Mech. Sci., 4, 153–166, 2013
www.mech-sci.net/4/153/2013/
doi:10.5194/ms-4-153-2013

Mechanical Sciences
Open Access

Multiple-task motion planning of non-holonomic systems with dynamics

A. Ratajczak and K. Tchoń

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

Correspondence to: A. Ratajczak (adam.ratajczak@pwr.wroc.pl)

Received: 14 November 2012 – Revised: 15 March 2013 – Accepted: 5 April 2013 – Published: 15 April 2013

Abstract. This paper addresses the motion planning problem in non-holonomic robotic systems. The system's kinematics and dynamics are represented as a control affine system with outputs. The problem is defined in terms of the end-point map of this system, using the endogenous configuration space approach. Special attention is paid to the multiple-task motion planning problem, i.e. a problem that beyond the proper motion planning task includes a number of additional tasks. For multiple-task motion planning two strategies have been proposed, called the egalitarian approach and the prioritarian approach. Also, two computational strategies have been launched of solving the motion planning problem: the parametric and the non-parametric. The motion planning and computational strategies have been applied to a motion planning problem of the trident snake robot. Performance of the motion planning algorithms is illustrated with computer simulations.

1 Introduction

The motion planning problem of a robotic system consists in determining an action in the configuration space that would drive the system along a desired trajectory or to a desired location in the task space. Frequently, reaching the goal is accompanied with avoiding obstacles and singularities, respecting constraints of motion, limitations of energy, etc. If the planning problem is decomposed into a number of tasks, it is called a multiple-task problem. Resolving multiple-task motion planning problems is enabled by the system's redundancy.

In the area of kinematic control of redundant manipulation robots the multiple-task problems have been addressed within the prioritized approach, initiated by Maciejewski and Klein (1985); Nakamura et al. (1987), and then refined in the works of Chiaverini (1997); Chiacchio et al. (1991); Choi et al. (2004); Antonelli (2009). An extension of these ideas towards mobile robotic systems, in particular mobile robots and mobile manipulators, has been promoted by the endogenous configuration space approach (Tchoń and Jakubiak, 2003; Tchoń and Zadarnowska, 2003). The endogenous configuration space approach employs a control system representation of the robotic system and focuses on the analysis of its end-point map. The central concept of endogenous configuration is identified with the system's control function, so singular controls become singular endogenous configurations, and controllability defines the dexterity of the robotic system. The concept of Jacobian of the robotic system relies on the linear approximation of the control system. Jacobian algorithms are introduced using the continuation method. On the kinematics level the endogenous configuration approach has been developed on the basis of the ideas of Sussmann (1993); Chitour and Sussmann (1998); Divelbiss et al. (1998). Founded on the end-point map of a control system, the endogenous configuration space approach extends in a natural way to robotic systems with dynamics (Zadarnowska and Tchoń, 2007; Ratajczak et al., 2010). Since the endogenous configuration space is infinite-dimensional, the mobile robotic systems have infinite redundancy, capable of accommodating an arbitrary big number of tasks. Using the endogenous configuration space approach, a prioritized approach to motion planning of underactuated robotic systems has been proposed in Ratajczak et al. (2010); Ratajczak (2012).

This paper addresses the multiple-task motion planning problem for control affine systems, that include the dynamics models of non-holonomic robotic systems. It is assumed that the proper motion planning task of reaching a desired point in the task space has been supplemented by a number of additional tasks, characterized by their specific task maps. Two methods of solving the problem have been proposed, called the egalitarian and the prioritarian approach. A motion planning algorithm has been derived by means of the endogenous configuration space approach, in the form of a functional differential equation for the control function. Furthermore, a parametric and a non-parametric strategy of computing numerically the control function have been launched.

Theoretical concepts are applied to the dynamics model of the trident snake robot (Ishikawa, 2004). The robot can be regarded as a realization of the undulatory locomotion principle and a demanding test bed of motion planning algorithms for non-holonomic systems. The design and kinematics analysis of the trident snake with passive wheels and active joints can be found in Ishikawa (2004); Ishikawa et al. (2010). Recently, this analysis has been extended to the case of active wheels in Paszuk et al. (2012) and complemented by a study of trident snake dynamics (Pietrowska, 2012). In both these cases the motion single-task planning algorithms have been derived from the endogenous configuration space approach.

Differently than in the references mentioned above, this paper concentrates on the multiple-task motion planning strategy for the trident snake robot. The robot's kinematics and dynamics are represented as a control affine system with outputs. The problem is defined in terms of the end-point map of this system. The control strategy of the system involves a preliminary state feedback. The motion planning problem includes two subtasks: the proper motion planning task of transferring the system to a desired task space location, and the singularity avoidance task guaranteeing well definiteness of the feedback transformation. Two motion planning strategies have been developed, referred to as the egalitarian and the prioritarian approach. The former strategy regards the component tasks as equivalent, the latter assigns the highest priority to the proper motion planning task. Simultaneously, two computational strategies have been proposed in order to solve the motion planning problem: the parametric and the non-parametric, depending on whether the computation of the control function utilizes a specific base in the endogenous configuration space or is base-independent. Computer simulations demonstrate the performance of the egalitarian non-parametric and the prioritarian parametric motion planning algorithms.

The paper is organized in the following way. Section 2 introduces a control system representation of the non-holonomic system, defines a Jacobian motion planning algorithm, and describes two computational strategies. The strategies of multiple-task motion planning are presented in Sect. 3. In Sect. 4 the motion planning strategies are specified to the trident snake robot. Results of numeric computations are included in Sect. 5. Section 6 concludes the paper. In order to not distract the reader's attention from the main thread of the paper, a derivation of the subtask Jacobian and its inverse as well as the dynamics model of the trident snake are placed in the Appendix.

2 Basic concepts

Since the dynamics of a non-holonomic robotic system can be represented as an affine control system with outputs, this system will define our *universe of discourse*. We begin with introducing basic concepts of the endogenous configuration space approach, and derive the Jacobian motion planning algorithm. This algorithm relies on the solution of a functional differential equation involving a Jacobian inverse operator. Depending on the method of solution of this equation, parametric or non-parametric motion planning algorithms are distinguished.

2.1 Endogenous configuration space approach

The basic concepts of the endogenous configuration space approach will be adopted to a general control affine system, of the form

$$\begin{cases} \dot{x} = f(x) + g(x)u = f(x) + \sum_{i=1}^{m} g_i(x)u_i, \\ y = k(x), \end{cases} \tag{1}$$

where $u \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, $y \in \mathbb{R}^r$. All the functions and vector fields appearing in (1) will be assumed smooth. Let $T > 0$ denote a control time horizon. The admissible control functions entering system (1) will be assumed to belong to the 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

$$\langle u_1(\cdot), u_2(\cdot)\rangle_R = \int_0^T u_1^T(t)R(t)u_2(t)\mathrm{d}t, \tag{2}$$

where $R(t) = R^T(t) > 0$, and the corresponding norm $||u(\cdot)||_R = \langle u(\cdot), u(\cdot)\rangle_R^{1/2}$. For a control function $u(\cdot)$, let $x(t) = \varphi_{x_0,t}(u(\cdot))$ denote the state trajectory of (1), initialized at x_0. It will be assumed that this trajectory exists for every $t \in [0,T]$. The output y is identified with the vector of task variables.

Given an initial state x_0 of system (1) and the time horizon T, a general motion planning problem consists in defining a control $u(t)$ that drives the system's output at T to a prescribed point y_d, so that $y(T) = y_d$.

Our analysis of the motion planning problem will be based on the concept of the end-point map of system (1), defined as the value at T of the output function resulting from the application of a control function $u(\cdot)$,

$$K_{x_0,T}(u(\cdot)) = k(x(T)) = k(\varphi_{x_0,T}(u(\cdot))). \tag{3}$$

For bounded measurable control functions $u(\cdot) \in \mathcal{X}$ the end-point map $K : \mathcal{X} \longrightarrow \mathbb{R}^r$ is continuously differentiable ($\mathbb{C}^1$)

(Sontag, 1990). In the context of mobile robots or mobile manipulators the space $\mathcal{X}$ has been called the endogenous configurations space (Tchoń and Jakubiak, 2003). The derivative of the end-point map is computed by means of the linear approximation to system (1)

$$\dot{\xi}(t) = A(t)\xi(t) + B(t)v(t), \quad \eta(t) = C(t)\xi(t), \quad \xi(0) = 0, \tag{4}$$

along the (control,trajectory) pair $(u(t), x(t))$, where

$$\xi(t) = D\varphi_{x_0,t}(u(\cdot))v(\cdot), \tag{5}$$

and

$$\begin{aligned} A(t) &= \frac{\partial(f(x(t)) + g(x(t))u(t))}{\partial x}, \quad B(t) = g(x(t)), \\ C(t) &= \frac{\partial k(x(t))}{\partial x}. \end{aligned} \tag{6}$$

Given the linear system (4), the derivative of the end-point map at $u(\cdot) \in \mathcal{X}$ is equal to

$$DK_{x_0,T}(u(\cdot))v(\cdot) = \eta(T) = C(T)\xi(T). \tag{7}$$

In compliance with the robotic terminology, the derivative (7) will be called the system's Jacobian,

$$DK_{x_0,T}(u(\cdot))v(\cdot) = J_{x_0,T}(u(\cdot))v(\cdot).$$

It follows that the computation of the Jacobian involves the integration of the differential equation (4) from 0 to T at zero initial condition. If $\Phi(t,s)$ denotes the transition matrix of (4),

$$\frac{\partial\Phi(t,s)}{\partial t} = A(t)\Phi(t,s), \quad \Phi(s,s) = I_n,$$

this means that the Jacobian $J_{x_0,T}(u(\cdot)) : \mathcal{X} \longrightarrow \mathbb{R}^r$ can be expressed as

$$J_{x_0,T}(u(\cdot))v(\cdot) = C(T)\int_0^T \Phi(T,s)B(s)v(s)\mathrm{d}s. \tag{8}$$

The Jacobian allows to distinguish regular and singular controls (endogenous configurations) of system (1). A control $u(\cdot) \in \mathcal{X}$ will be called regular, if the Jacobian is surjective onto $\mathbb{R}^r$, otherwise the control $u(\cdot)$ is referred to as singular. It can be shown that at regular controls the control affine system (1) is locally controllable.

Using the inner product (2) in the endogenous configuration space and the Euclidean structure of the output space, the dual Jacobian map $J^*_{x_0,T}(u(\cdot)) : \mathbb{R}^r \longrightarrow \mathcal{X}$ can be defined in the following way

$$\left(J^*_{x_0,T}(u(\cdot))\eta\right)(t) = R^{-1}(t)B^T(t)\Phi^T(T,t)C^T(T)\eta. \tag{9}$$

2.2 Jacobian motion planning

Using the end-point map, the general motion planning problem in system (1) is tantamount to computing a control function $u_d(\cdot)$, such that $K_{x_0,T}(u_d(\cdot)) = y_d$. The problem can be solved by means of a Jacobian motion planning algorithm whose derivation relies on the continuation (homotopy) method (Sussmann, 1993). Given the motion planning problem, we begin with any initial control $u_0(\cdot) \in \mathcal{X}$. If the initial choice does not solve the problem, i.e. $K_{x_0,T}(u_0(\cdot)) \neq y_d$, we choose in $\mathcal{X}$ a differentiable curve $u_\theta(\cdot)$, $\theta \in \mathbb{R}$, passing at $\theta = 0$ through $u_0(\cdot)$, and compute the task space error along this curve

$$e(\theta) = K_{x_0,T}(u_\theta(\cdot)) - y_d. \tag{10}$$

Next, we request that the error decrease exponentially along with θ, with a prescribed decay rate $\gamma > 0$, i.e.

$$\frac{\mathrm{d}e(\theta)}{\mathrm{d}\theta} = -\gamma e(\theta). \tag{11}$$

By differentiating the formula (10) with respect to θ, we arrive at the Ważewski-Davidenko equation

$$J_{x_0,T}(u_\theta(\cdot))\frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -\gamma e(\theta), \tag{12}$$

involving the Jacobian (8). If $J^{\#}_{x_0,T}(u(\cdot)) : \mathbb{R}^r \longrightarrow \mathcal{X}$ denotes a right Jacobian inverse, such that $J_{x_0,T}(u(\cdot))J^{\#}_{x_0,T}(u(\cdot)) = I_r$, then (12) can be converted into a dynamic system

$$\frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -\gamma J^{\#}_{x_0,T}(u_\theta(\cdot))e(\theta) \tag{13}$$

evolving in $\mathcal{X}$. Finally, a solution of the motion planning problem is obtained as the limit

$$u_d(t) = \lim_{\theta\to+\infty} u_\theta(t).$$

A frequently used right Jacobian inverse is the Moore–Penrose inverse derived from minimizing the square norm $||v(\cdot)||^2_R$ under the equality constraint (a Jacobian equation)

$$J_{x_0,T}(u(\cdot))v(\cdot) = \eta, \tag{14}$$

$\eta \in \mathbb{R}^r$. The resulting formula is (Tchoń and Jakubiak, 2003),

$$\left(J^{\#}_{x_0,T}(u(\cdot))\eta\right)(t)=R^{-1}(t)B^T(t)\Phi^T(T,t)C^T(T)\mathcal{G}^{-1}_{x_0,T}(u(\cdot))\eta, \tag{15}$$

where

$$\mathcal{G}_{x_0,T}(u(\cdot))=C(T)\int_0^T \Phi(T,s)B(s)R^{-1}(s)B^T(s)\Phi^T(T,s)\mathrm{d}sC^T(T).$$

denotes the Gram matrix associated with the linear system (4), that in robotics context is referred to as the dexterity or mobility matrix of system (1) (Tchoń and Zadarnowska, 2003). At regular control functions the Gram matrix is full rank, making the inverse (15) well defined. The Gram matrix can be conveniently computed by integrating the Lyapunov differential equation

$$\dot{M}(t) = B(t)R^{-1}(t)B^T(t) + A(t)M(t) + M(t)A^T(t), \tag{16}$$

at zero initial condition $M(0) = 0$, and then substituting $\mathcal{G}_{x_0,T}(u(\cdot)) = C(T)M(T)C^T(T)$. By invoking the definition of dual Jacobian (9), it is easily checked that the Gram matrix equals the composition

$$\mathcal{G}_{x_0,T}(u(\cdot)) = J_{x_0,T}(u(\cdot))J^*_{x_0,T}(u(\cdot)).$$

Moreover, it follows that the Jacobian inverse (15) can be written as

$$\left(J^\#_{x_0,T}(u(\cdot))\eta\right)(t) = \left(J^*_{x_0,T}(u(\cdot))\mathcal{G}^{-1}_{x_0,T}(u(\cdot))\eta\right)(t),$$

what justifies the identity

$$J^\#_{x_0,T}(u(\cdot)) = J^*_{x_0,T}(u(\cdot))\mathcal{G}^{-1}_{x_0,T}(u(\cdot)). \tag{17}$$

2.3 Computations

The Jacobian motion planning algorithm derived in the previous subsection exploits a solution of a functional differential equation (13) in the endogenous configuration space $\mathcal{X}$. In order to make this equation tractable, we need to discretize the θ variable, and pass to a discrete control updating scheme

$$u_{\theta+1}(t) = \\ u_\theta(t) - \gamma R^{-1}(t)B^T_\theta(t)\Phi^T_\theta(T,t)C^T_\theta(T)\mathcal{G}^{-1}_{x_0,T}(u_\theta(\cdot))e(\theta), \tag{18}$$

where $B_\theta(t)$, $\Phi_\theta(t,s)$, $C_\theta(t)$ are computed along $(u_\theta(t), x_\theta(t))$ in agreement with (6), and $e(\theta) = K_{x_0,T}(u_\theta(\cdot)) - y_d$, for $\theta = 0, 1, \ldots$.

Given $u_\theta(t)$, a basic step of the updating (18) consists in solving simultaneously a system of differential equations

$$\begin{cases} \dot{x}_\theta(t) = f(x_\theta(t)) + g(x_\theta(t))u_\theta(t), \\ \frac{\partial\Phi_\theta(T,t)}{\partial t} = -\Phi_\theta(T,t)A_\theta(t), \\ \dot{M}_\theta(t) = B_\theta(t)R^{-1}(t)B^T_\theta(t) + A_\theta(t)M_\theta(t) + M_\theta(t)A^T_\theta(t) \end{cases}$$

with respective boundary conditions $x_\theta(0) = x_0$, $\Phi_\theta(T,T) = I_n$ and $M_\theta(0) = 0$.

Alternatively, we can expand the control functions into a truncated orthogonal series that leads to a finite parametrization of controls. More specifically, we select a finite base of orthogonal functions $\{\varphi_0(t), \varphi_1(t), \ldots, \varphi_k(t)\}$, and assume that

$$u_\lambda(t) = \Psi(t)\lambda, \quad u_{\lambda i}(t) = \sum_{j=0}^{k} \varphi_j(t)\lambda_{ij}, \;\; i = 1, 2, \ldots, m,$$

where $\lambda \in \mathbb{R}^s$, $s = m(k+1)$, and matrix $\Psi(t)$ of dimension $m \times s$ contains the suitably arranged basic functions. By orthogonality of the base $\int_0^T \Psi^T(t)\Psi(t)\mathrm{d}t = I_s$. Under assumption that $v(t) = \Psi(t)\mu$, the Jacobian (8) takes the matrix form

$$\hat{J}_{x_0,T}(\lambda) = C_\lambda(T)\int_0^T \Phi_\lambda(T,s)B_\lambda(s)\Psi(s)\mathrm{d}s,$$

where the matrices (6) need to be computed for $u(t) = u_\lambda(t)$ and $x(t) = \varphi_{x_0,t}(u_\lambda(\cdot))$. It follows that this Jacobian can be obtained by the integration of the matrix differential equation

$$\dot{N}_\lambda(t) = B_\lambda(t)\Psi(t) + A_\lambda(t)N_\lambda(t)$$

with initial condition $N_\lambda(0) = 0$, followed by the substitution $\hat{J}_{x_0,T}(\lambda) = C_\lambda(T)N_\lambda(T)$. For the matrix Jacobian, the Jacobian equation (14) assumes the form $\hat{J}_{x_0,T}(\lambda)\mu = \eta$. It can be shown that the Moore–Penrose Jacobian inverse assumes the matrix form

$$\hat{J}^\#_{x_0,T}(\lambda) = S^{-1}\hat{J}_{x_0,T}(\lambda)\left(\hat{J}_{x_0,T}(\lambda)S^{-1}\hat{J}^T_{x_0,T}(\lambda)\right)^{-1}, \tag{19}$$

where $S = \int_0^T \Psi^T(t)R(t)\Psi(t)\mathrm{d}t$. Consequently, the updating scheme of the control function (18) converts to

$$\lambda_{\theta+1} = \lambda_\theta - \gamma\hat{J}^\#_{x_0,T}(\lambda_\theta)e(\theta). \tag{20}$$

The representation of the control functions by means of a truncated orthogonal series will be called parametric. By contrast, the representation that does not use any base will be referred to as non-parametric. This terminology extends to the corresponding motion planning algorithms.

3 Multiple-task motion planning

If the motion planning problem consists only of the motion planning task, it will be called single-task. When the motion planning task is augmented with additional tasks, the motion planning problem will be referred to as a multiple-task problem. We shall assume that the additional tasks are defined by task maps ${}^iK_{x_0,T} : \mathcal{X} \longrightarrow R \; i = 1, 2, \ldots, p$, operating in the endogenous configuration space of (1). Most frequently, these task maps have the form

$${}^iK_{x_0,T}(u(\cdot)) = \int_0^T F_i(x(t), u(t))\mathrm{d}t, \tag{21}$$

for a non-negative function $F_i(x,u) \geq 0$ that is assumed differentiable wherever defined. Given p additional tasks, the multiple-task motion planning problem consists in determining a control function $u(\cdot) \in \mathcal{X}$ minimizing the collective error

$$\mathbf{e}(\theta) = \left({}^0e(\theta), {}^1e(\theta), \ldots, {}^pe(\theta)\right), \tag{22}$$

where ${}^{0}e(\theta) = K_{x_0,T}(u_\theta(\cdot)) - y_d$ and ${}^{i}e(\theta) = {}^{i}K_{x_0,T}(u_\theta(\cdot))$ $i = 1,2,\ldots,p$ are errors corresponding to subsequent subtasks. We associate with the task maps the Jacobians: the Jacobian for the task number 0 is given by (8), while for the i-th subtask the Jacobian

$$ {}^{i}J_{x_0,T}(u(\cdot))v(\cdot) = \int_0^T \left(\frac{\partial F_i(x(t),u(t))}{\partial x}\xi(t) + \frac{\partial F_i(x(t),u(t))}{\partial u}v(t) \right) \mathrm{d}t, \tag{23} $$

where $\xi(t)$ denotes the solution of (4), see Appendix for a derivation. The Moore–Penrose inverse for ${}^{0}J_{x_0,T}(u(\cdot))$ is defined by expression (15). In Appendix we have shown that the Moore–Penrose inverse of the ith Jacobian, $i = 1,2,\ldots,p$, takes the following form

$$ \left({}^{i}J^{\#}_{x_0,T}(u(\cdot))\eta \right)(t) = \frac{b_i(t)+c_i(t)}{||b_i(\cdot)+c_i(\cdot)||^2_R}\eta, \tag{24} $$

$\eta \in \mathbb{R}$, where

$$ b_i(t) = R^{-1}(t)B^T(t)\int_t^T \Phi^T(s,t)\left(\frac{\partial F_i(x(s),u(s))}{\partial x}\right)^T \mathrm{d}s, \tag{25} $$

$$ c_i(t) = R^{-1}(t)\left(\frac{\partial F_i(x(t),u(t))}{\partial u}\right)^T. \tag{26} $$

The task Jacobians can be arranged into a collective Jacobian

$$ \mathbf{J}_{x_0,T}(u_\theta(\cdot))v(\cdot) = \left({}^{0}J_{x_0,T}(u(\cdot))v(\cdot), {}^{1}J_{x_0,T}(u(\cdot))v(\cdot), \ldots, {}^{p}J_{x_0,T}(u(\cdot))v(\cdot) \right). \tag{27} $$

These subtasks can either be treated as equivalent and solved simultaneously or ordered according to their importance and solved sequentially. These two approaches will be further referred to as egalitarian and prioritarian. In the following two subsections we shall derive an egalitarian and a prioritarian Jacobian motion planning algorithm.

3.1 Egalitarian approach

In accordance with the egalitarian approach all subtasks need to be solved simultaneously. Imposing the exponential decrease of the error (22) with a decay rate $\gamma > 0$, and using the collective Jacobian (27), we arrive at the Ważewski-Davidenko equation

$$ \mathbf{J}_{x_0,T}(u_\theta(\cdot))\frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -\gamma\mathbf{e}(\theta). \tag{28} $$

Finally, similarly to (13), the equation (28) can be solved by means of the Moore–Penrose Jacobian inverse of the collective Jacobian,

$$ \frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -\gamma\mathbf{J}^{\#}_{x_0,T}(u_\theta(\cdot))\mathbf{e}(\theta). \tag{29} $$

The egalitarian solution of the multiple-task motion planning problem is obtained as the limit $u_d = \lim_{\theta\to+\infty} u_\theta(t)$ of the trajectory of dynamic system (29).

3.2 Prioritarian approach

Differently to the egalitarian algorithm, now the subtasks will be ordered with decreasing priorities. The essential assumption of the prioritarian approach is that the solution of a lower priority task is sought in a space rendered accessible by the higher priority tasks, so it should not affect the solution of any task with higher priority. Given the i-th error (22) and the i-th Jacobian (27), we start the derivation from a statement of the Jacobian equation for the i-th subtask,

$$ \frac{\mathrm{d}\,{}^{i}e(\theta)}{\mathrm{d}\theta} = {}^{i}J_{x_0,T}(u_\theta(\cdot))\frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -{}^{i}\gamma\,{}^{i}e(\theta), \tag{30} $$

with ${}^{i}\gamma > 0$ and $i = 0,1,\ldots,p$. A general solution of this equation involves the Jacobian inverse with projection (Tchoń and Jakubiak, 2003),

$$ \frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -{}^{i}\gamma\,{}^{i}J^{\#}_{x_0,T}(u_\theta(\cdot))\,{}^{i}e(\theta) + {}^{i}P_{x_0,T}(u_\theta(\cdot))\,{}^{i}\zeta_\theta(\cdot), \tag{31} $$

where

$$ {}^{i}P_{x_0,T}(u_\theta(\cdot)) = \mathrm{id}_{\mathcal{X}} - {}^{i}J^{\#}_{x_0,T}(u_\theta(\cdot))\,{}^{i}J_{x_0,T}(u_\theta(\cdot)) $$

denotes the projection onto $\ker {}^{i}J_{x_0,T}(u_\theta(\cdot))$, and ${}^{i}\zeta_\theta(\cdot) \in \mathcal{X}$ is any element of the endogenous configuration space. For the highest priority task (number 0) the equality (31) is required to hold in the whole endogenous configuration space,

$$ \frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -{}^{0}\gamma\,{}^{0}J^{\#}_{x_0,T}(u_\theta(\cdot))\,{}^{0}e(\theta) + {}^{0}P_{x_0,T}(u_\theta(\cdot))\,{}^{0}\zeta_\theta(\cdot). \tag{32} $$

For the lower priority task (number 1) this equality needs to be satisfied only within the kernel of ${}^{0}J^{\#}_{x_0,T}(u_\theta(\cdot))$, i.e.

$$ {}^{0}P_{x_0,T}(u_\theta(\cdot))\frac{du_\theta(\cdot)}{d\theta} = -{}^{1}\gamma{}^{0}P_{x_0,T}(u_\theta(\cdot))\,{}^{1}J^{\#}_{x_0,T}(u_\theta(\cdot))\,{}^{1}e(\theta) + {}^{0}P_{x_0,T}(u_\theta(\cdot)){}^{1}P_{x_0,T}(u_\theta(\cdot))\,{}^{1}\zeta_\theta(\cdot). \tag{33} $$

Having projected (32) onto $\ker {}^{0}J_{x_0,T}(u_\theta(\cdot))$, we get

$$ {}^{0}P_{x_0,T}(u_\theta(\cdot))\frac{du_\theta(\cdot)}{d\theta} = {}^{0}P_{x_0,T}(u_\theta(\cdot))\,{}^{0}\zeta_\theta(\cdot), \tag{34} $$

where we have used the identity ${}^{i}P_{x_0,T}(u_\theta(\cdot))\,{}^{i}J^{\#}_{x_0,T}(u_\theta(\cdot)) = 0$, and the idempotency of the projection. Now, a combination of (33) and (34) results in

$$ {}^{0}P_{x_0,T}(u_\theta(\cdot))\,{}^{0}\zeta_\theta(\cdot) = -{}^{1}\gamma{}^{0}P_{x_0,T}(u_\theta(\cdot))\,{}^{1}J^{\#}_{x_0,T}(u_\theta(\cdot))\,{}^{1}e(\theta) + {}^{0}P_{x_0,T}(u_\theta(\cdot)){}^{1}P_{x_0,T}(u_\theta(\cdot))\,{}^{1}\zeta_\theta(\cdot). $$

Finally, a substitution of the above identity into (32) yields the following prioritarian Jacobian motion planning algorithm for two subtasks

$$ \frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -{}^{0}\gamma\,{}^{0}J^{\#}_{x_0,T}(u_\theta(\cdot))\,{}^{0}e(\theta) - {}^{1}\gamma\,{}^{0}P_{x_0,T}(u_\theta(\cdot))\,{}^{1}J^{\#}_{x_0,T}(u_\theta(\cdot))\,{}^{1}e(\theta) + {}^{0}P_{x_0,T}(u_\theta(\cdot)){}^{1}P_{x_0,T}(u_\theta(\cdot))\,{}^{1}\zeta_\theta(\cdot) \tag{35} $$

If there are only two subtasks, the last term ${}^{1}\zeta_{\theta}(\cdot)$ is zero, otherwise it could be used to define the prioritarian algorithm for the remaining lower priority tasks.

In this way the presented derivation extends to $p+1$ subtasks, resulting in the following motion planning algorithm (Ratajczak et al., 2010; Ratajczak, 2012),

$$\frac{\mathrm{d}u_{\theta}(t)}{\mathrm{d}\theta} = -\sum_{i=0}^{p} {}^{i}\gamma \left(\prod_{j=0}^{i} {}^{j-1}P_{x_0,T}(u_{\theta}(\cdot)) \right) {}^{i}J^{\#}_{x_0,T}(u_{\theta}(\cdot))\, {}^{i}e(\theta), \qquad (36)$$

where ${}^{-1}P_{x_0,T}(u_{\theta}(\cdot)) = \mathrm{id}_{\mathcal{X}}$. In the terminology of Antonelli (2009) this algorithm belongs to the successive inverse-based projection methods. Again, the solution of the multiple-task motion planning problem is obtained as the limit $u_d(t) = \lim_{\theta\to+\infty} u_{\theta}(t)$ of the trajectory of system (36).

Let us take a closer examination of the control formula (35). By a substitution of this formula into (32), it is easily seen that the first error

$$\frac{\mathrm{d}^{0}e(\theta)}{\mathrm{d}\theta} = -{}^{0}\gamma\, {}^{0}e(\theta)$$

decreases exponentially, as requested in (11). However, because of the task-priority assumption, the second error

$$\frac{\mathrm{d}^{1}e(\theta)}{\mathrm{d}\theta} = -{}^{1}\gamma\, {}^{1}e(\theta) - {}^{0}\gamma {}^{1}J_{x_0,T}(u_{\theta}(\cdot))\, {}^{0}J^{\#}_{x_0,T}(u_{\theta}(\cdot))\, {}^{0}e(\theta)$$
$$+ {}^{1}\gamma {}^{1}J_{x_0,T}(u_{\theta}(\cdot))\, {}^{0}J^{\#}_{x_0,T}(u_{\theta}(\cdot))\, {}^{0}J_{x_0,T}(u_{\theta}(\cdot))\, {}^{1}J^{\#}_{x_0,T}(u_{\theta}(\cdot))\, {}^{1}e(\theta),$$

behaves differently. The error ${}^{1}e(\theta)$ will converge toward zero exponentially, provided that it does not affect convergence of the error ${}^{0}e(\theta)$. This happens when ${}^{1}J_{x_0,T}(u_{\theta}(\cdot))\, {}^{0}J^{\#}_{x_0,T}(u_{\theta}(\cdot)) = 0$. Taking into account the form of (17), the last equality will be satisfied whenever

$${}^{1}J_{x_0,T}(u_{\theta}(\cdot))\, {}^{0}J^{*}_{x_0,T}(u_{\theta}(\cdot)) = 0,$$

what may be interpreted as a sort of orthogonality of Jacobians ${}^{0}J_{x_0,T}(u(\cdot))$ and ${}^{1}J_{x_0,T}(u(\cdot))$.

4 Motion planning of trident snake

The presented algorithms will be utilized in order to solve a motion planning problem for the trident snake mobile robot (Ishikawa, 2004; Ishikawa et al., 2010; Paszuk et al., 2012). In the following subsections we are going to present a control-theoretic model, state a multiple-task motion planning problem, and introduce the egalitarian and the prioritarian motion planning algorithm for the trident snake robot.

4.1 Model

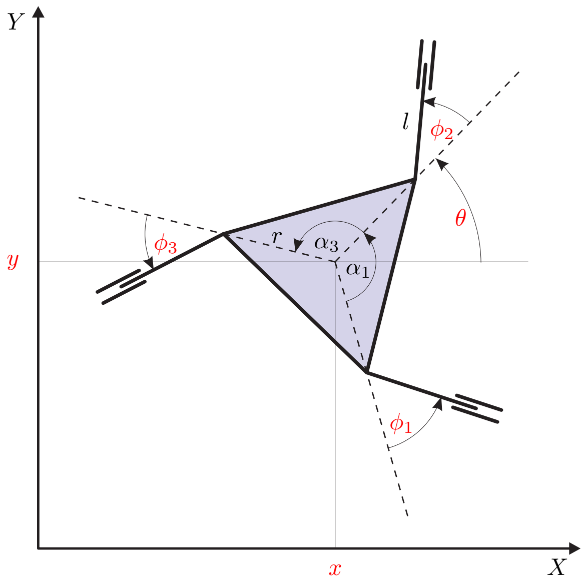

Figure 1. Trident snake robot: conceptual model.

The trident snake is a non-holonomic mobile robot. It consists of a triangular-shape body and three links able to rotate around the attachment points (see Fig. 1). Each link is ended with a passive wheel which is subjected to non-holonomic constraints. The control inputs of the robot are the driving torques exerted at the joints between the body and the links. The angles $\alpha_1 = -\frac{2}{3}\pi$, $\alpha_2 = 0$, $\alpha_3 = \frac{2}{3}\pi$ are constant geometric parameters of the robot, l denotes the length of each link, and r stands for the distance between the center of the robot body and the joints.

4.1.1 Kinematics

According to Fig. 1, the vector of generalized coordinates is defined as $q = (x, y, \theta, \phi_1, \phi_2, \phi_3) = (x, y, \theta, \phi) \in \mathbb{R}^6$. For every wheel the non-holonomic constraints arise from the assumption of non-slipping laterally. The corresponding constraints matrix takes the form (Ishikawa et al., 2010; Paszuk et al., 2012)

$$\mathsf{A}(q) = \begin{bmatrix} \mathsf{A}_1(q)\mathrm{Rot}^T(Z,\theta) & -lI_3 \end{bmatrix},$$

where $\mathrm{Rot}(Z,\theta)$ is the rotation matrix around Z-axis by θ angle, and $\mathsf{A}_1(q)$ is defined as

$$\mathsf{A}_1(q) = \begin{bmatrix} \sin(\alpha_1+\phi_1) & -\cos(\alpha_1+\phi_1) & -l - r\cos\phi_1 \\ \sin(\alpha_2+\phi_2) & -\cos(\alpha_2+\phi_2) & -l - r\cos\phi_2 \\ \sin(\alpha_3+\phi_3) & -\cos(\alpha_3+\phi_3) & -l - r\cos\phi_3 \end{bmatrix}.$$

Using the relation $\mathsf{A}(q)\mathsf{G}(q) = 0$, we find

$$\mathsf{G}(q) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \\ \frac{\sin(\alpha_1+\phi_1)}{l} & -\frac{\cos(\alpha_1+\phi_1)}{l} & -1 - \frac{r\cos\phi_1}{l} \\ \frac{\sin(\alpha_2+\phi_2)}{l} & -\frac{\cos(\alpha_2+\phi_2)}{l} & -1 - \frac{r\cos\phi_2}{l} \\ \frac{\sin(\alpha_3+\phi_3)}{l} & -\frac{\cos(\alpha_3+\phi_3)}{l} & -1 - \frac{r\cos\phi_3}{l} \end{bmatrix} = \begin{bmatrix} \mathsf{G}_1(\theta) \\ \mathsf{G}_2(\phi) \end{bmatrix}, \qquad (37)$$

so the trident snake kinematics can be represented by the driftless control system

$$\dot{q} = \mathsf{G}(q)v,$$

$v \in \mathbb{R}^3$ denoting the control variable.

4.1.2 Dynamics

The dynamics equations of the trident snake can be obtained from the Lagrange formalism together with the d'Alembert principle. A detailed derivation can be found in Pietrowska (2012). After the elimination of the traction forces, the resulting equations of motion take the form

$$\begin{cases} \dot{q} = \mathsf{G}(q)v \\ \dot{v} = \mathsf{N}(q,v) + \mathsf{P}(q)w, \\ y = k(q,v). \end{cases} \tag{38}$$

The matrix $\mathsf{G}(q)$ is given by (37), the terms $\mathsf{N}(q,v)$ and $\mathsf{P}(q)$ can be reconstructed from the data provided in the Sect. A2. It follows that the trident snake has $\dim v = \dim w = 3$, so it is fully controlled. Therefore, it is possible to apply a partially linearizing feedback transformation

$$w = \mathsf{P}^{-1}(q)\,(u - \mathsf{N}(q,v))\,, \tag{39}$$

u denoting a new control variable, that simplifies the control system representation substantially, and produces a kinematically reduced system (Lewis, 1999),

$$\begin{cases} \dot{q} = \mathsf{G}(q)v \\ \dot{v} = u, \\ y = k(q,v). \end{cases} \tag{40}$$

In Paszuk et al. (2012), it has been shown that the feedback (39) is well defined provided that the lower submatrix of (37) is non-singular, i.e. $\det \mathsf{G}_2(\phi) \neq 0$.

To obtain a control affine system (1), we need to introduce a state vector $x = (q,v) \in \mathbb{R}^9$. Then, the kinematics and dynamics of the trident snake may be rewritten as

$$\begin{cases} \dot{x} = f(x) + g(x)u, \\ y = k(x), \end{cases} \tag{41}$$

where $u \in \mathbb{R}^3$ and

$$f(x) = \begin{pmatrix} \mathsf{G}(q)v \\ 0_{3\times 1} \end{pmatrix}, \quad g(x) = \begin{bmatrix} 0_{3\times 3} \\ I_3 \end{bmatrix}.$$

Assuming that all state variables x are subject to motion planning, we get the output function $y = k(x) = x$.

4.2 Motion planning

The motion planning problem for the trident snake will consist in defining a control function $u(t)$ in system (41) that drives the system from an initial state to a desired terminal output, over a prescribed time horizon $T > 0$. In order to make the feedback (39) well defined, the control $u(t)$ should also prevent matrix $\mathsf{G}_2(\phi)$ from getting singular. Clearly, the motion planning problem includes two subtasks, of which the former will be called the proper motion planning task (index 0), whereas the latter will be named the singularity avoidance task (index 1). The task map ${}^0K_{x_0,T}(u(\cdot))$ is just the end-point map of system (41). The singularity avoidance task will be assigned the task map ${}^1K_{x_0,T}(u(\cdot))$ of the form (21) with function $F_1(x,u) = \det^{-2}(\mathsf{G}_2(\phi))$. The collective error $\mathbf{e}(\theta)$ and the collective Jacobian $\mathbf{J}_{x_0,T}(u(\cdot))$ can be derived directly from (22) and (27).

4.3 Algorithms

The motion planning problem for the trident snake robot will be solved by means of the multiple-task motion planning algorithms introduced in Sect. 3. For the egalitarian motion planning the collective error is defined as

$$\mathbf{e}(\theta) = ({}^0e(\theta), \delta\, {}^1e(\theta)),$$

where the second component has been multiplied by a scaling factor δ in order to maintain a dynamic balance of the two subtasks. The egalitarian motion planning algorithm takes the form (29). The solution is the limit $u_d = \lim_{\theta\to+\infty} u_\theta(t)$ of the trajectory of (29).

In the prioritarian approach, the proper motion planning subtask will be assigned a higher priority, while the singularity avoidance subtask will get a lower priority. Having defined two errors: ${}^0e(\theta)$, ${}^1e(\theta)$ and two Jacobian inverses: ${}^0J^{\#}_{x_0,T}(u(\cdot))$ and ${}^1J^{\#}_{x_0,T}(u(\cdot))$ given, respectively, by (15) and (24), we can provide the following formula for the prioritarian motion planning algorithm (36)

$$\frac{\mathrm{d}u_\theta(\cdot)}{\mathrm{d}\theta} = -{}^0\gamma\, {}^0J^{\#}_{x_0,T}(u_\theta(\cdot))\, {}^0e(\theta) - {}^1\gamma\, {}^0P_{x_0,T}(u_\theta(\cdot))\, {}^1J^{\#}_{x_0,T}(u_\theta(\cdot))\, {}^1e(\theta),$$

where ${}^0P_{x_0,T}(u(\cdot))$ denotes the projection into $\ker {}^0J_{x_0,T}(u(\cdot))$. In the next section the performance of both these algorithms will be illustrated by means of numerical computations.

5 Implementation and computations

Potentially, in the egalitarian as well as in the prioritarian motion planning algorithm, both the parametric and nonparametric computational approaches introduced in Sect. 2.3 are applicable. In order to demonstrate this possibility, two numerical solutions of the same motion planning problem for the trident snake will be presented. The problem involves the proper motion planning task and the singularity avoidance task. In the first case, the solution will be delivered by the non-parametric egalitarian algorithm. In the second case, the parametric prioritarian algorithm is used. The

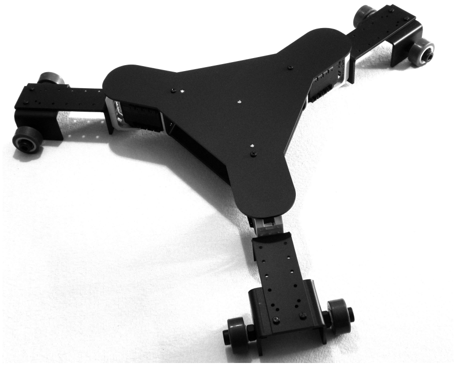

Figure 2. Trident snake robot: physical model.

proper motion planning task consists in driving the system (41) from the initial state $x_0 = (0,0,0,0,0,0,0,0,0)$ to the desired $y_d = x_d = (0.1,0,0,0,0,0,0,0,0)$, over the time horizon $T = 1$. Taking into account geometric parameters of the trident snake $r = l = 0.12$, this task could be interpreted as an elementary a rest-to-rest move forward. For both these simulations, the motion planning problem is regarded as solved, when the norm $\|{}^0e(\theta)\|$ drops below 10^{-4}, and the determinant $\det(\mathsf{G}_2(\phi)) \neq 0$ during the whole motion time. The initial control functions for both cases are chosen constant, $u_0 = (2,1,-1)$. In the egalitarian algorithm the error decay rate $\gamma = 0.1$, and the scaling factor $\delta = 10^{-4}$. The prioritarian algorithm uses decay coefficients ${}^0\gamma = 0.5$ and ${}^1\gamma = 10^{-3}$. In the parametric representation a truncated Fourier series of length $k = 20$ has been chosen, so the vector of control parameters $\lambda \in \mathbb{R}^{63}$. The dynamics model of the trident snake robot has been borrowed from Pietrowska (2012). The model corresponds to the trident snake robot designed at our laboratory, Gospodarek (2011), displayed in Fig. 2. Its dynamic parameters are the following:

– body mass $m_0 = 0.52$kg,

– wheel mass $m_\mathrm{w} = 0.03$kg,

– wheel radius $r_\mathrm{w} = 0.02$m,

– wheel thickness $d = 0.001$m,

– moment of inertia of the body around its center of mass $I_0 = \frac{m_0 r^2}{4}$,

– moment of inertia of the wheel around the center of mass of the robot body $I_{0\mathrm{w}} = m_\mathrm{w}(\frac{r_\mathrm{w}^2}{4} + \frac{d^2}{12})$,

– link mass $m_\mathrm{l} = 0.07$kg,

– motor mass $m_\mathrm{m} = 0.055$kg,

– total mass $m_c = m_0 + 3(m_\mathrm{w} + m_\mathrm{m} + m_\mathrm{l})$.

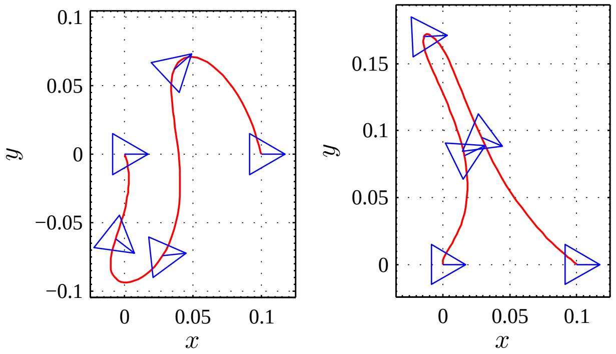

Figure 3. Motion path in XY plane: egalitarian solution (left) and prioritarian solution (right).

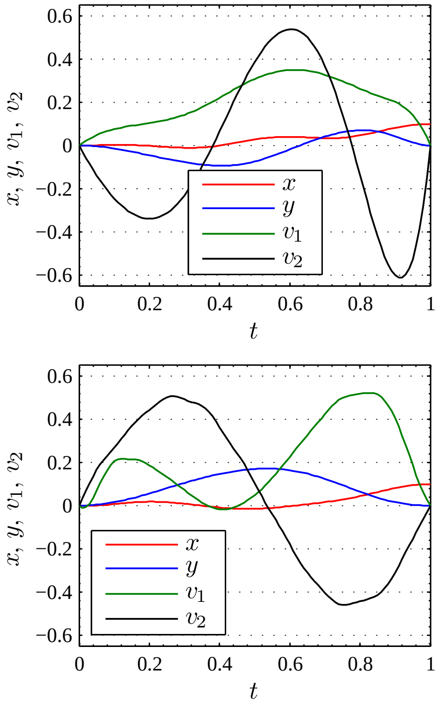

Figure 4. Position and linear velocity: egalitarian solution (top) and prioritarian solution (bottom).

The results of computations are depicted in Figs. 3–11. SI units of measure have been employed. The motion paths corresponding to the solution $u_d(t)$ of the motion planning problem provided by the algorithms are presented in Fig. 3, that also displays the robot body every 0.2 s. As it can be seen, these paths are different for the two algorithms. Figures 4–6 show the state space trajectories. One can observe that the proper motion planning task has been solved correctly. The robot position and orientation (x,y,θ) as well as the joint variables ϕ have reached the desired values at T. Also, all velocities v have become equal to zero. Figure 7 presents the control $u(t)$ in the linearized model. Comparing the control

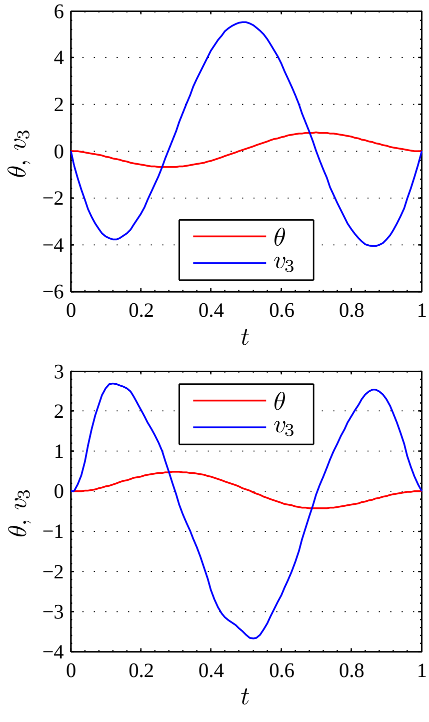

Figure 5. Orientation and angular velocity: egalitarian solution (top) and prioritarian solution (bottom).

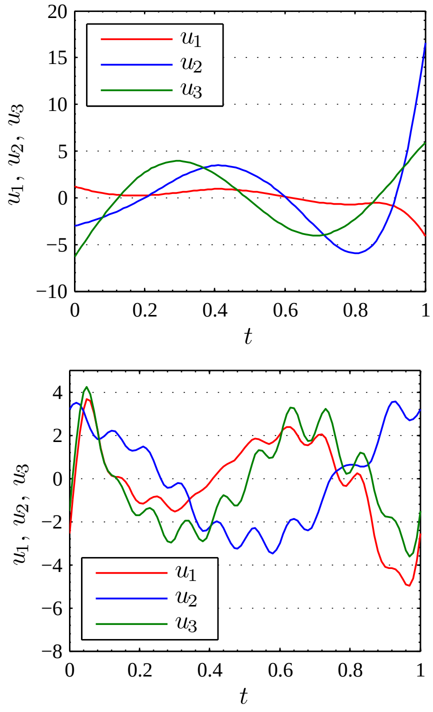

Figure 7. Linearized control $u(t)$, egalitarian solution (top) and prioritarian solution (bottom).

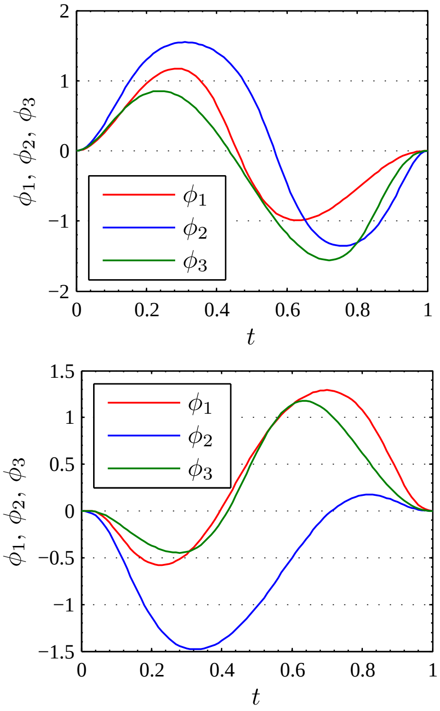

Figure 6. Joint angles: egalitarian solution (top) and prioritarian solution (bottom).

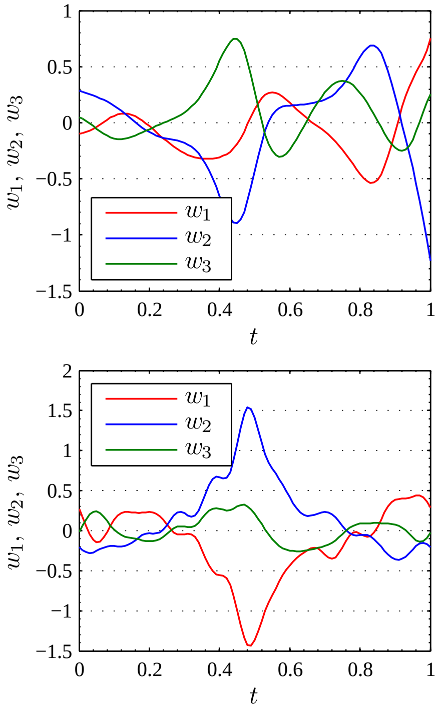

Figure 8. Original control $w(t)$: egalitarian solution (top) and prioritarian solution (bottom).

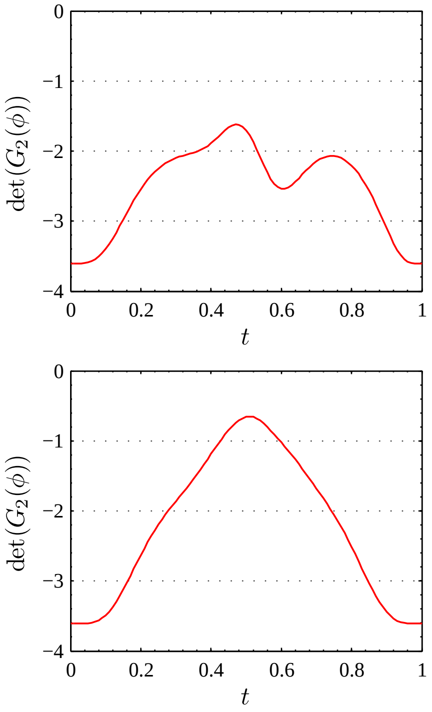

Figure 9. Determinant $\det(G_2(\phi))$: egalitarian solution (top) and prioritarian solution (bottom).

functions obtained from the non-parametric egalitarian and the parametric prioritarian approaches, one can see that the non-parametric controls are smoother. The original control $w(t)$ representing the torques in the joints is depicted in Fig. 8. Again, as in the case of $u(t)$ control, the function $w(t)$ produced by the non-parametric algorithm is smoother. Figure 9 refers to the singularity avoidance subtask. The plots present the value of the determinant $\det(\mathsf{G}_2(\phi))$ during the motion time. In both algorithms the determinant stays quite far away from zero, what means that the egalitarian and the prioritarian algorithms have solved the second task in a satisfactory way. This conclusion is confirmed by Fig. 10 showing three-dimensional plots in the joint space ϕ. The surface $\det(\mathsf{G}_2(\phi)) = 0$ represents singularities. Joint trajectories $\phi(t)$ (starting with the black color and going towards the light green) remain safely inside the regular set. Finally, the last Fig. 11 illustrates the convergence of the algorithms. It follows that the non-parametric egalitarian algorithm needs more steps in order to fulfill the stop condition $||^0e(\theta)|| < 10^{-4}$. In the egalitarian algorithm the total error $\mathbf{e}(\theta)$ decreases exponentially. In the case of the prioritarian algorithm the error of the second task $|^1e(\theta)|$ has been forced to increase by the higher priority task.

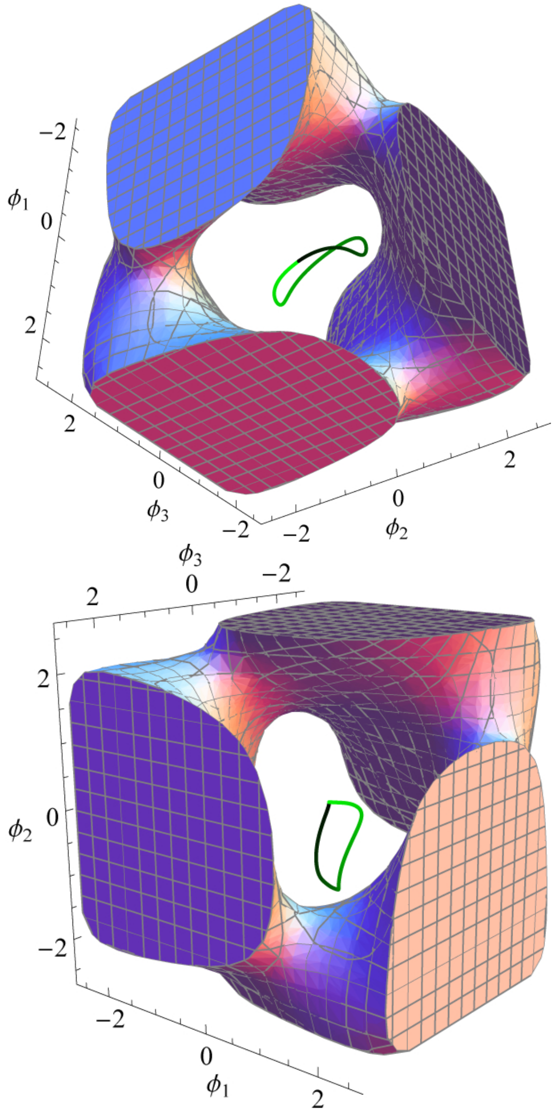

Figure 10. Motion in joint space ϕ: egalitarian solution (top) and prioritarian solution (bottom).

6 Conclusions

We have designed two multiple-task motion planning algorithms for non-holonomic systems with dynamics. Conceptually, their derivation is rooted in the endogenous configuration space approach, so the solution of the problem is a control function in the control system representation of the dynamics. The egalitarian algorithm treats all the component tasks as equivalent. The prioritarian algorithm arranges the tasks according to decreasing priorities. Computations of the control function resulting from these algorithms can either use a finite-dimensional expansion of control functions at a prescribed base in the endogenous configuration space or be base-independent. This gives rise to either the parametric or the non-parametric computational technique. The

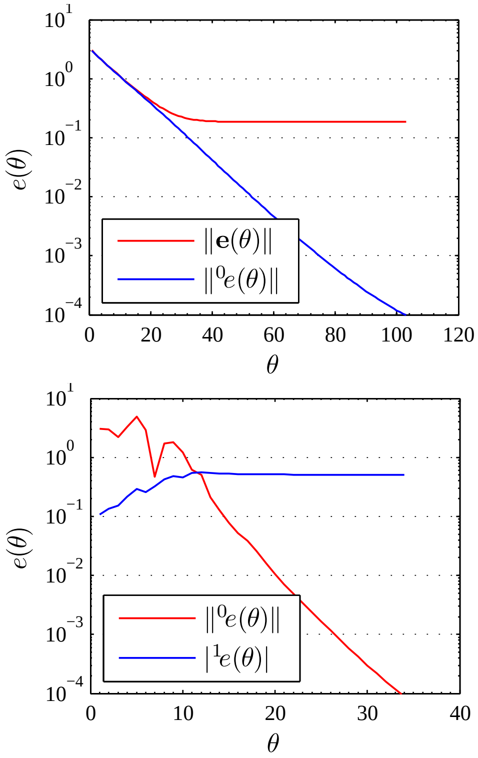

Figure 11. Convergence: egalitarian solution (top) and prioritarian solution (bottom)top.

algorithms have been applied to solve a multiple-task motion planning problem for the trident snake robot, that consists of the proper motion planning task and the singularity avoidance task. Numerical computations have shown that both the algorithms provide correct, although different, solutions to the problem. Also, the algorithms behave differently. In the case of egalitarian algorithm, when all the tasks are solved simultaneously, there are only two possibilities: either both tasks are solved correctly or none of the tasks is solved whatsoever. Contrary to that, in the prioritarian algorithm there is a risk that only the highest priority tasks will be solved. This observation has been confirmed by the plots of error convergence. In the egalitarian algorithm the total error decreases. It follows from the derivation of the prioritarian algorithm that the error of a lower priority task can even increase to enable the decrease of a higher priority task error. The parametric computations of solution of the motion planning problem are base-dependent, what usually appears to be quite restrictive. The non-parametric representation of control functions does not depend on the base choice, and is free from the limitations of the parametric approach. The computational effort in the parametric approach depends on the dimension of the parameter space. If the number of parameters is small, the parametric computations are much more efficient than the non-parametric. This, however, reverts, when the number of parameters grows up. In the example presented in this paper (63 control parameters), the parametric computations have been about 1.5 times more time consuming than the non-parametric. This being so, the choice of the motion planning strategy should depend on the complexity of the non-holonomic system subject to motion planning as well as on the type of the tasks defining the problem.

Appendix A

In this section we shall make a derivation of the Jacobian (23) and its Moore–Penrose inverse (24), and present the dynamics model of the trident snake robot used in Sect. 4.

A1 Subtask Jacobian and its inverse

The *i*-th subtask Jacobian is equal to

$${}^iJ_{x_0,T}(u(\cdot))v(\cdot) = D\,{}^iK_{x_0,T}(u(\cdot))v(\cdot)$$

$$= \frac{\mathrm{d}}{\mathrm{d}\alpha}\Big|_{\alpha=0}\int_0^T F_i(\varphi_{x_0,t}(u(\cdot)+\alpha v(\cdot)), u(t)+\alpha v(t))\mathrm{d}t,$$

where $x(t)=\varphi_{x_0,t}(u(\cdot))$ denotes the state trajectory of (1) driven by the control function $u(\cdot)$. The differentiation gives

$${}^iJ_{x_0,T}(u(\cdot))v(\cdot)$$

$$= \int_0^T \left(\frac{\partial F_i(x(t),u(t))}{\partial x} D\varphi_{x_0,t}(u(\cdot))v(\cdot) + \frac{\partial F_i(x(t),u(t))}{\partial u} v(t)\right)\mathrm{d}t.$$

Finally, a substitution for $\xi(t) = D\varphi_{x_0,t}(u(\cdot))v(\cdot)$ from (5) yields (23).

In order to find a formula for the Moore–Penrose Jacobian inverse, we begin with a Jacobian equation

$${}^iJ_{x_0,T}(u(\cdot))v(\cdot) = \eta,$$

where $\eta \in \mathbb{R}$. A solution of this equation will be sought by minimizing the squared norm of the control function,

$$\min_{v(\cdot)}\left(\|v(\cdot)\|_R^2 = \int_0^T v^T(t)Rv(t)\mathrm{d}t\right),$$

with the equality constraint. After the substitution

$$\xi(t) = \int_0^t \Phi(t,s)B(s)v(s)\mathrm{d}s,$$

the corresponding Lagrange function becomes

$$L(v(\cdot),\lambda) = \int_0^T v^T(t)Rv(t)\mathrm{d}t + \lambda\int_0^T \left(\frac{\partial F_i(x(t),u(t))}{\partial x}\right.$$

$$\left.\int_0^t \Phi(t,s)B(s)v(s)\mathrm{d}s + \frac{\partial F_i(x(t),u(t))}{\partial u} v(t)\mathrm{d}t\right),$$

$\lambda \in \mathbb{R}$. Now, the differentiation of the Lagrange function with respect to $v(\cdot)$

$$DL(v(\cdot),\lambda)w(\cdot) = \left.\frac{\mathrm{d}}{\mathrm{d}\alpha}\right|_{\alpha=0} L(v(\cdot)+\alpha w(\cdot),\lambda),$$

use of the identity

$$\int_0^T \int_0^t f(t,s)\mathrm{d}s\mathrm{d}t = \int_0^T \int_s^T f(t,s)\mathrm{d}t\mathrm{d}s,$$

and then equating the derivative to 0 yield

$$v_i(t) = -\frac{1}{2}\lambda R^{-1}(t)\left(B^T(t)\int_t^T \Phi^T(s,t)\left(\frac{\partial F_i(x(s),u(s))}{\partial x}\right)^T \mathrm{d}s + \left(\frac{\partial F_i(x(t),u(t))}{\partial u}\right)^T\right),$$

where the subscript i refers to the i-th subtask. To simplify the notations, let us define a pair of functions

$$b_i(t) = R^{-1}(t)B^T(t)\int_t^T \Phi^T(s,t)\left(\frac{\partial F_i(x(s),u(s))}{\partial x}\right)^T \mathrm{d}s$$

and

$$c_i(t) = R^{-1}(t)\left(\frac{\partial F_i(x(t),u(t))}{\partial u}\right)^T,$$

so that

$$v_i(t) = -\frac{1}{2}\lambda(b_i(t)+c_i(t)).$$

The Lagrange multiplier λ can be eliminated by inserting the control $v_i(t)$ into the Jacobian equation, that implies

$$-\frac{1}{2}\lambda\int_0^T \left(b_i^T(t)R(t)+c_i^T(t)R(t)\right)(b_i(t)+c_i(t))\mathrm{d}t$$
$$= -\frac{1}{2}\lambda\|b_i(\cdot)+c_i(\cdot)\|_R^2 = \eta.$$

From the last identity we compute λ, and conclude that

$$v_i(t) = \frac{b_i(t)+c_i(t)}{\|b_i(\cdot)+c_i(\cdot)\|_R^2}\eta,$$

what is just (24).

A2 Dynamics model of trident snake

A standard derivation based on the Lagrangian mechanics and d'Alembert principle leads to the following definition of terms appearing in the equations of motion (38) of the trident snake robot

$$\mathsf{P}(q) = \mathsf{M}^{-1}(q)\mathsf{G}^T(q)\mathsf{B}(q),$$
$$\mathsf{M}(q) = \mathsf{G}^T(q)\mathsf{Q}(q)\mathsf{G}(q),$$
$$\mathsf{N}(q,v) = -\mathsf{M}^{-1}(q)\mathsf{G}^T(q)\left(\mathsf{Q}(q)\dot{\mathsf{G}}(q)+\mathsf{C}(q,\mathsf{G}(q)v)\mathsf{G}(q)\right)v,$$

where $\mathsf{G}(q)$ is given by (37),

$$\mathsf{B}(q) = \begin{bmatrix} 0_{3\times 3} \\ I_3 \end{bmatrix}$$

denotes a control matrix, $\mathsf{Q}(q)$ is the inertia matrix defined below, and $\mathsf{C}(q,\dot q)$ is the matrix of Coriolis and centripetal forces whose entries

$$\mathsf{C}_{kj}(q,\dot q) = \sum_{i=1}^{6} c_{ij}^k(q)\dot q_i$$

are determined by the Christoffel's symbols of the first kind associated with $\mathsf{Q}(q)$

$$c_{ij}^k(q) = \frac{1}{2}\left(\frac{\partial \mathsf{Q}_{ik}(q)}{\partial q_j}+\frac{\partial \mathsf{Q}_{jk}(q)}{\partial q_i}-\frac{\partial \mathsf{Q}_{ij}(q)}{\partial q_k}\right), \quad i,j,k = 1,\ldots,6.$$

The following form of the inertia matrix for the trident snake robot can be found in Pietrowska (2012)

$$\mathsf{Q}(q) = \begin{bmatrix} m_{11} & 0 & m_{13} & m_{14} & m_{15} & m_{16} \\ 0 & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{13} & m_{23} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{14} & m_{24} & m_{34} & m_{44} & 0 & 0 \\ m_{15} & m_{25} & m_{35} & 0 & m_{55} & 0 \\ m_{16} & m_{26} & m_{36} & 0 & 0 & m_{66} \end{bmatrix},$$

where

$$
\begin{aligned}
&m_{11}=m_{22}=m_c,\\
&m_{33}=I_0+3I_{0\mathrm{w}}+3m_\mathrm{w}(r^2+l^2)+2m_\mathrm{w}rl\cos\phi_1\\
&+2m_\mathrm{w}rl\cos\phi_2+2m_\mathrm{w}rl\cos\phi_3+m_\mathrm{l}(l^2+3r^2+lr\cos\phi_1\\
&+lr\cos\phi_2+lr\cos\phi_3)+6m_\mathrm{m}r^2,\\
&m_{44}=m_{55}=m_{66}=I_{0\mathrm{w}}+m_\mathrm{w}l^2+\frac{1}{3}m_\mathrm{l}l^2,\\
&m_{13}=-m_\mathrm{w}l\sin(\alpha_1+\phi_1+\theta)-m_\mathrm{w}r\sin(\alpha_1+\theta)\\
&-m_\mathrm{w}l\sin(\alpha_2+\phi_2+\theta)-m_\mathrm{w}r\sin(\alpha_2+\theta)\\
&-m_\mathrm{w}l\sin(\alpha_3+\phi_3+\theta)-m_\mathrm{w}r\sin(\alpha_3+\theta)\\
&-\frac{1}{2}m_\mathrm{l}(2r\sin(\alpha_1+\theta)+l\sin(\alpha_1+\phi_1+\theta)+2r\sin(\alpha_2+\theta)\\
&+l\sin(\alpha_2+\phi_2+\theta)+2r\sin(\alpha_3+\theta)+l\sin(\alpha_3+\phi_3+\theta))\\
&-m_\mathrm{m}r\sin(\theta+\alpha_1)-m_\mathrm{m}r\sin(\theta+\alpha_2)-m_\mathrm{m}r\sin(\theta+\alpha_3),\\
&m_{14}=-m_\mathrm{w}l\sin(\alpha_1+\phi_1+\theta)-\frac{1}{2}m_\mathrm{l}l\sin(\alpha_1+\phi_1+\theta),\\
&m_{15}=-m_\mathrm{w}l\sin(\alpha_2+\phi_2+\theta)-\frac{1}{2}m_\mathrm{l}l\sin(\alpha_2+\phi_2+\theta),\\
&m_{16}=-m_\mathrm{w}l\sin(\alpha_3+\phi_3+\theta)-\frac{1}{2}m_\mathrm{l}l\sin(\alpha_3+\phi_3+\theta),\\
&m_{23}=m_\mathrm{w}l\cos(\alpha_1+\phi_1+\theta)+m_\mathrm{w}r\cos(\alpha_1+\theta)\\
&+m_\mathrm{w}l\cos(\alpha_2+\phi_2+\theta)+m_\mathrm{w}r\cos(\alpha_2+\theta)\\
&+m_\mathrm{w}l\cos(\alpha_3+\phi_3+\theta)+m_\mathrm{w}r\cos(\alpha_3+\theta)\\
&+\frac{1}{2}m_\mathrm{l}(2r\cos(\alpha_1+\theta)+l\cos(\alpha_1+\phi_1+\theta)+2r\cos(\alpha_2+\theta)\\
&+l\cos(\alpha_2+\phi_2+\theta)+2r\cos(\alpha_3+\theta)+l\cos(\alpha_3+\phi_3+\theta))\\
&+m_\mathrm{m}r\cos(\theta+\alpha_1)+m_\mathrm{m}r\cos(\theta+\alpha_2)+m_\mathrm{m}r\cos(\theta+\alpha_3),\\
&m_{24}=m_\mathrm{w}l\cos(\alpha_1+\phi_1+\theta)+\frac{1}{2}m_\mathrm{l}l\cos(\alpha_1+\phi_1+\theta),\\
&m_{25}=m_\mathrm{w}l\cos(\alpha_2+\phi_2+\theta)+\frac{1}{2}m_\mathrm{l}l\cos(\alpha_2+\phi_2+\theta),\\
&m_{26}=m_\mathrm{w}l\cos(\alpha_3+\phi_3+\theta)+\frac{1}{2}m_\mathrm{l}l\cos(\alpha_3+\phi_3+\theta),\\
&m_{34}=I_{0\mathrm{w}}+m_\mathrm{w}l^2+m_\mathrm{w}rl\cos\phi_1+\frac{1}{6}m_\mathrm{l}l(2l+3r\cos\phi_1),\\
&m_{35}=I_{0\mathrm{w}}+m_\mathrm{w}l^2+m_\mathrm{w}rl\cos\phi_2+\frac{1}{6}m_\mathrm{l}l(2l+3r\cos\phi_2),\\
&m_{36}=I_{0\mathrm{w}}+m_\mathrm{w}l^2+m_\mathrm{w}rl\cos\phi_3+\frac{1}{6}m_\mathrm{l}l(2l+3r\cos\phi_3).
\end{aligned}
$$

Acknowledgements. This research was supported by the Wrocław University of Technology under a statutory grant.

Edited by: A. Müller
Reviewed by: two anonymous referees

References

Antonelli, G.: Stablity analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems, IEEE Trans. Robotics, 25, 985–994, 2009.

Chiacchio, P., Chiaverini, S., Sciavicco, L., and Siciliano, B.: Closed loop inverse kinematics schemes for constrained redundant manipulators with the task space augmentation and task priority strategy, Int. J. Robotics Res., 10, 410–425, 1991.

Chiaverini, S.: Singularity-robust task-priority redundancy resolution for real time kinematic control of robot manipulators, IEEE Trans. Robotics Autom., 13, 398–410, 1997.

Chitour, Y. and Sussmann, H. J.: Motion planning using the continuation method, in: Essays on Mathematical Robotics, edited by: Baillieul, J., Sastry, S. S., and Sussmann, H. J., Springer-Verlag, New York, 91–125, 1998.

Choi, Y., Oh, Y., Oh, S., Park, J., and Chung, W. K.: Multiple task manipulation for a robotic manipulator, Adv. Robotics, 18, 637–653, 2004.

Divelbiss, A., Seereeram, S., and Wen, J. T.: Kinematic path planning for robots with holonomic and nonholonomic constraints, in: Essays on Mathematical Robotics, edited by: Baillieul, J., Sastry, S. S., and Sussmann, H. J., Springer-Verlag, New York, 127–150, 1998.

Gospodarek, S.: Design and modelling of the trident snake type mobile robot, Master's thesis, Wrocław University of Technology, 2011.

Ishikawa, M.: Trident snake robot: Locomotion analysis and control, in: Proc 6th IFAC Symp. NOLCOS, Stuttgart, Germany, 1169–1174, 2004.

Ishikawa, M., Minati, Y., and Sugie, T.: Development and control experiment of the trident snake robot, IEEE/ASME Trans. Mechatronics, 15, 9–16, 2010.

Lewis, A.: When is a mechanical control system kinematic?, in: Proc. 38th IEEE CDC, Phenix, Arizona, 1162–1167, 1999.

Maciejewski, A. A. and Klein, C. A.: Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, Int. J. Robotics Res., 4, 109–117, 1985.

Nakamura, Y., Hanafusa, H., and Yoshikawa, T.: Task-priority based redundancy control of robot manipulators, Int. J. Robotics Res., 6, 3–15, 1987.

Paszuk, D., Tchoń, K., and Pietrowska, Z.: Motion planning of the trident snake robot equipped with passive or active wheels, in: Bull. Pol. Acad. Sci., Tech. Sci., 60, 547–554, 2012.

Pietrowska, Z.: Kinematics, dynamics, and control of a trident snake nonholonomic system, Master's thesis, Wrocław University of Technology, 2012.

Ratajczak, A.: Motion planning of underactuated robotic systems, Ph.D. thesis, Wrocław University of Technology, 2012.

Ratajczak, A., Karpińska, J., and Tchoń, K.: Task-priority motion planning of underactuated systems: An endogenous configuration space approach, Robotica, 28, 885–892, 2010.

Sontag, E. D.: Mathematical Control Theory, Springer-Verlag, New York, 1990.

Sussmann, H. J.: A continuation method for non-holonomic path finding problems, in: Proc. 32nd IEEE CDC, San Antonio, Texas, 2718–2723, 1993.

Tchoń, K. and Jakubiak, J.: Endogenous configuration space approach to mobile manipulators: A derivation and performance assessment of Jacobian inverse kinematics algorithms, Int. J. Control, 76, 1387–1419, 2003.

Tchoń, K. and Zadarnowska, K.: Kinematic dexterity of mobile manipulators: An endogenous configuration space approach, Robotica, 21, 521–530, 2003.

Zadarnowska, K. and Tchoń, K.: A control theory framework for performance evaluation of mobile manipulators, Robotica, 25, 703–715, 2007.