J Intell Robot Syst (2017) 85:511–522
DOI 10.1007/s10846-016-0394-4

Lagrangian Jacobian Motion Planning: A Parametric Approach

Ida Góral · Krzysztof Tchoń

Received: 12 December 2015 / Accepted: 24 May 2016 / Published online: 13 July 2016

Abstract This paper addresses the motion planning problem of nonholonomic robotic systems. The system's kinematics are described by a driftless control system with output. It is assumed that the control functions are represented in a parametric form, as truncated orthogonal series. A new motion planning algorithm is proposed based on the solution of a Lagrange-type optimisation problem stated in the linear approximation of the parametrised system. Performance of the algorithm is illustrated by numeric computations for a motion planning problem of the rolling ball.

Keywords Jacobian algorithm · Motion planning · Nonholonomic system · Parametrisation

1 Introduction

The motion planning problem of robotic systems can be regarded as a sort of an inverse kinematics problem, and solved by application of the continuation method prototyped in the work of Ważewski [1]. Such an approach gives rise to various Jacobian motion planning algorithms. In the context of robotics the continuation method was introduced by Sussmann [2], and then widely used in motion planning of robotic systems, such as mobile robots [3, 4], mobile manipulators [5], rolling bodies [6], as well as developed theoretically [7, 8]. This paper concentrates exclusively on nonholonomic systems subject to Pfaffian phase constraints, whose kinematics are represented by a driftless control system with output. In such cases the motion planning problem consists of inverting the end point map of the control system. A system Jacobian is introduced by means of the linear approximation to the original system [9]. A Jacobian inverse can then be derived by solving a constrained optimisation problem in the linearised system. Commonly, for the constrained optimisation problem a minimisation of the control squared norm (equating its energy) is employed. In this way the Jacobian pseudoinverse is obtained. As a natural generalization, in [10] we have designed a Lagrangian Jacobian inverse, based on the minimisation of the Lagrangian objective function, which takes into consideration both the control and the trajectory of the the linearised system. This new inverse leads to the Lagrangian Jacobian motion planning algorithm. It has been noticed that by a proper choice of the Lagrangian objective function this algorithm, beyond solving the primary motion planning problem, also allows to shape the system's performance. Some of those features were presented in a preliminary way in [11] and then proven in [12].

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

Ida Góral (✉)
Chair of Cybernetics and Robotics, Electronics Faculty, Wroclaw University of Technology, ul. Janiszewskiego 11/17, 50–372 Wroclaw, Poland
e-mail: ida.goral@pwr.edu.pl

Jacobian motion planning algorithms can be run in either parametric or non-parametric formulation [13]. For practical and computational reasons the parametric formulation is beneficial, based on transferring the Lagrangian Jacobian motion planning algorithm from the infinite dimensional space of control functions to a finite dimensional space of control parameters. This is obtained by introducing a parametrisation of control functions by truncated orthogonal series, e.g. Fourier, Legendre, Chebyshev, Laguerre, etc. [14]. In this paper, assuming the continuation method, the parametric form of the Lagrangian Jacobian inverse is introduced, referred to as the parametric Lagrangian Jacobian inverse. The parametric inverse is derived by means of the methods of the calculus of variations. Along the new inverse a respective Jacobian motion planning algorithm is proposed, called parametric Lagrangian Jacobian. Performance features of this algorithm are examined, focusing on its capabilities to shape the system's trajectories. Three specific cases are distinguished in this context: keeping the trajectory or control close to the initial one, bounding the length of the resulting trajectory, and obstacle avoidance. These features of performance have been demonstrated by numeric solutions of a number of motion planning problems for the kinematics of the rolling ball.

The main contribution of this paper lies in the derivation of the parametric Lagrangian Jacobian inverse and of the respective motion planning algorithm. Additionally, with regard to the provided algorithm, a proof of bounding the length of resulting trajectory is presented. Performance of the Lagrangian motion planning algorithm is illustrated with motion planning problems of the rolling ball in the plane. Admittedly, the primary motion planning problem for this specific example could have been solved by a sort of the inverse dynamics approach, however, our intention has been to demonstrate the trajectory shaping capabilities of the Lagrangian motion planing. By design, the Lagrangian algorithm applies far beyond the scope of the inverse dynamics approach. As has been said, the driving force behind this paper is the calculus of variations. Alternatively, a non-classical formulation of the Pontryagin maximum principle could be employed [15].

The remaining part of this paper is composed in the following way. Section 2 presents the basic concepts. The parametrisation of control functions is presented in Section 3. Section 4 contains the main result of this paper, i.e. the parametric Lagrangian Jacobian inverse. Performance features and the discrete version of the corresponding motion planning algorithm are described in Section 5. A performance study of the parametric Lagrangian Jacobian motion planning algorithm is included in Section 6. Section 7 concludes the paper. Proofs of the main results are exposed in Appendix.

2 Basic Concepts

The kinematics of a nonholonomic robotic system can be represented as a driftless control system with output

$$\begin{cases} \dot{q} = G(q)u = \sum_{i=1}^{m} G_i(q)u_i, \\ y = k(q), \end{cases} \tag{1}$$

where $q = (q_1, q_2, \ldots, q_n)^T \in \mathrm{R}^n$ is the state variable, $u = (u_1, u_2, \ldots, u_m)^T \in \mathrm{R}^m$ represents the control and $y = (y_1, y_2, \ldots, y_r)^T \in \mathrm{R}^r$ stands for the output variable. The functions and vector fields appearing in Eq. 1 are assumed smooth. Let $T > 0$ denote a control time horizon. The admissible control functions $u(\cdot)$ belong to the space $L^2_m[0, T]$ of Lebesgue square integrable functions defined on the interval $[0, T]$, with inner product

$$\langle u_1(\cdot), u_2(\cdot)\rangle_S = \int_0^T u_1^T(t)S(t)u_2(t)\,dt. \tag{2}$$

The matrix $S(t) = S^T(t) > 0$ imposes certain weights on components of the control.

Let $q(t) = \varphi_{q_0,t}(u(\cdot))$ denote the state trajectory of Eq. 1 starting from q_0 and driven by a control function $u(\cdot)$. This control function will be called admissible if $q(t)$ exists for every $t \in [0, T]$ and every $q_0 \in \mathrm{R}^n$. In the control system (1) the motion planning problem can be formulated as follows: given an initial state q_0 and a time horizon T, find a control $u(t)$ steering the system's output to a desired point y_d in the task space, so that $y(T) = y_d$.

The end-point map of system (1), defined as the value of its output function at T driven by the control $u(\cdot)$, takes the form of

$$K_{q_0,T}(u(\cdot)) = k(q(T)) = k(\varphi_{q_0,T}(u(\cdot))). \tag{3}$$

For bounded measurable control functions $u(\cdot) \in \mathcal{U} \subset L^2_m[0, T]$ the end-point map $K : \mathcal{U} \to \mathrm{R}^r$ is continuously differentiable (of class C^1), [9]. Its derivative can be computed by means of the linear approximation to system (1) along $(u(t), q(t))$,

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

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

$$A(t) = \frac{\partial(G(q(t))u(t))}{\partial q}, \quad B(t) = G(q(t)),$$
$$C(t) = \frac{\partial k(q(t))}{\partial q}. \tag{5}$$

The derivative of system's end-point map with respect to control will be referred to as the system's Jacobian

$$J_{q_0,T}(u(\cdot))v(\cdot) = \mathcal{D}K_{q_0,T}(u(\cdot))v(\cdot) = C(T)\xi(T) = \eta(T).$$

Solving Eq. 4 for $\xi(0) = 0$ yields

$$\xi(T) = \int_0^T \Phi(T, t)B(t)v(t)\,dt, \tag{6}$$

where the transition matrix $\Phi(t, w)$ satisfies the following equation

$$\frac{\partial \Phi(t, w)}{\partial t} = A(t)\Phi(t, w), \qquad \Phi(w, w) = I_n. \tag{7}$$

Therefore, the Jacobian can be expressed as

$$J_{q_0,T}(u(\cdot))v(\cdot) = C(T)\int_0^T \Phi(T, t)B(t)v(t)\,dt. \tag{8}$$

For a nonholonomic system with kinematics described by Eq. 1, the motion planning problem can be expressed in terms of finding a control function $u_d(\cdot) \in \mathcal{U}$ such that

$$K_{q_0,T}(u_d(\cdot)) = y_d. \tag{9}$$

This motion planning problem can be solved using the continuation method which leads to a Jacobian motion planning algorithm. The algorithm arises as follows: consider an arbitrary chosen control function $u_0(\cdot) \in \mathcal{U}$. If this function solves the problem, we are done. Otherwise, a deformation of $u_0(\cdot)$ is made into a smooth curve $u_\theta(\cdot) \in \mathcal{U}, \theta \in \mathrm{R}$, $u_{\theta=0}(\cdot) = u_0(\cdot)$. The motion planning error along this curve amounts to

$$e(\theta) = K_{q_0,T}(u_\theta(\cdot)) - y_d \tag{10}$$

and is requested to decrease exponentially

$$\frac{de(\theta)}{d\theta} = -\gamma e(\theta), \qquad \gamma > 0. \tag{11}$$

By differentiation of the error we get an implicit differential equation

$$J_{q_0,T}(u_\theta(\cdot))\frac{du_\theta(\cdot)}{d\theta} = -\gamma(K_{q_0,T}(u_\theta(\cdot)) - y_d), \tag{12}$$

which can be made explicit by application of any right inverse of the Jacobian, $J^{\#}_{q_0,T}(u(\cdot)) : \mathrm{R}^r \to \mathcal{U}$, so that

$$\frac{du_\theta(\cdot)}{d\theta} = -\gamma J^{\#}_{q_0,T}(u_\theta(\cdot))(K_{q_0,T}(u_\theta(\cdot)) - y_d). \tag{13}$$

By computing the control function as the limit

$$u_d(t) = \lim_{\theta \to +\infty} u_\theta(t), \tag{14}$$

a solution to the motion planning problem is obtained.

The Eq. 13 defines any Jacobian motion planning algorithm for the system (1). A commonly used right inverse of the Jacobian is obtained by minimisation of the control energy,

$$\min_{v(\cdot)} \int_0^T v^T(t)v(t)\,dt, \tag{15}$$

on condition that $J_{q_0,T}(u(\cdot))v(\cdot) = C(T)\xi(T) = \eta$, which results in the Jacobian pseudoinverse [5]

$$\left(J^{P\#}_{q_0,T}(u(\cdot))\eta\right)(t) = B^T(t)\Phi^T(T, t)C^T(T)\mathcal{G}^{-1}_{q_0,T}(u(\cdot))\eta. \tag{16}$$

Hereabove $\mathcal{G}_{q_0,T}(u(\cdot))$ is the output controllability Gramian of the system (4) that can be computed as

$$\mathcal{G}_{q_0,T}(u(\cdot)) = C(T)\int_0^T \Phi(T, t)B(t)B^T(t)\Phi^T(T,t)\,dt\,C^T(T). \tag{17}$$

In the context of mobile robotics the Gramian is identified with the mobility matrix. It is easily seen that the mobility matrix can be computed by solving for $M(0) = 0$ the Lyapunov differential equation

$$\dot{M}(t) = B(t)B^T(t) + A(t)M(t) + M(t)A^T(t), \tag{18}$$

and then setting $\mathcal{G}_{q_0,T}(u(\cdot)) = C(T)M(T)C^T(T)$. In order for the Jacobian pseudoinverse to exist the Gramian must be of full rank r. In such cases the control function $u(\cdot)$ is called regular, otherwise it is singular.

3 Parametrisation

Generally, in order to improve the efficiency of the computations, a finite-dimensional representation of control functions by the truncated orthogonal series can be employed. Such an approach is referred to as parametric. Consider a row matrix $P_b(t) = [\phi_0(t), \phi_1(t), \ldots, \phi_p(t)]$ whose entries are some basic functions defined on the interval $[0, T]$. The control function of system (1) can be expressed as

$$u_\lambda(t) = P(t)\lambda, \tag{19}$$

where

$$P(t) = \mathrm{diag}\{P_b(t), P_b(t), \ldots, P_b(t)\} \tag{20}$$

is a block diagonal matrix consisting of m copies of $P_b(t)$, and $\lambda \in \mathrm{R}^s$, $s = m(p+1)$, denotes a vector of control parameters. The basic functions are assumed orthogonal with respect to the inner product (2), so

$$\int_0^T P^T(t)S(t)P(t)\,dt = I_s. \tag{21}$$

On account of Eq. 19, the control system (1) takes form of

$$\begin{cases} \dot{q} = G(q)u_\lambda(t), \\ y = k(q). \end{cases} \tag{22}$$

Let $q_\lambda(t) = \varphi_{q_0,t}(u_\lambda(\cdot))$ denote the trajectory of Eq. 22 starting from q_0. The end-point map of Eq. 22 can be expressed as $K_{q_0,T}(\lambda) = k(q_\lambda(T))$. With some abuse of notation the parametric Jacobian can be computed as

$$J_{q_0,T}(\lambda) = \left.\frac{d}{d\sigma}\right|_{\sigma=0} k(q_{\lambda+\sigma\eta}(T)) = C_\lambda(T)\int_0^T \Phi_\lambda(T,t)B_\lambda(t)P(t)\,dt, \tag{23}$$

where matrices $A_\lambda(t)$, $B_\lambda(t)$, $C_\lambda(t)$ are given by Eq. 5 taken along $(u_\lambda(t), q_\lambda(t))$, and

$$\frac{\partial\Phi_\lambda(t,w)}{\partial t} = A_\lambda(t)\Phi_\lambda(t,w), \qquad \Phi_\lambda(w,w) = I_n. \tag{24}$$

The parametric Jacobian $J_{q_0,T}(\lambda)$ is a matrix of dimension $r \times s$. It can be observed that $J_{q_0,T}(\lambda) = C_\lambda(T)J_\lambda(T)$, where $J_\lambda(t)$ fulfils a matrix differential equation

$$\dot{J}_\lambda(t) = A_\lambda(t)J_\lambda(t) + B_\lambda(t)P(t), \tag{25}$$

with initial condition $J_\lambda(0) = 0$.

The motion planning problem for system (22) is described in the following way: find a vector of control coefficients $\lambda_d \in \mathrm{R}^s$ which satisfies $K_{q_0,T}(\lambda_d) = y_d$. Again, the continuation method leads to a Ważewski-Davidenko equation

$$J_{q_0,T}(\lambda(\theta))\frac{d\lambda(\theta)}{d\theta} = -\gamma_\lambda(K_{q_0,T}(\lambda(\theta)) - y_d), \tag{26}$$

which, with application of some right inverse of the Jacobian, can be transformed into a more explicit form

$$\frac{d\lambda(\theta)}{d\theta} = -\gamma_\lambda J^{\#}_{q_0,T}(\lambda(\theta))(K_{q_0,T}(\lambda(\theta)) - y_d). \tag{27}$$

The solution of the motion planning problem is achieved as the limit $\lambda_d = \lim_{\theta\to+\infty}\lambda(\theta)$.

4 Parametric Lagrangian Jacobian Inverse

As mentioned before, a right inverse of the Jacobian (23) can be obtained by solving a constrained optimisation problem for the linearised system

$$\begin{cases} \dot{\xi}_\lambda(t) = A_\lambda(t)\xi_\lambda(t) + B_\lambda(t)P(t)\mu, \\ \eta(t) = C_\lambda(t)\xi_\lambda(t), \end{cases} \tag{28}$$

where

$$\xi_\lambda(t) = \frac{\partial\varphi_{q_0,t}(u_\lambda(\cdot))}{\partial\lambda}\mu = \mathcal{D}\varphi_{q_0,t}(u_\lambda(\cdot))P(t)\mu \tag{29}$$

denotes a variation of system's trajectory corresponding to a variation μ of control coefficients. In consequence

$$q_{\lambda+\mu}(t) = \varphi_{q_0,t}(u_{\lambda+\mu}(\cdot)) \cong q_\lambda(t) + \xi_\lambda(t). \tag{30}$$

A natural generalisation of the objective function considered in Eq. 15 is an objective function in the Lagrange form, that leads to the Lagrange-type optimisation problem in the linear system described by Eq. 28

$$\min_\mu \int_0^T \left(\xi_\lambda^T(t)Q(t)\xi_\lambda(t) + \mu^T P^T(t)R(t)P(t)\mu\right)dt, \tag{31}$$

where $Q(t) = Q^T(t) \geq 0$ and $R(t) = R^T(t) > 0$, on condition that $J_{q_0,T}(\lambda)\mu = C_\lambda(T)\xi_\lambda(T) = \eta$. The resulting Jacobian inverse will be referred to as the parametric Lagrangian Jacobian inverse $J^{\mathcal{L}\#}_{q_0,T}(\lambda)$. The following theorem establishes an explicit form of the parametric Lagrangian Jacobian inverse. Its proof can be found in Appendix (A.1).

Theorem 1 *The parametric Lagrangian Jacobian inverse*

$$J^{\mathcal{L}\#}_{q_0,T}(\lambda) = I^{-1}_{\lambda}(T)F^T_{\lambda}(T)C^T_{\lambda}(T)\mathcal{M}^{-1}_{q_0,T}(\lambda), \quad (32)$$

where the mobility matrix

$$\mathcal{M}_{q_0,T}(\lambda) = C_{\lambda}(T)F_{\lambda}(T)I^{-1}_{\lambda}(T)F^T_{\lambda}(T)C^T_{\lambda}(T), \quad (33)$$

and the matrices $I_{\lambda}(t)$ and $F_{\lambda}(t)$ solve the differential equations

$$\begin{aligned} \dot{I}_{\lambda}(t) &= F^T_{\lambda}(t)Q(t)F_{\lambda}(t) + P^T(t)R(t)P(t), \\ \dot{F}_{\lambda}(t) &= B_{\lambda}(t)P(t) + A_{\lambda}(t)F_{\lambda}(t), \end{aligned} \quad (34)$$

with initial conditions $I_{\lambda}(0) = 0$, $F_{\lambda}(0) = 0$.

The mobility matrix will be assumed to have full rank r. It is easily shown that, after setting $Q(t) = 0$ and $R(t) = I_m$, the Lagrangian Jacobian inverse and the Jacobian pseudoinverse coincide.

5 Motion Planning

Given the parametric form (32) of the Lagrangian Jacobian inverse, the Lagrangian Jacobian motion planning algorithm is obtained by inserting this inverse into Eq. 27. In the sequel we shall first analyse performance features of the algorithm concerned with shaping trajectories of the robotic system, and then transform the parametric motion planning algorithm to a discrete setting.

5.1 Performance

The parametric Lagrangian Jacobian algorithm, beyond solving the motion planning problem, is also able to shape the trajectories of system (22) due to appropriate choice of matrices $Q(t)$ and $R(t)$. In order to support this claim the following cases will be studied. To begin with, for a fixed value of parameter θ, let vector μ_θ be the solution of the Jacobian equation

$$J_{q_0,T}(\lambda(\theta))\mu_\theta = K_{q_0,T}(\lambda(\theta)) - y_d, \quad (35)$$

provided by the parametric Lagrangian Jacobian inverse, i.e.

$$\mu_\theta = J^{\mathcal{L}\#}_{q_0,T}(\lambda(\theta))(K_{q_0,T}(\lambda(\theta)) - y_d). \quad (36)$$

Invoking (27) implies that

$$\frac{d\lambda(\theta)}{d\theta} = -\gamma_\lambda \mu_\theta. \quad (37)$$

Now, the differentiation of the parametrised system's trajectory $q_{\lambda(\theta)}(t) = \varphi_{q_0,t}(u_{\lambda(\theta)}(\cdot))$,combined with Eqs. 29 and 37, results in

$$\frac{q_{\lambda(\theta)}(t)}{d\theta} = \frac{\partial\varphi_{q_0,t}(u_{\lambda(\theta)}(\cdot))}{\partial\lambda}\frac{d\lambda(\theta)}{d\theta} = -\gamma_\lambda\xi_{\lambda(\theta)}(t). \quad (38)$$

By inspection of the identities (37) and (38) it can be seen that vector μ_θ and function $\xi_{\lambda(\theta)}(\cdot)$ can be regarded as directions of motion in the control and in the trajectory space, respectively.

Given a control curve $\lambda(\theta)$, let us choose a smooth curve $c_\theta(t)$ in the state space of system (22), parametrised by θ, and let $V_\theta(t)$ denote a vector field along $c_\theta(t)$. Allowing the matrix $Q(t)$ to be made θ-dependent, we set

$$Q_\theta(t) = V_\theta(t)V^T_\theta(t), \quad (39)$$

Then, for a fixed θ, the objective function (31) becomes

$$\int_0^T \left((\xi^T_{\lambda(\theta)}(t)V_\theta)^2 + \mu^T_\theta P^T(t)R_\theta(t)P(t)\mu_\theta\right)dt, \quad (40)$$

where matrix $R(t)$ is also allowed to depend on θ. With a suitable choice of the vector field $V_\theta(t)$ and the matrix $R_\theta(t)$, the minimisation of this objective function will prefer the direction of motion $\xi_{\lambda(\theta)}(\cdot)$ close to the orthogonal to $V_\theta(t)$ at each t. In the next section w shall show how this property may contribute to the obstacle avoidance.

Furthermore, there exists another possibility of using the matrix $Q(t)$ for shaping system's trajectories. Once again, consider the optimisation problem (31). In case when both the matrices $Q(t)$ and $R(t)$ are positive and do not depend on θ, the following inequality can be proved, see [12],

$$\begin{aligned} &\int_0^T (q_{\lambda(\theta)} - q_0)^T(t)Q(t)(q_{\lambda(\theta)} - q_0)\,dt + \\ &\int_0^T (\lambda(\theta) - \lambda_0)^T P^T(t)R(t)P(t)(\lambda(\theta) - \lambda_0)\,dt \le \gamma^2\theta \times \\ &\int_0^\theta\int_0^T \left(\xi^T_{\lambda(\alpha)}(t)Q(t)\xi_{\lambda(\alpha)}(t) + \mu^T_\alpha P^T(t)R(t)P(t)\mu_\alpha\right)dt\,d\alpha. \end{aligned} \quad (41)$$

It is easily noticed that the inner integral on the right hand side of the inequality represents the objective function (31) for a fixed α. As such, Eq. 41 imposes an upper bound on the distance between the system's current trajectory and control, and the initial ones. Pursuing further this line of reasoning, one can observe that the relative weight assigned to matrices $Q(t)$ and $R(t)$ results in favouring one part of the minimised objective function – either the variation of the system's trajectory, $\xi_\lambda(\theta)$, or the energy of control variations, $P(t)\mu_\theta$. This dependency manifests in an attract–repel behaviour between the final and initial trajectory/control of the system. For a sufficiently large matrix $Q(t)$ the motion planning algorithm steers the system as close to the initial trajectory as possible. In contrary, if the matrix $R(t)$ dominates, then the resulting plan of motion approaches the one obtained by employment of the Jacobian pseudoinverse.

Yet another choice of the matrix $Q(t)$ allows one to further shape the system's trajectories. Specifically, for a curve $\lambda(\theta)$, let $Q_\theta(t) = A^T_{\lambda(\theta)}(t)A_{\lambda(\theta)}(t)$. We have the following result.

Lemma 1 *For the matrix $Q_\theta(t) = A^T_{\lambda(\theta)}(t)A_{\lambda(\theta)}(t)$, where $A_{\lambda(\theta)}(t) = \frac{\partial(G(q_{\lambda(\theta)}(t))u_{\lambda(\theta)}(t))}{\partial q}$, and the matrix $R_\theta(t) = B^T_{\lambda(\theta)}(t)B_{\lambda(\theta)}(t)$, for $B_{\lambda(\theta)}(t) = G(q_{\lambda(\theta)}(t))$, $t \in [0, T]$, the length of $\xi_{\lambda(\theta)}(t)$ is upper-bounded in the following way*

$$\int_0^T ||\dot{\xi}_{\lambda(\theta)}(t)||dt \le \sqrt{2T} \times\sqrt{\int_0^T (\xi^T_{\lambda(\theta)}(t)Q_\theta(t)\xi_{\lambda(\theta)}(t) + \mu^T_\theta P(t)R_\theta(t)P(t)\mu_\theta)dt}. \quad (42)$$

A proof of this lemma is found in Appendix (A.2). It can be noticed that the integral on the right hand side of Eq. 42 represents the objective function minimised by the algorithm Thus, the choice of matrix $Q(t)$ as in the lemma bounds from above the total length of all variations of the system's trajectory (30) in the subsequent steps of the motion planning algorithm.

5.2 Algorithm

Associated with the parametric Lagrangian Jacobian inverse is a parametric Lagrangian Jacobian motion planning algorithm. For computational convenience the algorithm will be represented in a discrete form based on the well-known Euler method of solving differential equations. The advantage of such a formulation lies in easy implementability often accompanied with satisfactory accuracy. Specifically, the θ variable is discretised with a step length h, such that at the ith step $\theta_{i+1} = \theta_i + h$. Let $\lambda_i = \lambda(\theta_i)$. Therefore, the Eq. 27 is transformed into the difference equation

$$\lambda_{i+1} = \lambda_i - h\gamma_\lambda J^{\mathcal{L}\#}_{q_0,T}(\lambda_i)(K_{q_0,T}(\lambda_i) - y_d) \quad (43)$$

initialised at $\lambda_0 \in R^s$. Set $\gamma = h\gamma_\lambda$. Having established this, a computational scheme of the motion planning algorithm is tantamount to solving for λ the following set of diffrential-difference equations

$$\begin{cases} \dot{q}_{\lambda_i}(t) = G(q_{\lambda_i}(t))P(t)\lambda_i, \\ \dot{F}_{\lambda_i}(t) = B_{\lambda_i}(t)P(t) + A_{\lambda_i}(t)F_{\lambda_i}(t), \\ \dot{I}_{\lambda_i}(t) = F^T_{\lambda_i}(t)Q(t)F_{\lambda_i}(t) + P^T(t)R(t)P(t), \\ \lambda_{i+1} = \lambda_i - \gamma J^{\mathcal{L}\#}_{q_0,T}(\lambda_i)(K_{q_0,T}(\lambda_i) - y_d), \\ J^{\mathcal{L}\#}_{q_0,T}(\lambda) = I^{-1}_\lambda(T)F^T_\lambda(T)C^T_\lambda(T)\mathcal{M}^{-1}_{q_0,T}(\lambda), \\ \mathcal{M}_{q_0,T}(\lambda) = C_\lambda(T)F_\lambda(T)I^{-1}_\lambda(T)F^T_\lambda(T)C^T_\lambda(T), \end{cases} \quad (44)$$

with initial conditions $q_{\lambda_i}(0) = q_0$, $F_{\lambda_i}(0) = 0$, $I_{\lambda_i}(0) = 0$, a chosen base representation of control functions $P_b(t)$, $P(t) = \mathrm{diag}\{P_b(t), P_b(t), \ldots, P_b(t)\}$, initial vector of control parameters λ_0, and desired point in the task space y_d. The solution of the motion planning problem is obtained as $\lambda_d = \lim_{i\to+\infty} \lambda_i$. In the next section this algorithm will be applied in order to solve example motion planning problems for a rolling ball. Trajectory shaping features of the algorithm will be emphasised.

6 Simulation

To illustrate performance of the Lagrangian motion planning algorithm, the kinematics of a rolling ball will be used. The ball's schematic view is presented in Fig. 1. The ball's coordinates are chosen as $q = (x_1, x_2, \phi, \theta, \psi)^T$, whose meaning is the following: (x_1, x_2) – the position of the ball's contact point with the ground in the space frame $X_0Y_0Z_0$, (ϕ, θ) – the position (azimuth and elevation) of the contact point in the body frame $X_kY_kZ_k$, and ψ – the ball's orientation defined as the angle between the X_0 axis and the axis X_P of a frame attached to the contact point.

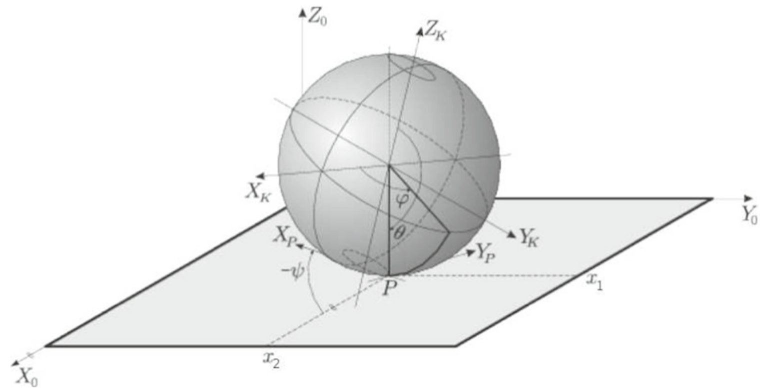

Fig. 1 The rolling ball

Its kinematics are represented by the driftless control system with output

$$\dot{q} = \begin{bmatrix} \sin\theta\sin\psi & \cos\psi \\ -\sin\theta\cos\psi & \sin\psi \\ 1 & 0 \\ 0 & 1 \\ -\cos\theta & 0 \end{bmatrix} u_\lambda = G(q)u_\lambda,$$

$$k(q) = (x_1, x_2)^T. \tag{45}$$

All computations of the algorithm (44) have been performed for the following parameters and initial conditions: $\gamma = 0.01$, $q(0) = (0, 0, 0, 0, 0)^T$, $y_d = (1, 1)^T$, $T = 2$. The control functions are taken in the form of truncated trigonometric series containing the constant term and up to the 2nd order harmonics. The stopping condition involves a decrease of the motion planning error below the predefined value, i.e. $||e(q(T))|| < 10^{-4}$. By definition this error is expressed in the length units.

The feature of directing the ball's trajectory in order to get the obstacle avoidance can be demonstrated in the following way. Suppose that $q_{\lambda(\theta)}(t)$ is the ball's trajectory for a certain fixed value of parameter θ. Consider a set $O = \{o_1, \ldots, o_p\}$ of p task space point-obstacles in R^2. Due to the form of the output function in Eq. 45 the task space trajectory equals $k(q_{\lambda(\theta)}(t)) = (q_{1\lambda(\theta)}(t), q_{2\lambda(\theta)}(t))^T = (x_1(t), x_2(t))^T$. For the obstacle o_l we define a direction

$$D_{l\theta}(t) = w_l Rot(Z, \frac{\pi}{2}) \frac{o_l - k(q_{\lambda(\theta)}(t))}{||o_l - k(q_{\lambda(\theta)}(t))||} \in \mathrm{R}^2, \tag{46}$$

orthogonal to the direction between the system's trajectory at t and the obstacle o_l. $Rot(Z, \frac{\pi}{2}) =$

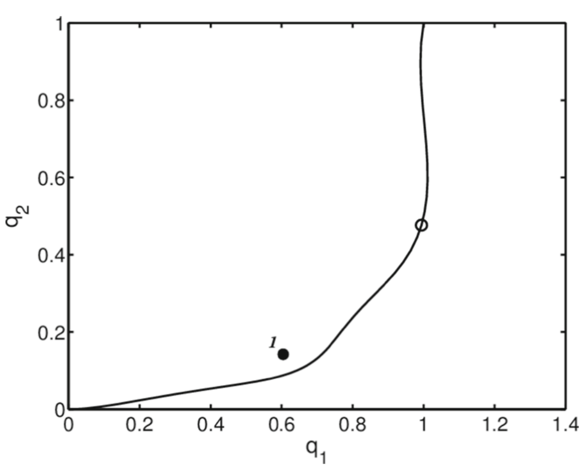

Fig. 2 Motion planning with one obstacle

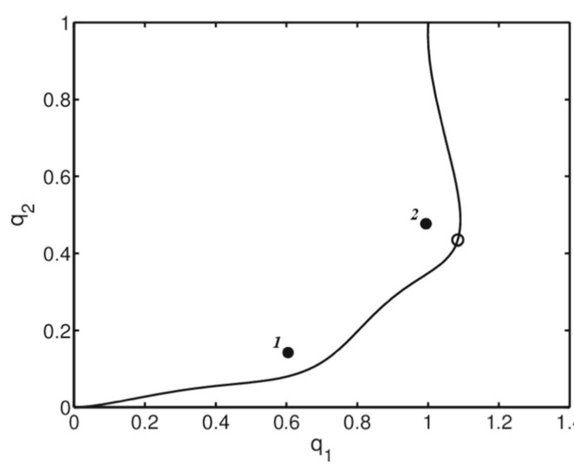

Fig. 3 Motion planning with two obstacles

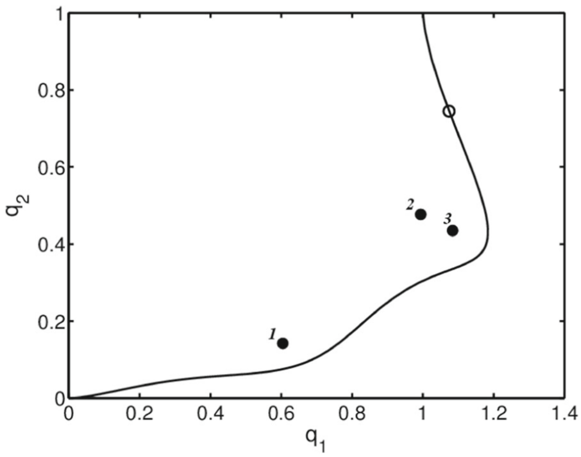

Fig. 4 Motion planning with three obstacles

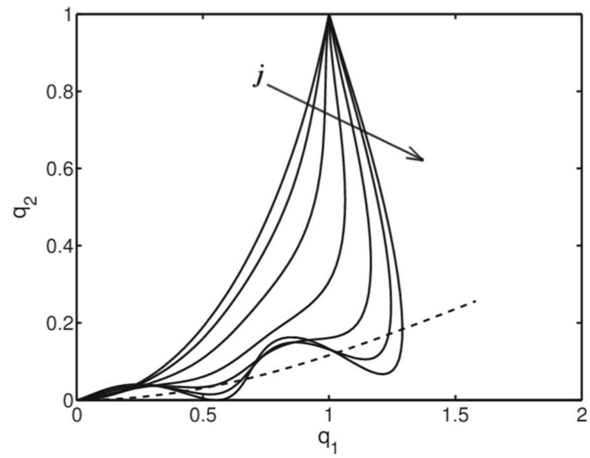

Fig. 6 Motion planning with dominating matrix $Q(t)$

$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ refers to the rotation by $\frac{\pi}{2}$ around the Z axis, perpendicular to the task space. The weighting coefficient

$$w_l = \begin{cases} 1 & if \ \ ||o_l - k(q_{\lambda(\theta)})|| \geq d, \\ w_p & otherwise, \end{cases} \tag{47}$$

expresses the penalty for invading the discomfort zone d in the vicinity of an obstacle, with w_p defining the penalty's weight. Using this data we set

$V_{l\theta}(t) = (D_{l\theta}(t), 0, \ldots, 0)^T \in \mathrm{R}^n$

and define the matrix $Q(t)$ as

$$Q_\theta(t) = \sum_{l=1}^{p} V_{l\theta}(t) V_{l\theta}^T(t). \tag{48}$$

As a result, the Lagrangian Jacobian motion planning achieves two goals – minimises the control energy and, what may be favoured, elicits a motion in the task space in the direction orthogonal to $D_\theta(t)$, and thus repels the task space trajectory $k(q_{\lambda(\theta)}(t))$ from obstacles. To illustrate this behaviour a set of simulations was performed. Firstly, the motion planning problem with no obstacles and $Q(t) = I_5$ was solved. Secondly, a new obstacle was being added, directly on the path resulting from the previous solution. The new problem was then solved with the matrix $Q(t)$ obtained from Eq. 48. Such steps were repeated three more times, thus up to four obstacles were defined. The computations were made for matrix $R(t) = I_2$, initial control $u_{\lambda_0}(t) = (-0.2, 1)^T$ and parameters $d = 0.1$, $w_p = 100$. The additional constant gain

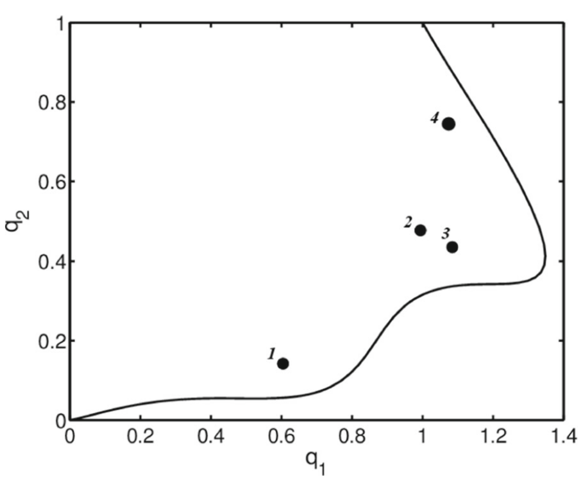

Fig. 5 Motion planning with four obstacles

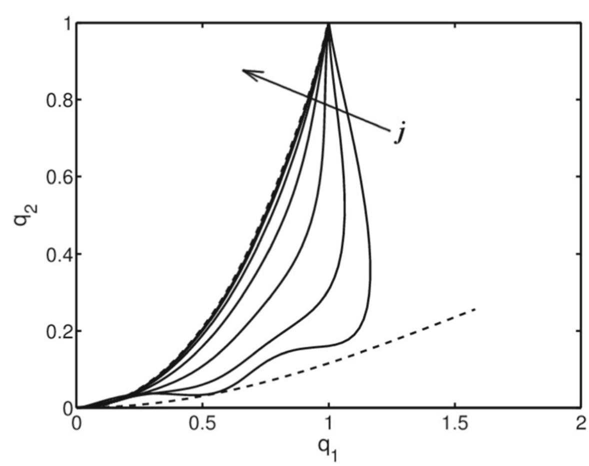

Fig. 7 Motion planning with dominating matrix $R(t)$

of matrix $Q(t)$ was set to 10. The results are shown in Figs. 2, 3, 4 and 5. The numbered dots represent the obstacles considered at the current step, while the circles depict the choice for a new obstacle, that will be placed in the next step of the simulation. In all cases the obstacles were success- fully avoided which supports the claim that the parametric Lagrangian Jacobian algorithm can be employed to such tasks.

The next set of computations illustrates the tendencies in shaping system's trajectory depending on the dominance of one of the matrices, either $Q(t)$ or $R(t)$. In the first case, the motion planning problem was solved for a series of matrices $Q_j(t) = 10^j Q(t)$, where $j = -1, -0.5, 0, 0.5, 1, 1, 5, 2$, $Q(t) = I_5$ and matrix $R(t) = I_2$. Then, the situation was reversed and the simulations were made for $Rj(t) = 10^j R(t)$, $R(t) = I_2$ and matrix $Q(t) = I_5$. All problems were solved for initial control $u_{\lambda_0}(t) = (-0.1, 0.8)^T$. Figures 6 and 7 present the results. In both figures the dashed line depicts the initial trajectory of the system, driven by $u_{\lambda_0}(\cdot)$. In case of Fig. 7 the additional bold dashed line represents the solution obtained by employment of the Jacobian pseudoinverse. It is easily seen that for sufficiently large matrices $Q(t)$ the motion planning algorithm indeed steers the system as close to the initial trajectory as possible. On the other hand, strengthening the matrix $R(t)$ results in prioritising the energy of controls, which is visible in Fig. 7, as the solutions approach the result obtained by the classical Moore-Penrose algorithm.

Finally, the employment of matrix $Q(t) = A^T(t)A(t)$ was compared against a standard case of $Q(t) = I_5$. The motion planning problem was solved for a series of matrices $Q_j(t) = 10^j Q(t)$, $j = -1, -0.5, 0, 0.5, 1, 1.5, 2$, and matrix $R(t) = B^T(t)B(t) = 2I_2$, with initial control $u_{\lambda_0}(t) = (-0.3, 0.9)^T$. The results are presented in Figs. 8 and 9. Moreover, for every solution the length of the final trajectory was computed, as shown in Table 1. The simulations support the claim that the choice of the matrix $Q(t) = A^T(t)A(t)$ imposes a bound on the length of the final trajectory obtained as a solution to the motion planning problem by the minimisation of the total length of all variations of the system's trajectory. Interestingly, such a behaviour is visible even for small gains of matrix Q(t), when the minimisation of the energy of controls is prioritised.

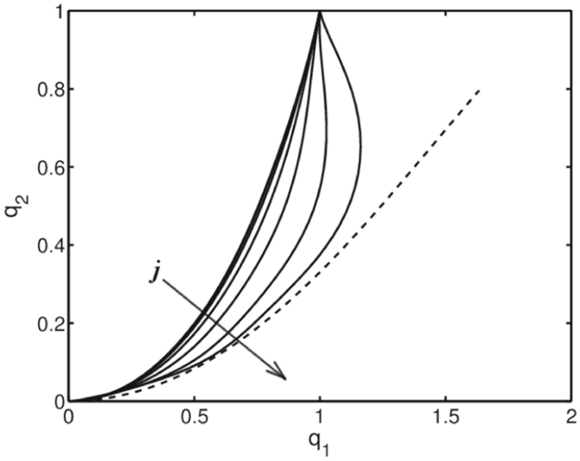

Fig. 9 Motion planning with $Q(t) = A^T(t)A(t)$

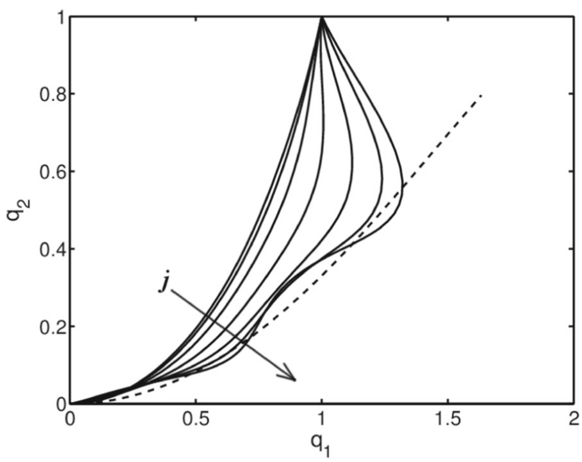

Fig. 8 Motion planning with $Q(t) = I_5$

Table 1 Length of the final trajectory $q_f(\cdot)$ depending on the choice of matrix $Q(t)$

	Length of $q_f(\cdot)$ for	
j	$Q(t) = A^T(t)A(t)$	$Q(t) = I_5$
-1.0	1.5042	1.5076
-0.5	1.5057	1.5162
0	1.5101	1.5428
0.5	1.5234	1.6151
1.0	1.5612	1.7505
1.5	1.6531	1.9121
2.0	1.8088	2.0499

7 Conclusion

The contribution of this paper lies in providing the parametric form of the Lagrangian Jacobian inverse and the corresponding motion planning algorithm for nonholonomic robotic systems. The inverse is derived from the solution of a Lagrange-type optimisation problem, that takes into consideration both system's controls and trajectory, and represents a natural generalisation of the Jacobian pseudoinverse. An analysis of the inverse reveals that by a skilful choice of the matrices Q and R in the Lagrangian objective function it is possible to shape the system's trajectory. Achieving a solution of the motion planning problem along with satisfying additional performance requirements has been demonstrated by numeric computations involving the kinematics of the rolling ball.

Acknowledgments The authors are grateful to anonymous reviewers whose remarks allowed them to improve the contents and the presentation of this paper, and suggested some topics for future research.

Appendix

A.1 Proof of Theorem 1

The parametric Lagrangian Jacobian inverse is derived from minimisation of the Lagrange-type objective function for the linearised system

$$\dot{\xi}_\lambda(t) = A_\lambda(t)\xi_\lambda(t) + B_\lambda(t)P(t)\mu,$$
$$\eta(t) = C_\lambda(t)\xi_\lambda(t), \quad \xi_\lambda(0) = 0, \tag{49}$$

i.e.

$$\min_\mu \int_0^T \left(\xi_\lambda^T(t)Q(t)\xi_\lambda(t) + \mu^T P^T(t)R(t)P(t)\mu\right) dt, \tag{50}$$

where $\lambda \in \mathrm{R}^s$ is fixed, $\mu \in \mathrm{R}^s$ varies, $Q(t) = Q^T(t) \geq 0$ and $R(t) = R^T(t) > 0$, on condition that

$$J_{q_0,T}(\lambda)\mu = C_\lambda(T)\xi_\lambda(T) = \eta. \tag{51}$$

To solve this problem the classical method of conditional optimisation is applied. Firstly, a Lagrange function for problem (50) is defined using the Jacobian (23) in the following way

$$\begin{aligned}\mathcal{L}(\mu,\alpha) = &\int_0^T \xi_\lambda^T(t)Q(t)\xi_\lambda(t)\,dt \\ &+\int_0^T \mu^T P(t)^T(t)R(t)P(t)\mu\,dt \\ &+\alpha^T C_\lambda(T)\int_0^T \Phi_\lambda(T,t)B_\lambda(t)P(t)\mu\,dt,\end{aligned}$$

where $\alpha \in \mathrm{R}^r$ denotes a vector of Lagrange multipliers and

$$\xi_\lambda(t) = \int_0^t \Phi_\lambda(t,s)B_\lambda(s)P(s)\,ds\,\mu \tag{52}$$

is a trajectory of Eq. 49. The derivative of the Lagrange function $\mathcal{L}(\mu,\alpha)$ for $w \in \mathrm{R}^s$ can be computed as

$$\begin{aligned}&\mathcal{DL}(\mu,\alpha)w = \frac{d}{d\beta}\Big|_{\beta=0}\mathcal{L}(\mu+\beta w,\alpha) = \\ &2\int_0^T \left[\xi_\lambda^T(t)Q(t)\int_0^t \Phi_\lambda(t,s)B_\lambda(s)P(s)\,ds\, + \right. \\ &\left.\mu^T P^T(t)R(t)P(t) + \frac{1}{2}\alpha^T C_\lambda(T)\Phi_\lambda(T,t)B_\lambda(t)P(t)\right] dt\,w\end{aligned}$$

The optimality condition, $\mathcal{DL}(\mu,\alpha)w = 0, \forall w \in \mathrm{R}^s$, enforces that

$$\begin{aligned}&\int_0^T \left[\xi_\lambda^T(t)Q(t)\int_0^t \Phi_\lambda(t,s)B_\lambda(s)P(s)\,ds\, + \right. \\ &\left.\mu^T P^T(t)R(t)P(t) + \frac{1}{2}\alpha^T C_\lambda(T)\Phi_\lambda(T,t)B_\lambda(t)P(t)\right] dt \\ &= 0\end{aligned} \tag{53}$$

Invoking (52), after transposition of Eq. 53 we obtain

$$\int_0^T \left(F_\lambda^T(t)Q(t)F_\lambda(t) + P^T(t)R(t)P(t)\right) dt\,\mu + \frac{1}{2}F_\lambda^T(T)C_\lambda^T\alpha = 0, \tag{54}$$

where

$$F_\lambda(t) = \int_0^t \Phi_\lambda(t,s)B_\lambda(s)P(s)\,ds. \tag{55}$$

Equation 54 can as well be expressed as

$$\mu = -\frac{1}{2} I_\lambda^{-1}(T) F_\lambda^T(T) C_\lambda^T(T)\alpha, \tag{56}$$

where

$$I_\lambda(t) = \int_0^T (F_\lambda^T(t)Q(t)F_\lambda(t) + P^T(t)R(t)P(t))\, dt = 0. \tag{57}$$

After considering Eq. 51 and combining it with Eq. 56, the vector of Lagrange multipliers can be derived

$$\alpha = -2\mathcal{M}_{q_0,T}^{-1}(\lambda)\eta,$$

where

$$\mathcal{M}_{q_0,T}(\lambda) = C_\lambda(T)F_\lambda(T)I_\lambda^{-1}(T)F_\lambda^T(T)C_\lambda^T(T)$$

is the mobility matrix of system (22). Finally, after the elimination of α from Eq. 56 we obtain the parametric form of Lagrange Jacobian inverse

$$J_{q_0,T}^{\mathcal{L}\#}(\lambda)\eta = \mu = I_\lambda^{-1}(T)F_\lambda^T(T)C_\lambda^T(T)\mathcal{M}_{q_0,T}^{-1}(\lambda)\eta.$$

It is easily seen that in order to compute the parametric Lagrange inverse additional rules for calculating $I_\lambda(T)$ and $F_\lambda(T)$ are advantageous. On account of

$$\frac{\partial \Phi_\lambda(t,s)}{\partial t} = A_\lambda(t)\Phi_\lambda(t,s), \qquad \Phi_\lambda(s,s) = I_n,$$

the time differentiation of Eq. 55 yields

$$\frac{dF_\lambda(t)}{dt} = B_\lambda(t)P(t) + A_\lambda(t)F_\lambda(t),$$

with $F_\lambda(0) = 0$. Analogically, matrix $I_\lambda(T)$ can be computed from

$$\frac{dI_\lambda(t)}{dt} = F_\lambda^T(t)Q(t)F_\lambda(t) + P^T(t)R(t)P(t),$$

for $I_\lambda(0) = 0$, what concludes the proof.

A.2 Proof of Lemma 1

Let $\lambda(\theta) = \lambda$. For the linearised system (49) and the Lagrange-type objective function (50) we set $\dot{\xi}_\lambda = \frac{d\xi_\lambda(t)}{dt}$. The square of its length can be computed as

$$||\dot{\xi}_\lambda||^2 = \dot{\xi}_\lambda^T\dot{\xi}_\lambda = ||A_\lambda(t)\xi_\lambda + B_\lambda(t)P(t)\mu||^2. \tag{58}$$

It is easily shown that

$$||A_\lambda(t)\xi_\lambda + B_\lambda(t)P(t)\mu||^2 \le 2(||A_\lambda(t)\xi_\lambda||^2 + ||B_\lambda(t)P(t)\mu||^2). \tag{59}$$

The right side of this inequality can be rewritten into

$$2(||A_\lambda(t)\xi_\lambda||^2 + ||B_\lambda(t)P(t)\mu||^2) = 2(\xi_\lambda^T A_\lambda^T(t)A_\lambda(t)\xi_\lambda + \mu^T P^T(t)B_\lambda^T(t)B_\lambda(t)P(t)\mu). \tag{60}$$

Let $Q(t) = A_\lambda^T(t)A_\lambda(t)$ and $R(t) = B_\lambda^T(t)B_\lambda(t)$. Then, combining the identity (58) with inequalities (59), (60), and then integrating both sides for $t \in [0,T]$ yields

$$\int_0^T ||\dot{\xi}_\lambda||^2\, dt \le 2\int_0^T (\xi_\lambda^T(t)Q(t)\xi_\lambda(t) + \mu^T P(t)R(t)P(t)\mu)dt.$$

Finally, an application of the Schwartz's inequality results in

$$\int_0^T ||\dot{\xi}_\lambda||\, dt \le \sqrt{T}\sqrt{\int_0^T ||\dot{\xi}_\lambda||^2\, dt} \le \sqrt{2T}\sqrt{\int_0^T (\xi_\lambda^T(t)Q(t)\xi_\lambda(t) + \mu^T P(t)R(t)P(t)\mu)},$$

concluding the proof.

References

1. Ważewski, T.: Sur l'évaluation du domaine d'existence des fonctions implicites réelles ou complexes. Ann. Soc. Pol. Math **20**, 81–120 (1947)
2. Sussmann, H.J.: A continuation method for nonholonomic path finding problems. In: Proc. 32nd IEEE CDC, 2718–2723 (1993)
3. Divelbiss, A.W., Seereeram, S., Wen, J.T.: Kinematic path planning for robots with holonomic and nonholonomic constraints, Essays on Mathematical Robotics, 127–150. Springer-Verlag, New York (1998)
4. Dulęba, I., Sąsiadek, J.Z.: Nonholonomic motion planning based on Newton algorithm with energy optimization. IEEE Trans. Contr. Syst. Technology **11**(3), 355–363 (2003)
5. Tchoń, K., Jakubiak, J.: Endogenous configuration space approach to mobile manipulators: a derivation and performance assessment of Jacobian inverse kinematics algorithms. Int. J. Control. **76**(9), 1387–1419 (2003)
6. Alouges, F., Chitour, Y., Long, R.: A motion planning algorithm for the rolling-body problem. IEEE Trans. Robotics **26**, 827–836 (2010)
7. Chitour, Y.: A continuation method for motion planning problems. ESAIM **12**, 139–168 (2006)
8. Chitour, Y., Sussmann, H.J.: Motion planning using the continuation method, Essays on Mathematical Robotics, 91–125. Springer-Verlag, New York (1998)
9. Sontag, E.D.: Mathematical Control Theory. Springer-Verlag, New York (1990)
10. Tchoń, K., Ratajczak, A., Góral, I.: Lagrangian Jacobian inverse for nonholonomic robotic systems. Nonlinear

Dynamics **82**, 1923–1932 (2015). 10.1007/s11071-015-2288-6
11. Tchoń, K., Góral, I., Ratajczak, A.: Jacobian motion planning for nonholonomic robots; the Lagrangian Jacobian algorithm. In: 10th international workshop on robot motion and control (RoMoCo), IEEE Xplore, 229–234 (2015). doi:10.1109/RoMoCo.2015.7219740
12. Tchoń, K., Góral, I.: Lagrangian Jacobian motion planning. In: Proc. IMA Conference on Mathematics of Robotics, St Annes College University of Oxford (2015)
13. Ratajczak, A., Tchoń, K.: Parametric and non-parametric Jacobian motion planning for non-holonomic robotic systems. J. Intell. Robotic Syst. **77**, 445–456 (2015). doi:10.1007/s10846-013-9880-0
14. Chirikjian, G.S., Kyatkin, A.B.: Engineering Applications of Noncommutative Analysis. CRC Press, Boca Raton (2001)
15. Galicki, M.: The planning of robotic optimal motions in the presence of obstacles. Int. J. Robot. Res. **17**(3), 248–259 (1998)

Ida Góral MSc, MEng, is a PhD student at the Electronics Faculty, Wrocław University of Technology, Poland. Her research interests include Jacobian algorithms of motion planning for non-holonomic robotic systems and medical robotic systems. I. Góral has been an author of 4 publications.

Krzysztof Tchoń PhD, DSc, is a professor of automation and robotics at the Electronics Faculty, Wrocław University of Technology, Poland. His research interests concentrate on mathematical aspects of control systems and robotics, specifically, on geometric control, singularities of manipulation and mobile robots, and motion planning and control of non-holonomic systems. Recently, he got involved in social robotics research. Over the last five years he was a leader of 15 robotics research projects. Professor K. Tchoń authored nearly 200 publications and promoted 13 PhD students. 5 more are working under his supervision, including the first author of this paper.