phi_1)\cos(\alpha_3 + \phi_3)\sin(\alpha_2 + \phi_2) + r\sin(\alpha_1 + \phi_1)\sin(\alpha_2 + \phi_2)\sin(\phi_3) + \cos(\alpha_1 + \phi_1)(-r\cos(\alpha_3 + \phi_3)\sin\phi_2 + r\cos(\alpha_2 + \phi_2)\sin\phi_3) - l\cos(\alpha_2 + \phi_2)\sin(\alpha_3 + \phi_3) - r\cos\phi_1\cos(\alpha_2 + \phi_2)\sin(\alpha_3 + \phi_3) - r\sin(\alpha_1 + \phi_1)\sin\phi_2\sin(\alpha_3 + \phi_3)),$$

$$f_{42} = -R(r\sin(\alpha_1 - \alpha_3 - \phi_3) + l\sin(\alpha_1 - \alpha_3 + \phi_1 - \phi_3) + r\sin\phi_3),$$

$$f_{43} = R(r\sin(\alpha_1 - \alpha_2 - \phi_2) + l\sin(\alpha_1 - \alpha_2 + \phi_1 - \phi_2) + r\sin\phi_2),$$

$$
\begin{aligned}
f_{51} = R(&r\sin(\alpha_2-\alpha_3-\phi_3)\\
&+ l\sin(\alpha_2-\alpha_3+\phi_2-\phi_3)+r\sin\phi_3),\\
f_{52} = R(&-(l+r\cos(\phi_2))\cos(\alpha_3+\phi_3)\sin(\alpha_1+\phi_1)\\
&- r\sin(\alpha_1+\phi_1)\sin(\alpha_2+\phi_2)\sin\phi_3\\
&+ r\cos(\alpha_2+\phi_2)(\cos(\alpha_3+\phi_3)\sin\phi_1\\
&- \cos(\alpha_1+\phi_1)\sin\phi_3)+l\cos(\alpha_1+\phi_1)\sin(\alpha_3+\phi_3)\\
&+ r\cos(\alpha_1+\phi_1)\cos(\phi_2)\sin(\alpha_3+\phi_3)\\
&+ r\sin\phi_1\sin(\alpha_2+\phi_2)\sin(\alpha_3+\phi_3)),\\
f_{53} = R(&-r\sin\phi_1+r\sin(\alpha_1-\alpha_2+\phi_1)\\
&+ l\sin(\alpha_1-\alpha_2+\phi_1-\phi_2)),\\
f_{61} = R(&-r\sin\phi_2+r\sin(\alpha_2-\alpha_3+\phi_2)\\
&+ l\sin(\alpha_2-\alpha_3+\phi_2-\phi_3)),\\
f_{62} = -R(&-r\sin\phi_1+r\sin(\alpha_1-\alpha_3+\phi_1)\\
&+ l\sin(\alpha_1-\alpha_3+\phi_1-\phi_3)),\\
f_{63} = R(&\cos(\alpha_2+\phi_2)(-r\cos(\alpha_3+\phi_3)\sin\phi_1\\
&+ (l+r\cos(\phi_3))\sin(\alpha_1+\phi_1))+\cos(\alpha_1+\phi_1)\\
&\cdot(r\cos(\alpha_3+\phi_3)\sin\phi_2-(l+r\cos(\phi_3))\sin(\alpha_2+\phi_2))\\
&+ r(\sin(\alpha_1+\phi_1)\sin\phi_2\\
&- \sin\phi_1\sin(\alpha_2+\phi_2))\sin(\alpha_3+\phi_3)),\\
\Delta = &\cos(\alpha_3+\phi_3)(\sin(\alpha_1+\phi_1)\sin\phi_2-\sin\phi_1\sin(\alpha_2+\phi_2))\\
&+\cos(\alpha_2+\phi_2)(-\sin(\alpha_1+\phi_1)\sin\phi_3+\sin\phi_1\sin(\alpha_3+\phi_3))\\
&+\cos(\alpha_1+\phi_1)(\sin(\alpha_2+\phi_2)\sin\phi_3-\sin\phi_2\sin(\alpha_3+\phi_3)).
\end{aligned}
$$

Acknowledgements. The work of the second author has been supported by a statutory grant provided by the Wrocław University of Technology. The authors are indebted to anonymous reviewers whose comments improved the presentation of this paper.

REFERENCES

[1] M. Ishikawa, "Trident snake robot: locomotion analysis and control", *Proc 6th IFAC Symp. NOLCOS* 1, 1169–1174 (2004).

[2] J. Szrek and P. Wojtowicz, "Idea of wheel-legged robot and its control system design", *Bull. Pol. Ac.: Tech.* 58 (1), 43–50 (2010).

[3] Sz. Gospodarek, "Design and modelling of the trident snake type mobile robot", *Master's Thesis*, Wrocław University of Technology, Wrocław, 2011, (in Polish).

[4] M. Ishikawa, Y. Minati, and T. Sugie, "Development and control experiment of the trident snake robot", *IEEE/ASME Trans. on Mechatronics* 15, 9–16 (2010).

[5] M. Ishikawa, P. Morin, and C. Samson, "Tracking control of the trident snake robot with the transverse function approach", *Proc. 48th IEEE CDC* 1, 4137–4143 (2009).

[6] I. Duleba, "Impact of control representations on efficiency of local nonholonomic motion planning", *Bull. Pol. Ac.: Tech.* 59 (2), 213–218 (2011).

[7] M. Ishikawa and T. Fujino, "Control of the double-linked trident snake robot based on the analysis of its oscillatory dynamics", *Proc. 2009 IEEE/RSJ Int. Conf. IROS* 1, 1314–1319 (2009).

[8] J. Jakubiak, K. Tchoń, and M. Janiak, "Motion planning of the trident snake robot: An endogenous configuration space approach", in *Robot Design, Dynamics, and Control*, pp. 159–166, Springer, Wien, 2010.

[9] D. Paszuk, "Motion planning and control algorithms of the trident snake system", *Master's Thesis*, Wrocław University of Technology, Wrocław, 2011, (in Polish).

[10] 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* 26, 1387–1419 (2003).

[11] M. Janiak and K. Tchoń, "Constrained motion planning of nonholomic systems", *Syst. Contr. Lett.* 60 (8), 625–631 (2011).

[12] M. Janiak, "Jacobian inverse kinematics algorithms for mobile manipulators with constraints on state, control, and performance", *Ph.D. Thesis*, Wrocław University of Technology, Wrocław, 2009, (in Polish).

[13] K. Tchoń and J. Jakubiak, "Fourier vs. non-Fourier band-limited Jacobian inverse kinematics for mobile manipulators", *Proc. 2004 MMAR Conf.* 2, 1005–1010 (2004).