Robotica (2016) volume 34, pp. 423–448. © Cambridge University Press 2014
doi:10.1017/S0263574714001544

Optimal power loss motion planning in legged robots

Alain Segundo Potts†* and José Jaime da Cruz‡

†*Center of Engineering, Modeling and Applied Social Sciences, Federal University of ABC, São Paulo, Brazil*
‡*Telecommunications and Control Department, University of São Paulo, São Paulo, Brazil*

(Accepted May 20, 2014. First published online: August 19, 2014)

SUMMARY
An iterative algorithm to minimize energy loss in kinematic chains is proposed. This algorithm is designed to low level of control where variables such as terminal states, runtime, and physical and electrical parameters of the movement are given by higher levels of control. An original complex problem of optimization is transformed into a simple quadratic programming problem subject to linear constraints by discretizing all dynamic system variables. The whole system is then converted into a recursive matrix equation that is solved iteratively. A proof of convergence is suggested. The performance of the algorithm is illustrated by using it in the motion planning of a quadruped robot.

KEYWORDS: Optimal control; Quadratic programming; Robot control; Legged robots.

1. Introduction

In recent decades, there has been a rapid development in the research of autonomous robots. Many of these robots are inspired by biological organisms, which, in one way or the other, have certain special features. In general, living organisms unconsciously execute their movements with a minimum expenditure of energy.

From the point of view of robotics science, the problem of performing movements with the least expenditure of energy is known as the Optimal Free Motion Planning Problem (OFMPP).[8] Although the first papers dealing with OFMPP for robots appeared about 30 years ago,[13] it remains an active and attractive research area in robotics. This interest can be justified by two main factors. First, and from an economic point of view, there is a direct relationship between productivity and quality of implemented motions. Second, and from an engineering point of view, strategies or algorithms developed for solving OFMPP are potentially applicable to a broader class of problems arising in different disciplines: control of unmanned vehicles, graphic animation of digital actors, computational chemistry and biology, design of product prototypes and their manufacturing using CAD/CAM systems, and medical applications.[15]

In robotics literature, one can find many papers dealing with OFMPP for various kinematic structures (closed and open chain), different locomotion modes (mobile robots, walking machines), and several degrees of freedom (redundant or non-redundant). Regardless of the robot type, in order to plan its motion, it is necessary to model its behavior from geometric, kinematic, or dynamic points of view. The complexity of a formulated problem strongly depends on the model adopted and formulated constraints.

The OFMPP can be considered a classical optimal control problem. It is commonly resolved using specific numerical methods that can be classified into two categories, namely direct and indirect methods.[5,12,23] The indirect methods are based on the calculus of variations, or on the Maximum Principle of Pontryagin.[7] These generally lead to a nonlinear multi-point boundary value problem that must be solved using appropriate techniques, such as the multiple-shooting method, and require a good initial solution's guess. Unfortunately, to be able to employ such methods, the user must have good knowledge of the optimal control theory and a deep insight into the physical and mathematical

* Corresponding author. E-mail: alain_2do@yahoo.com

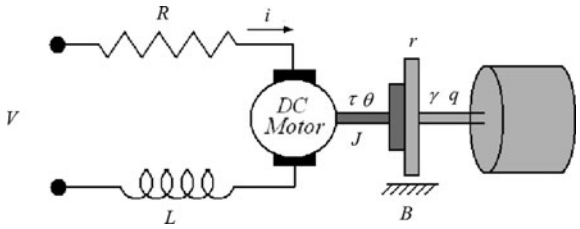

Fig. 1. Diagram of a DC motor.

natures of optimization problem. The direct methods are based on the transformation of the original problem into a nonlinear programming problem by discretizing some or all of the dynamic system variables (states, controls). Such methods have proven to be robust, simple, and need minimal user interaction to define the optimization problem. Because of the importance of solving these problems, many numerical algorithms and commercial software packages have been developed since the 1960s. However, most of the existing algorithms do not have adequate properties of numerical stability and are too slow computationally to solve optimal control problems for current multibody systems. In this sense, in recent years, authors such as Liu and Atkeson,[16] Zhong and Todorov,[26] and Majumdar and Tedrake[18] have developed interesting methods to optimize trajectories in legged robots using dynamic programming and other more sophisticated techniques of robust, nonlinear, and stochastic optimal control.

In mobile robots, the dynamics of walking machines involves special dynamic features that render these systems more intricate since they present a time-varying topology. This means that these systems include kinematic loops that open when a leg takes off and open chains that close when a leg touches the ground.[1] This fact implies that the number of degrees of freedom is time-varying.[19] Works on optimal motion planning for walking robots are few and relatively recent. This may be due to the complexity of the motion planning problem, and due to difficulties to get dynamic models for such structures. Authors highlighted the difficulty in computing the C-space since a system of nonlinear equations, inherent to closure constraints, must be solved at each iteration of the search process.[8]

The optimization criteria commonly employed in robotics are based on physical criteria involving energy, control effort, jerk, or time.[6,9,14,17,25] A minimum energy formulation may thus be an interesting approach for mobile robots, particularly in applications in which the battery weight is a critical issue. The convergence analysis of such a minimum energy problem is a relevant issue to both design and operation time.

An iterative algorithm is proposed in this paper to minimize power loss in mobile robots. Although the algorithm is applied to the optimization of power loss in the motors driving active joints, it can be applied to a wide range of kinematic chains. The algorithm is based on an iterative process that solves a quadratic programming problem (QPP) obtained from the inverse dynamic model of the system. The power of this approach is that the nonlinear hard problem of optimization is transformed into a sequence of simple QPP that can be solved efficiently. The algorithm is tested in the motion planning of gait of a quadruped robot.

2. Formulation

The control problem to be solved consists in determining the time histories of armature voltages in the motor joints of a kinematic chain necessary to produce a desired motion with minimal power loss.

2.1. Dynamics of DC motor

The actuator chosen was a DC motor (Fig. 1). This kind of motor is divided into three subsystems: magnetic, electrical, and mechanical.

The magnetic subsystem can be described by the following equations:

$$\chi_m(t) = K_e i_e(t), \tag{1}$$

$$\tau(t) = K_p i(t) \chi_m(t), \tag{2}$$

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

where $i_e(t)$ is the excitation current, $i(t)$ is the armature current, $\chi_m(t)$ is the flow generated by the excitation current K_e and K_p constants, and τ is the motor torque.

As the excitation current may be assumed as constant, Eq. (2) is reformulated as:

$$\tau(t) = Ki(t), \tag{3}$$

where K is the electromotive constant.

In the same way, as the magnetic flow $\chi_m(t)$ is assumed constant, there is a counter-electromotive force (CEMF) that induces a voltage proportional to the angular velocity of rotor:

$$\upsilon(t) = K_b\dot{\theta}(t), \tag{4}$$

where $\theta(t)$ is the angular position of the rotor, and K_b is a CEMF constant.

In turn, the electrical subsystem is defined by the loop equation:

$$v(t) = Ri(t) + L\frac{di(t)}{dt} + \upsilon(t). \tag{5}$$

Here, v, R, L, and i are the voltage, resistance, inductance, and current of the armature motor respectively.

Finally, the mechanical subsystem is represented by the following equation:

$$\tau(t) = J\ddot{\theta}(t) + B\dot{\theta}(t) + r\gamma(t), \tag{6}$$

where r is the gear reduction factor, J is the inertia of the rotor, B is the viscous damping of the motor, and γ is the load torque.

As the time-constant of the mechanical subsystem (6) is much larger than the time-constant of electrical subsystem (5), inductance L may be discarded without impairing model accuracy.

Then, using Eqs. (2), (5), and (6), the model is reformulated as:

$$J\ddot{\theta}(t) + B\dot{\theta}(t) = Ki(t) - r\gamma(t), \tag{7}$$

$$v(t) = Ri(t) + K_b\dot{\theta}(t), \tag{8}$$

and

$$q(t) = r\theta(t), \tag{9}$$

where $q(t)$ is the angular position of the load.

2.2. Independent joint control and control by torque

Complex mechatronic systems usually comprises one or more kinematics chains. The use of DC motors to drive joints in a kinematic chain is commonly a simple way to control that. Because of the wide use of these systems, there are different approaches to control them efficiently. In this sense, the classical independent joint control philosophy is considered herein. In this case, each axis of kinematic chain is controlled as a single input/single output (SISO) system. Any coupling effects due to the motion of other links are either ignored or treated as disturbances. The key point of the independent joint approach is the fact that the torque that arises from the motion of other links is considered as a perturbation reduced by the gear reduction factor r, which is generally much less than 1. Thus, the gear ratio reduces the coupling nonlinearities effects present in the limbs dynamics.[22]

Let us assume that we have a mechatronic system with L kinematic chains, each of these with M joints driven by DC motors. The subscript l ($l = 1, \ldots, L$) is introduced to indicate the lth kinematic chain of the system, and the sub-subscript i ($i = 1, \ldots, M$) is introduced to indicate its ith joint.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

Defining:

$$\mathbf{x}_{l_i} = \begin{bmatrix} \theta_{l_i} \\ \dot{\theta}_{l_i} \end{bmatrix}, \tag{10}$$

the state-space model that describes the dynamics of the DC motor located at the ith joint ($1 \leq i \leq M$) of the lth kinematic chain ($1 \leq l \leq L$) is given by:

$$\begin{cases} \dot{\mathbf{x}}_{l_i} = \mathbf{A}_{l_i}\mathbf{x}_{l_i} + \mathbf{B}_{l_i}(\tau_{l_i} - r\gamma_{l_i}) \\ y_{l_i} = \mathbf{C}_{l_i}\mathbf{x}_{l_i} \end{cases}, \tag{11}$$

where:

$$\mathbf{A}_{l_i} = \begin{bmatrix} 0 & 1 \\ 0 & -\dfrac{B_{l_i}}{J_{l_i}} \end{bmatrix}, \tag{12}$$

$$\mathbf{B}_{l_i} = \begin{bmatrix} 0 \\ \dfrac{1}{J_{l_i}} \end{bmatrix}, \tag{13}$$

and

$$\mathbf{C}_{l_i} = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{14}$$

Since the objective is to minimize power loss in a system formed by kinematic chains, the performance index adopted is the Joule loss in the armature of motors:

$$\mathcal{E} = \sum_{l,i} \varepsilon_{l_i}, \tag{15}$$

where:

$$\varepsilon_{l_i} = \min_{\tau_{l_i}} \int_0^{t_f} P_{l_i} dt, \tag{16}$$

subject to:

$$\mathbf{x}_{l_i}(t_f) = \tilde{\mathbf{x}}_{l_i} - \mathbf{x}_{l_i}(0), \tag{17}$$

$$|i_{l_i}(t)| \leq I_{\max} \ \forall t \in [0, t_f], \tag{18}$$

$$|v_{l_i}(t)| \leq V_{\max} \ \forall t \in [0, t_f], \tag{19}$$

$$\check{\theta}_{l_i} \leq \theta_{l_i}(t) \leq \hat{\theta}_{l_i} \ \forall t \in [0, t_f], \tag{20}$$

where P_{l_i} represents the instantaneous power loss of motor at the ith joint of the lth kinematic chain,

$$P_{l_i} = R_{l_i} i_{l_i}^2(t), \tag{21}$$

and $i_{l_i}(t)$ could be obtained from (3).

The values of t_f, which is the time required to move the system from its initial to its final position, the maximum armature current $I_{\max}$, the maximum armature voltage $V_{\max}$, the minimum and maximum angular positions $\check{\theta}_{l_i}$ and $\hat{\theta}_{l_i}$ respectively, and the vector of final angular position and velocity of the rotor given by vetor $\tilde{\mathbf{x}}_{l_i}$ are used to constrain the problem.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

2.3. Discretization

The first step to solve the above problem is its discretization in time. For this, the time-step is defined as:

$$\Delta t = \frac{t_f}{N} = t_j - t_{j-1}, \tag{22}$$

for $1 < j \leq N$, and N is sufficiently large in order that a smooth solution can be obtained. It is assumed that $\tau_{l_i}(t)$ is a stepwise constant function and $\tau_{l_i}(t_j)$ is used to denote the value of $\tau_{l_i}(t)$ $\forall t, \in [t_{j-1}, t_j)$.

Thus, the solution of (11) is:

$$\mathbf{x}_{l_i}(t_j) = e^{\mathbf{A}_{l_i} t_j} \mathbf{x}_{l_i}(0) + \mathbf{H}_{l_i}(t_j)(\mathbf{T}_{l_i} - r\boldsymbol{\Gamma}_{l_i}), \tag{23}$$

where:

$$\mathbf{H}_{l_i}(t_j) = e^{\mathbf{A}_{l_i} t_j} \left[\left(\int_0^{t_1} e^{-\mathbf{A}_{l_i} t} dt \right) \mathbf{B}_{l_i}, \ldots, \left(\int_{t_{N-1}}^{t_N} e^{-\mathbf{A}_{l_i} t} dt \right) \mathbf{B}_{l_i} \right] \mathbf{E}_{l_i}(t_j), \tag{24}$$

$$\mathbf{T}_{l_i} = \begin{bmatrix} \tau_{l_i}(t_1) & \tau_{l_i}(t_2) & \ldots & \tau_{l_i}(t_N) \end{bmatrix}^T, \tag{25}$$

$$\boldsymbol{\Gamma}_{l_i} = \begin{bmatrix} \gamma_{l_i}(t_1) & \gamma_{l_i}(t_2) & \ldots & \gamma_{l_i}(t_N) \end{bmatrix}^T, \tag{26}$$

and

$$\mathbf{E}_{l_i}(t_j) = \begin{bmatrix} e_1 & 0 & 0 & 0 \\ 0 & e_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & e_N \end{bmatrix}, \tag{27}$$

where

$$e_z = \begin{cases} 1, & 1 \leq z \leq j \\ 0, & j < z \leq N \end{cases}. \tag{28}$$

The performance index of Eq. (16) can be rewritten as:

$$\varepsilon_{l_i} = \min_{\mathbf{T}_{l_i}} \frac{1}{2} \mathbf{T}_{l_i}^T \mathbf{Q}_{l_i} \mathbf{T}_{l_i}, \tag{29}$$

subject to:

$$\mathbf{x}_{l_i}(t_N) = \tilde{\mathbf{x}}_{l_i} - \mathbf{x}_{l_i}(0), \tag{30}$$

$$|i_{l_i}(t_j)| \leq I_{\max} \; \forall j = 1, 2, \ldots, N, \tag{31}$$

$$|v_{l_i}(t_j)| \leq V_{\max} \; \forall j = 1, 2, \ldots, N, \tag{32}$$

$$\check{\theta}_{l_i} \leq \theta_{l_i}(t_j) \leq \hat{\theta}_{l_i} \; \forall j = 1, 2, \ldots, N, \tag{33}$$

where

$$\mathbf{Q}_{l_i} = 2 \frac{R_{l_i}}{K_{l_i}^2} \mathbf{I}. \tag{34}$$

Note that $\mathbf{I}$ is a $N \times N$ identity matrix and $\mathbf{Q}_{l_i}$ depends only on the motor parameters.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

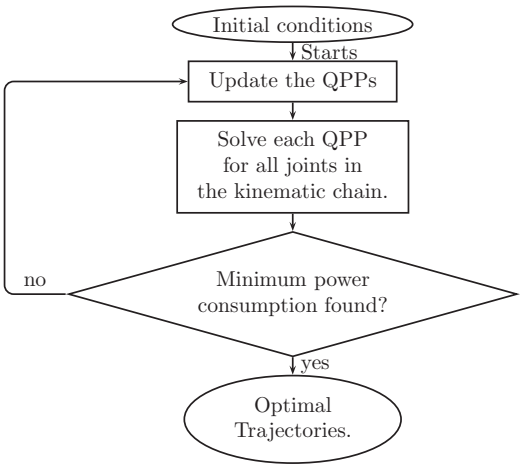

Fig. 2. Flowchart of the algorithm.

3. Algorithm

The idea behind the algorithm is trying to solve the QPP iteratively for active joints using the advantages of quadratic programming and the fact that the strategy used is independent joint control. Then, by an iterative process, the effects of coupling between the joints are updated at the beginning of each iteration and used to solve new QPP in the next iteration. In each iteration, the algorithm is expected to converge to a global minimum point (Fig. 2).

Before continuing, let us make three definitions that will be used hereafter.

Definition 1

For any vector $\boldsymbol{\chi}$ *the notation* $\chi_{l_i}^k(t_j)$ *means that the elements correspond to the* i*th joint* $\forall i = 1, 2, \ldots, M$ *of the* l*th kinematic chain* $\forall l = 1, 2, \ldots, L$ *in the time interval* $[t_j, t_{j+1}) \forall j = 1, 2, \ldots, N$ *and in the iteration* $k : k \in \mathbb{N}$.

Definition 2

From the vector $\boldsymbol{\Theta}_{l_i}^k = \left[\, \theta_{l_i}^k(t_1), \theta_{l_i}^k(t_2), \ldots, \theta_{l_i}^k(t_N) \,\right]^T$ *the new vector* $\boldsymbol{\Theta}_l^k \in \mathcal{B}$ *is defined as:*

$$\boldsymbol{\Theta}_l^k = \begin{bmatrix} \boldsymbol{\Theta}_{l_1}^k \\ \vdots \\ \boldsymbol{\Theta}_{l_M}^k \end{bmatrix}, \tag{35}$$

where $\mathcal{B}$ *is defined below.*

Definition 3

$\mathcal{B}$ *is a subset of Banach Space, in the form of a ball with radius* θ_r*, and center in* $\bar{\boldsymbol{\Theta}}_l \in \mathbb{R}^{MN}$*:*

$$(\bar{\boldsymbol{\Theta}}, \theta_r) = \{\boldsymbol{\Theta}_l^k; \|\boldsymbol{\Theta}_l^k - \bar{\boldsymbol{\Theta}}\| \leq \theta_r\}, \tag{36}$$

where $\theta_r = \max \boldsymbol{\Theta}_{l_i};\ i = 1, \ldots, M$.

3.1. Structure of algorithm

In general, the process is started by assuming an initial load torque vector: $\boldsymbol{\Gamma}_l^0 = \mathbf{0}$, where $\boldsymbol{\Gamma}_l$ represents the coupling effect among links in the kinematic chain l and is computed by the Euler–Lagrange equation:

$$\boldsymbol{\Gamma}_l = \mathbf{M}_l(\mathbf{q}_l)\ddot{\mathbf{q}}_l + \boldsymbol{\Phi}_l(\mathbf{q}_l)[\dot{\mathbf{q}}_{x_l}\dot{\mathbf{q}}_{y_l}] + \boldsymbol{\Psi}_l(\mathbf{q}_l)[\dot{\boldsymbol{q}}_l^2] + \mathbf{G}_l(\mathbf{q}_l), \tag{37}$$

where $\mathbf{q}_l$ is the vector composed by the joint angles $\mathbf{q}_{l_i}$ for all $i = 1, \ldots, M$, $\mathbf{M}_l(\mathbf{q}_l) \in \mathbb{R}^{M\times M}$, $\mathbf{M}_l(\mathbf{q}_l) = \mathbf{M}_l(\mathbf{q}_l)^T > \mathbf{0}$ is the matrix of mass of the kinematic chain, $\boldsymbol{\Phi}_l \in \mathbb{R}^{M\times M}$ is the Coriolis/centripetal coefficients matrix, $[\dot{\boldsymbol{q}}_{x_l}\dot{\boldsymbol{q}}_{y_l}]$ is a vector of products $\forall\ x_l \neq y_l$, $\boldsymbol{\Psi}_l \in \mathbb{R}^{M\times M}$ is the friction coefficients matrix, and $\mathbf{G}_l \in \mathbb{R}^M$ is the gravity matrix.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

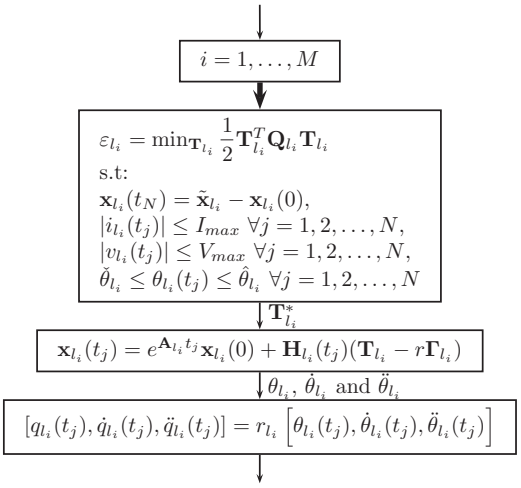

Fig. 3. Solving the QPPs.

From the initial vector $\mathbf{\Gamma}_l^0$, different QPPs for all active joints in the kinematic chain l are solved. Now assume that in the next step ($k = 1$) we obtain an optimal control law $\mathbf{T}_l^{*^1}$ and this one is then used as a control input of system (11) to generate histories $\theta_{l_i}^1(t_j)$, $\dot{\theta}_{l_i}^1(t_j)$, and $\ddot{\theta}_{l_i}^1(t_j)$ for all $i = 1, \ldots, M$ and $j = 1, 2, \ldots, N$. Once these values are multiplied by the gear reduction ratio ($r_{l_i} \ll 1$), the positions, velocities, and angular accelerations of each joint are obtained. See Fig. 3.

With the calculated values $q_{l_i}^k(t_j)$, $\dot{q}_{l_i}^k(t_j)$, and $\ddot{q}_{l_i}^k(t_j)$ it is possible to derive the new load torque vector by solving the Euler–Lagrange equation (37) that relates all load torques in the kinematic chain l. Thus, at the beginning of each iteration the load torque vector is updated and remains constant until the beginning of next iteration. Then, with this new load torque vector ($\mathbf{\Gamma}_l^k$), it is possible to compute next optimal torque vector $\mathbf{T}_l^{k+1^*}$, and a new iteration begins. Finally, for the algorithm, the difference of the same vector can be used as stopping criterion in two consecutive iterations. The vector used as stopping criterion should be, for example, the optimal torque vector $\mathbf{T}_l^*$, the load torque vector $\mathbf{\Gamma}_l$, or the motor angular position vector $\mathbf{\Theta}_l$. In this case we compare the difference of the motor angular position vector $\mathbf{\Theta}_l$ by means of Eq. (38). If the result of the infinite norm of the difference between these vectors is greater than the tolerance $\xi > 0$ chosen, a new iteration of the algorithm is started (Fig. 4) and the optimal trajectory is found,

$$\|\mathbf{\Theta}_l^k - \mathbf{\Theta}_l^{k-1}\|_\infty \leq \xi. \tag{38}$$

Note that the algorithm proposed may be represented by a loop. The infinity norm was used here to guarantee that the largest value of the difference between the corresponding elements of $\mathbf{\Theta}_l^k$ and $\mathbf{\Theta}_l^{k+1}$ be smaller than the value of the stopping parameters ξ. Thus, we obtained an iterative cycle that must converge to a global minimum point, as will be shown in the following section.

3.2. Proof of convergence

To prove the convergence of the algorithm, first the existence of a feasible solution for each QPP will be discussed and then the convergence of the external loop will be proved.

3.2.1. Optimal solution for QPP. The internal part of the algorithm is formed by two nested loops that will solve QPP for all the joints in the kinematic chain l. In this sense, it is necessary to prove that all the stated QPPs have a single solution. For that, one must prove that the constraints define a non-empty set, and that the quadratic matrix $\mathbf{Q}_{l_i}$ is positive definite in the null space of active constraints.

According to (34), matrix $\mathbf{Q}_{l_i}$ is a positive multiple of identity matrix, then condition $\mathbf{Q}_{l_i} \succ 0$ is always satisfied.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

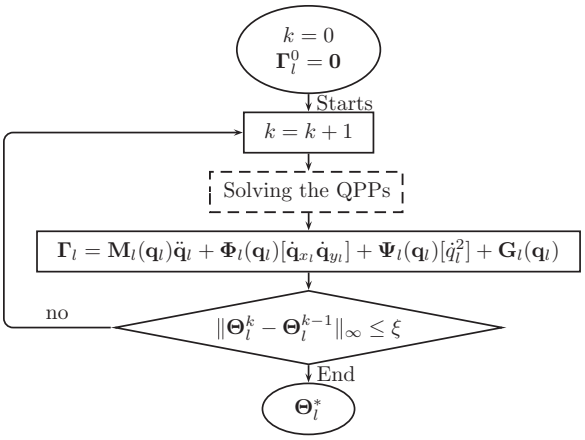

Fig. 4. External loop.

The next step consists in verifying the existence of some feasible vector $\mathbf{T}_l^*$ that satisfies the constraints of QPP. One way to test this is to check the Karush–Kuhn–Tucker (KKT) conditions for constrained problems.

Let us reformulate the QPP defined by Eqs. (29) to (33) as:

$$\varepsilon_{l_i} = \min_{\mathbf{T}_{l_i}} \frac{1}{2}\mathbf{T}_{l_i}^T \mathbf{Q}_{l_i} \mathbf{T}_{l_i}, \tag{39}$$

subject to:

$$\mathbf{H}_{l_i}(t_N)\mathbf{T}_{l_i} = \mathbf{F}_{l_i}, \tag{40}$$

$$\Lambda_{l_i}\mathbf{T}_{l_i} \leq \Omega_{l_i}, \tag{41}$$

where $\mathbf{F}_{l_i}$, Λ_{l_i}, and Ω_{l_i} are matrices that depend of motor parameters, starting point, and load torque vector $\Gamma_{l_i}^{k-1}$ found in the last iteration. A detailed derivation of these matrices is found in Appendix A. It is assumed that $\Gamma_{l_i}^{k-1}$ is constant when the QPP is solved but is updated at the beginning of each iteration.

Now if there are vectors $\boldsymbol{\mu}_{l_i}^* = [\,\mu_{l_{i_1}} \quad \mu_{l_{i_2}} \quad \cdots \quad \mu_{l_{i_{4N}}}\,]$, $\lambda_{l_i}^* = [\,\lambda_{l_{i_1}} \quad \lambda_{l_{i_2}}\,]$, and, in addition, the following conditions are satisfied:

$$\frac{\partial \mathfrak{L}_{l_i}(T_{l_i}^*, \lambda_{l_i}^*, \mu_{l_i}^*)}{\partial T_{l_i}} = 0, \tag{42}$$

$$\frac{\partial \mathfrak{L}_{l_i}(T_{l_i}^*, \lambda_{l_i}^*, \mu_{l_i}^*)}{\partial \lambda_{l_i}} = 0, \tag{43}$$

$$\boldsymbol{\mu}_{l_i}^* \geq \mathbf{0}, \tag{44}$$

$$\mu_{l_i}^* \mathbf{W}_{i_j}(\mathbf{T}_{l_i}^*) = 0, \tag{45}$$

$$\mathbf{W}_{l_i}(\mathbf{T}_{l_i}^*) \leq \mathbf{0}, \tag{46}$$

where

$$\mathfrak{L}_{l_i}(T_{l_i}^*, \lambda_{l_i}^*, \mu_{l_i}^*) = \frac{1}{2}\mathbf{T}_{l_i}^T \mathbf{Q}_{l_i} \mathbf{T}_{l_i} + \boldsymbol{\lambda}_{l_i}^* \mathbf{X}_{l_i}(\mathbf{T}_{l_i}) + \mu_{l_i}^* \mathbf{W}_{l_i}(\mathbf{T}_{l_i}), \tag{47}$$

$$\mathbf{X}_{l_i}(\mathbf{T}_{l_i}) = \mathbf{H}_{l_i}(t_N)\mathbf{T}_{l_i} - \mathbf{F}_{l_i}, \tag{48}$$

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

and

$$\mathbf{W}_{l_i}(\mathbf{T}_{l_i}) = \Lambda_{l_i}\mathbf{T}_{l_i} - \Omega_{l_i}, \tag{49}$$

then there exists a point of local minimum $\mathbf{T}^*_{l_i}$ that minimizes cost function (29).

The general solution to the problem formulated by Eqs. (43) to (46) can be demonstrated as:

$$\mathbf{T}^{*^T}_{l_i} = (-\lambda^*_{l_i}\mathbf{H}_{l_i}(t_N) - \mu^{*^{\mathcal{S}}}_{l_i}\mathbf{\Lambda}^{\mathcal{S}}_{l_i})\mathbf{Q}^{-1}_{l_i}, \tag{50}$$

where $\mathbf{\Lambda}^{\mathcal{S}}_{l_i}$ is a matrix with rows formed by the coefficients of the s active inequalities constraint for $s \in \mathcal{S}$ and where $\mathcal{S}$ is the set of indices of active constrains at current feasible solution $\mathbf{T}_{l_i}$.

If no active inequality constraints $\mu^*_{l_i} = \mathbf{0}$, and $\lambda^*_{l_i}$ is given as:

$$\boldsymbol{\lambda}^*_{l_i} = \frac{-\mathbf{F}^T_{l_i}\mathbf{Q}_{l_i}}{\mathbf{H}_{l_i}(t_N)\mathbf{H}^T_{l_i}(t_N)}, \tag{51}$$

where matrices $\mathbf{Q}_{l_i}$ and $\mathbf{H}_{l_i}(t_N)\mathbf{H}_{l_i}(t_N)^T$ must be nonsingular.

3.2.2. Convergence of external loop. The next step is to prove the convergence of algorithm's external loop. For that, the solution obtained in (50) is used to solve (11). Thus, the position, velocity, and acceleration of motors are calculated, for example, by:

$$\theta^k_{l_i}(t_j) = \mathbf{C}_{l_i}\left(e^{\mathbf{A}_{l_i}t_j}\mathbf{x}_{l_i}(0) + \mathbf{H}_{l_i}(t_j)(\mathbf{T}^{*^k}_{l_i} - r_{l_i}\mathbf{\Gamma}^{k-1}_{l_i})\right), \tag{52}$$

$$\dot{\theta}^k_{l_i}(t_j) = [0\ 1]\left(e^{\mathbf{A}_{l_i}t_j}\mathbf{x}_{l_i}(0) + \mathbf{H}_{l_i}(t_j)(\mathbf{T}^{*^k}_{l_i} - r_{l_i}\mathbf{\Gamma}^{k-1}_{l_i})\right), \tag{53}$$

$$\ddot{\theta}^k_{l_i}(t_j) = \frac{K_{l_i}}{J_{l_i}} - \frac{B_{l_i}}{J_{l_i}}\dot{\theta}^k_{l_i}(t_j). \tag{54}$$

Multiplying these equations by the gear reduction ratio r_{l_i} gives the kinematics of the joint:

$$\left[q^k_{l_i}(t_j), \dot{q}^k_{l_i}(t_j), \ddot{q}^k_{l_i}(t_j)\right] = r_{l_i}\left[\theta^k_{l_i}(t_j), \dot{\theta}^k_{l_i}(t_j), \ddot{\theta}^k_{l_i}(t_j)\right]. \tag{55}$$

After solving the internal loop for all $i \in [1, M]$ and $j \in [1, N]$, the values $\mathbf{q}^k_l$, $\dot{\boldsymbol{q}}^k_l$, and $\ddot{\boldsymbol{q}}^k_l$ obtained in (55) are applied in the Euler–Lagrange equation (37) to update the load torque vector $\mathbf{\Gamma}_l$. As the optimal control law $\mathbf{T}^{*^k}_l$ is a function of the load torque vector $\mathbf{\Gamma}^{k-1}_l$, Eq. (37) may be written as a recurrence law that depends only on the torques of the previous step:

$$\ldots \mathbf{\Gamma}^{k-1}_l \xrightarrow{\text{Eq. (50)}} \mathbf{T}^{*^k}_l \xrightarrow{\text{Eq. (11)}} \mathbf{\Theta}^k_l \xrightarrow{\text{Eq. (37)}} \mathbf{\Gamma}^k_l \ldots \tag{56}$$

Note that from recurrence law (56), it is possible to formulate three recursive equations: $\mathbf{\Theta}^{k+1}_l = f(\mathbf{\Theta}^k_l)$, $\mathbf{T}^{*^{k+1}}_l = f(\mathbf{T}^{*^k}_l)$, and $\mathbf{\Gamma}^{k+1}_l = f(\mathbf{\Gamma}^k_l)$. In addition, the iterative process should be initialized by vectors $\mathbf{\Gamma}^0_l$, $\mathbf{T}^{*^0}_l$, or $\mathbf{\Theta}^0_l$. As our interest is to find an optimal trajectory, we use the recursive equation:

$$\mathbf{\Theta}^{k+1}_l = f\left(\mathbf{\Theta}^k_l\right), \quad k = 1, 2, 3, \ldots, \tag{57}$$

where $f : \mathbb{R}^{MN} \to \mathbb{R}^{MN}$ and the dimension MN is given by the product of joint numbers (M) and time intervals of discretization (N).

In accordance with the Theorem on Contracting Maps,[10] if the following conditions are satisfied:

(*i*) Function $f(\mathbf{\Theta}_l)$ is defined and continuous in $\mathcal{B}$;
(*ii*) For each $\mathbf{\Theta}_l \in \mathcal{B}$, point $f(\mathbf{\Theta}_l)$ also lies in $\mathcal{B}$;

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

(*iii*) There is a constant $\alpha < 1$ such that for any two points $\mathbf{\Theta}_l^x, \mathbf{\Theta}_l^y \in \mathcal{B}$, where $x, y \in \mathbb{N}$, the following inequality holds:

$$\| f(\mathbf{\Theta}_l^x) - f(\mathbf{\Theta}_l^y) \| \leq \alpha \| \mathbf{\Theta}_l^x - \mathbf{\Theta}_l^y \|, \tag{58}$$

then the following statements are true:

(a) Equation (57) has precisely one solution $\mathbf{\Theta}_l^* \in \mathcal{B}$.
(b) For any choice of $\mathbf{\Theta}_l^0 \in \mathcal{B}$, the sequence $\mathbf{\Theta}_l^k \; \forall k = 1, 2, 3, \ldots$ given by (57) is defined and converges to a global minimum $\mathbf{\Theta}_l^*$.
(c) For any $k = 1, 2, \ldots$, the following inequality holds:

$$\| \mathbf{\Theta}_l^k - \mathbf{\Theta}_l^* \| \leq \frac{\alpha^k}{1 - \alpha} \| \mathbf{\Theta}_l^x - \mathbf{\Theta}_l^{x-1} \|. \tag{59}$$

Proof:

(*i*) The first condition of the theorem is satisfied since Eqs. (11), (37), and (50) are continuous. If there is a feasible solution for each QPP in the kinematic chain l, it is ensured that there is a trajectory inside $\mathcal{B}$ that satisfies constraints (41) and (41).
(*ii*) The second condition is satisfied since each QPP is subject to constraint (33) that limits the angular trajectory of the joint; then if there is a feasible solution for QPP, this condition is guaranteed.
(*iii*) The third condition of the theorem is called the Lipschitz condition, and it expresses the fact that mapping $\mathbf{\Theta}_l \rightarrow f(\mathbf{\Theta}_l)$ decreases the distance between any two points in $\mathcal{B}$ by at least factor α. Let function $f(\mathbf{\Theta}_l)$ have continuous partial derivatives in region $\mathcal{B}$ defined above. Then the following equality holds:

$$\alpha = \max_{\mathbf{\Theta}_l \in \mathcal{B}} \| \mathfrak{J}_f \|_F, \tag{60}$$

where $\mathfrak{J}_f$ is the Jacobian matrix of $f(\mathbf{\Theta}_l)$, and notation $\| \bullet \|_F$ represents the Frobenius norm of matrix $\bullet$ according to this application of the theorem.[10]

In a real situation, ensure that $\alpha \leq 1$ from (60) is hard. Matrix $\mathfrak{J}_f$ depends on the physical parameters of kinematic chain, QPPs' constraints, and the gear reduction factor, which has to be much smaller than 1 (see Appendix B for a more detailed explanation). A practical way to validate (60) may be to evaluate the Jacobian matrix at the boundary of $\mathcal{B}$.

Finally, it is worth clarifying that the Theorem of Contracting Maps provides only sufficient conditions to ensure the convergence of the algorithm.

4. Results

The results presented in the previous sections are illustrated by one example in this section. The algorithm will be applied to minimize the loss of energy caused by the armature current in DC motors used to move a quadruped robot.

Kamambaré I[1] is a symmetrical bioinspired quadruped robot. It was designed for climbing vertical objects such as trees, power poles, bridges, etc.[4] It has four legs, each of these with four revolute joints. See Figs. 5 and 6 and Table I for more information. At the end of each leg, there is a gripper. All the joints are powered by DC Servo motors Electro-Craft model E-576.

Although Kamambaré I is being designed to climb vertical surfaces, the tests developed are performed based on simple trot gaits. In this type of gait, the diagonal legs move in tandem. While a pair of legs are fixed to the supporting surface and pushes the robot forward, the other pair is in the air, seeking a new foothold (see Fig. 7).

[1] Kamambaré means gecko in the Tupi-Guarani language.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

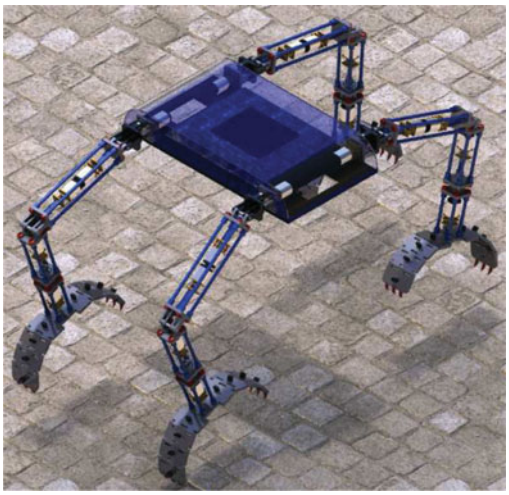

Fig. 5. Kamambaré I robot.

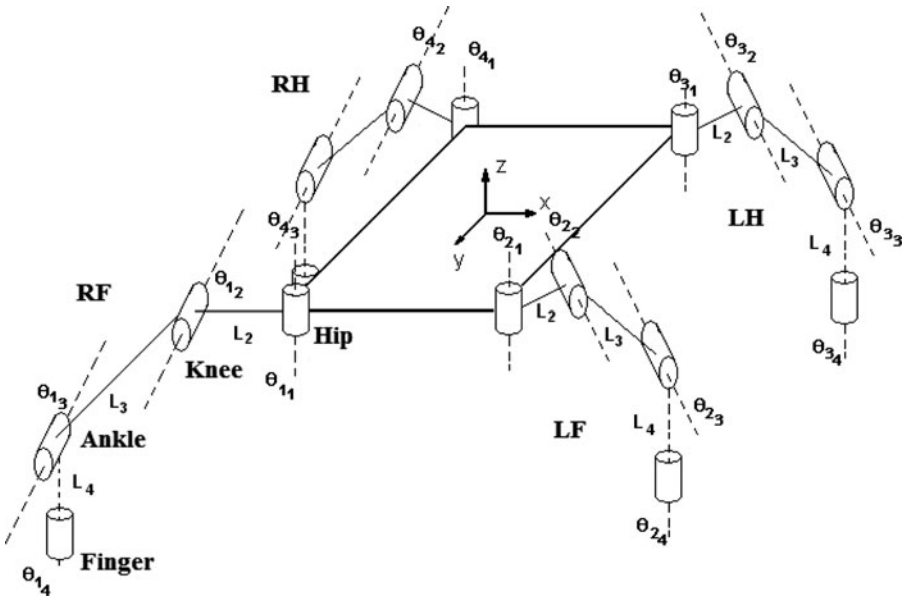

Fig. 6. Mechanical structure of Kamambaré I. The legs were named as follows: Left hind (LF), left front (LF), right front (RF), and right hind (RH).

Table I. Physical parameters of Kamambaré I robot.

Link	Length (m)	Weight (kg)	Moment of inertia (kgm^2)
4	0.2	0.2	17.6×10^{-5}
3	0.3	0.25	130.83×10^{-5}
2	0.1	0.06	1.625×10^{-5}
Body	0.42×0.42	5.25	2932.45×10^{-5}

The dynamics of walking machines involves special features that render these systems more elaborate from the dynamics viewpoint, for they present a time-varying topology. This means that these systems include kinematic loops that open when a leg takes off, and open chains that close when a leg touches the ground.[1] In addition, other effects such as the friction force in the joints should be taken into account in the dynamics model of the robot.[21]

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

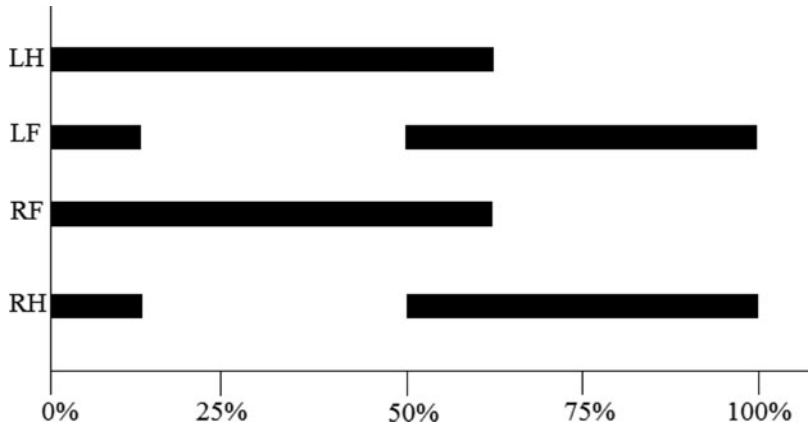

Fig. 7. Gait graphs for the trot. Dark areas indicate times of contact; the bottom axis is the percentage of the cycle.[11]

In accordance with the gait shown in Fig. 7 and to test the algorithm in a simple situation, two scenarios are treated separately: First, when two legs are fixed to the ground and the robot body is moving forward; and second, when the two other legs are in the air looking for a new gripping point. Hereafter, these two situations will be named body movement and leg movement respectively. Once the application of the algorithm is slightly different in both movement scenarios, its performance in each case is discussed in next sections.

4.1. Classes of joints

The joints may be grouped in the following classes: active/passive or independent/dependent.[20] The set of active joints is denoted by $\mathcal{A}$, whereas the set of passive joints is represented by $\mathcal{P}$. A joint is said to be active when there is an effective control law that acts on the DC motor that controls it; in the absence of this control law, the joint is called passive. Note that the set of passive joints is not part of the optimization search space.

In addition, the classification dependent $\mathcal{D}$ and independent $\mathcal{I}$ for the joints are a consequence of geometric constraints imposed on the mechanism. For example, if for the solution of an inverse kinematic problem it is necessary to solve a system in which there are more unknown variables than equations, then the number of unknowns that exceed the amount of equations will correspond to the group of independent joints $\mathcal{I} \subseteq \mathcal{A}$ and the remaining ones will be the set of dependent joints $\mathcal{D}$.

4.2. Minimum energy loss for body movement

To move the platform, the robot needs at least two legs to support it. This means that there is at least one closed kinematic chain between the platform and the support surface. Then the closed kinematic chain should be formed by legs 1 (right front or RF) and 3 (left hind or LH) or legs 2 (left front or LF) and 4 (right hind or RH), according to the gait (Fig. 7), when these legs are gripping the surface (i.e $l = \{1, 3\}$ or $l = \{2, 4\}$). Under these circumstances, the robot behaves as a parallel robot.

For parallel robots, the load torques vector $\mathbf{\Gamma}_l$ originally computed by the Euler–Lagrange equation (37), should be efficiently computed by the Principle of Virtual Work as stated in ref. [21].

To reproduce a more realistic situation, a friction torque $\tau_f(\dot{\theta}_{l_i})$ was introduced into the model. This torque is commonly modeled according to the following equation:

$$\tau_f(\dot{\theta}_{l_i}) = \beta sign(\dot{\theta}_{l_i}) + \sigma \dot{\Theta}_{l_i}, \tag{61}$$

where β and σ are the Coulomb and viscous friction coefficients respectively.[24] For all joints in the robot, it was assumed that $\beta = 0.795 \times 10^{-2}$ N $\cdot$ m and $\sigma = 5.16 \times 10^{-3}$N $\cdot$ s. Note that function $sign$ could be approximated by a continuous function (e.g. an approximation by Fourier Series) to satisfy the first condition of the Theorem of Contracting Maps.

In the body movement, the number of joints is larger than the number of degrees of freedom. Then, in addition to the set of active joints $\mathcal{A}$, there will be a set of independent joints $\mathcal{I}$ from which the

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

Table II. Classification of joints for the *gait*.

Sets	Body movement Gripping legs $l = 1, 3$ (RF and LH)	Leg movement Legs $l = 2, 4$ (LF and RH)
$\mathcal{A}$	$\{\theta_{1_2}(K), \theta_{1_3}(A), \theta_{1_4}(F), \theta_{3_2}(K), \theta_{3_3}(A), \theta_{3_4}(F)\}$	$\{\theta_{2_1}(H), \theta_{2_2}(K), \theta_{2_3}(A)\}$, $\{\theta_{4_1}(H), \theta_{4_2}(K), \theta_{4_3}(A)\}$
$\mathcal{P}$	$\{\theta_{1_1}(H), \theta_{3_1}(H)\}$	$\{\theta_{2_4}(F)\}$, $\{\theta_{4_4}(F)\}$
$\mathcal{I}$	$\{\theta_{3_2}(K), \theta_{3_3}(A), \theta_{3_4}(F)\}$	$\{\theta_{2_1}(H), \theta_{2_2}(K), \theta_{2_3}(A)\}$, $\{\theta_{4_1}(H), \theta_{4_2}(K), \theta_{4_3}(A)\}$
$\mathcal{D}$	$\{\theta_{1_{2(K)}}, \theta_{1_3}(A), \theta_{1_4}(F), \theta_{1_1}(H), \theta_{3_1}(H)\}$	$\{\emptyset\}$, $\{\emptyset\}$

Table III. Control constraints.

Parameters	Body movement	Legs movements ($l = 2$)
N	100	100
t_f	2 s	1.8 s
ξ	10^{-7}	10^{-7}
$I_{\max_{l_i}}$	2.0 A	2.0 A
$V_{\max_{l_i}}$	12 V	12 V
$[\check{\theta}_{3_2}, \hat{\theta}_{3_2}]$	[0.85, 0.92] rad	
$[\check{\theta}_{3_3}, \hat{\theta}_{3_3}]$	[2.24, 2.3] rad	
$[\check{\theta}_{3_4}, \hat{\theta}_{3_4}]$	[−0.54, 0.55] rad	
$[\check{\theta}_{2_1}, \hat{\theta}_{2_1}]$		[−0.78, 0.59] rad
$[\check{\theta}_{2_2}, \hat{\theta}_{2_2}]$		[−0.87, 0.78] rad
$[\check{\theta}_{2_3}, \hat{\theta}_{2_3}]$		[−1.5, −0.66] rad

robot movement will be defined. The motion of this set of independent joints will be calculated by the algorithm. After that these joints will be used to calculate the performance of other joints $\mathcal{D}$ by solving the inverse kinematics problem.

Table II shows an example of joint sets chosen for the movement of body and legs. These sets do not necessarily have to be identical to that used in this example.

The control constraints for the robot movements are presented in Table III. The interval $[\check{\theta}_{l_i}; \hat{\theta}_{l_i}]$ for each joint is calculated according to the solution of the inverse kinematics problem such that all the points generated by the algorithm are inside the body and gripper workspace.[20]

Observe that the optimal trajectory generated for the body motion causes a translational movement practically only in the x–y plane (Fig. 8). That performance is similar to the one observed in the motion of geckos and chameleons.[3] Note that no electrical power is used to vary the potential energy of the body and all the power is spent for the translation of the robot.

According to Table II, the algorithm is applied to the set of independent joints θ_{3_2}, θ_{3_3}, and θ_{3_4}, namely knee (K), ankle (A), and finger (F) respectively of the LH leg. Thus, an effective control law is computed to drive the motors associated with these joints. Figure 8 shows that the optimal movement computed is focused in the forward direction. To obtain this performance, most of the torque is generated in the finger of the RH leg (joint θ_{3_4}). The motor of this joint is the one that guides the movement of the body in this case. This fact is responsible for the current i_{3_4} being the greatest of all mainly at the end of the step when the movement has to be stopped. Although joints θ_{1_3} and θ_{3_3} (knees) are motionless, they support the robot body weight. Hence, their currents and torques are practically constant throughout the movement (Figs. 9 and 11).

4.3. Minimum energy loss for the leg movement

When the leg is in the air, it behaves as an industrial manipulator attached to the corner of the platform. For a given initial and final positions of the gripper, the algorithm computes an optimal solution. In this case, the set of active joints is $\mathcal{A} = \{\theta_{l_1}, \theta_{l_2}, \theta_{l_3}\}$ for $l = 2, 4$ (Table II).

Figs. 12–15 show the optimal control law corresponding to leg $l = 2$ (LF) when it is moving in the air.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

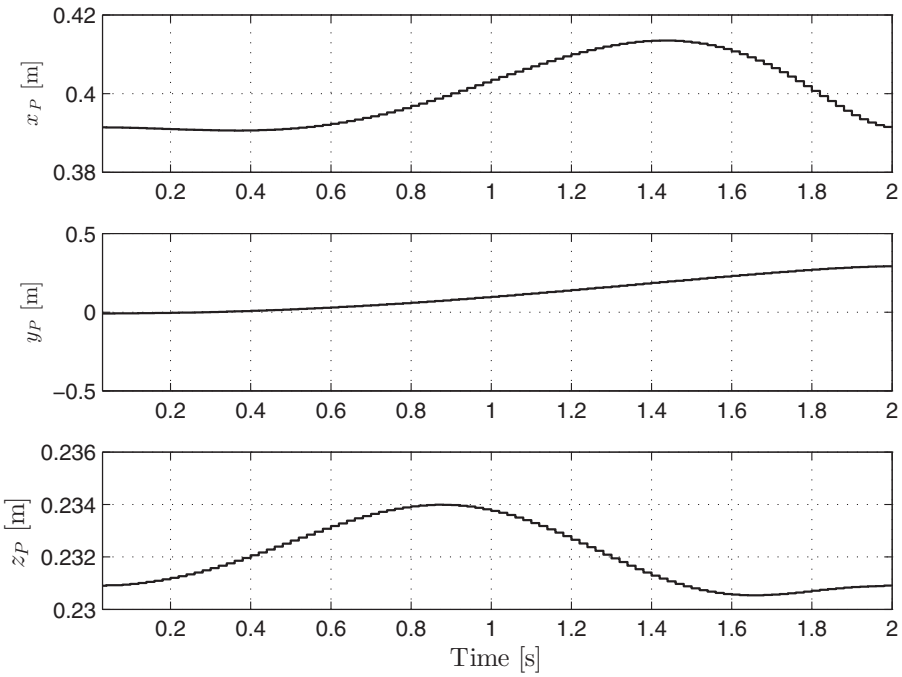

Fig. 8. Coordinates of the center of mass of robot platform in the full body movement.

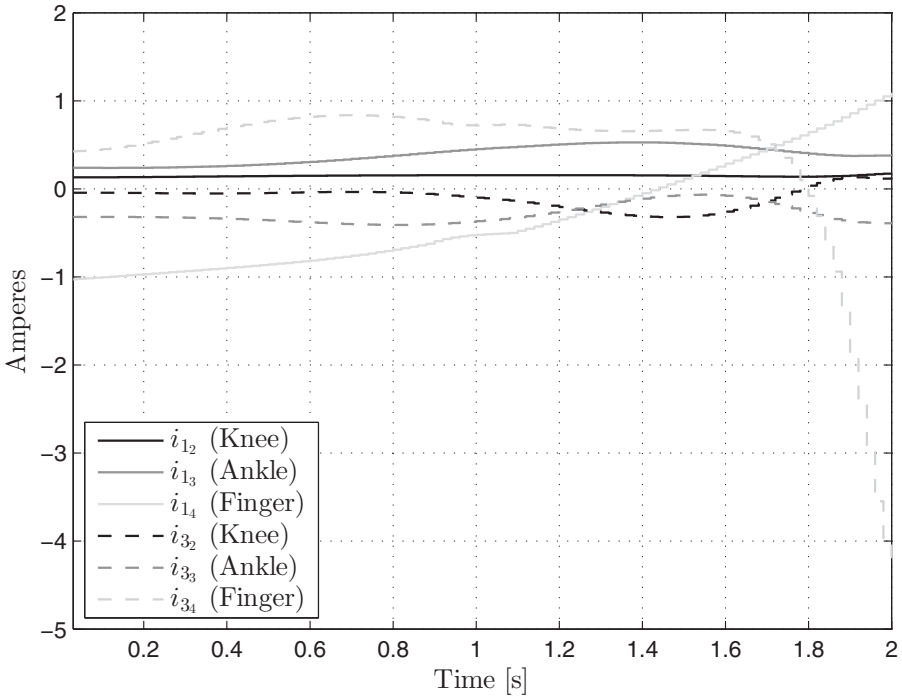

Fig. 9. Armature current of active joints in the full body movement.

Note from Table III that the constraints of joint angles are more relaxed than in the previous movement; this is because in an open kinematic chain, the constraints of the movement are minor (Fig. 12).

Figure 13 shows that, although the total current was brought to constraint limits to increase the velocity of the movement, the torques computed were not very high. Observe in Fig. 13 that the performance of torques of joints θ_{2_1} (hip) and θ_{2_2} (knee) are quasi symmetric, while the torque of joint

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

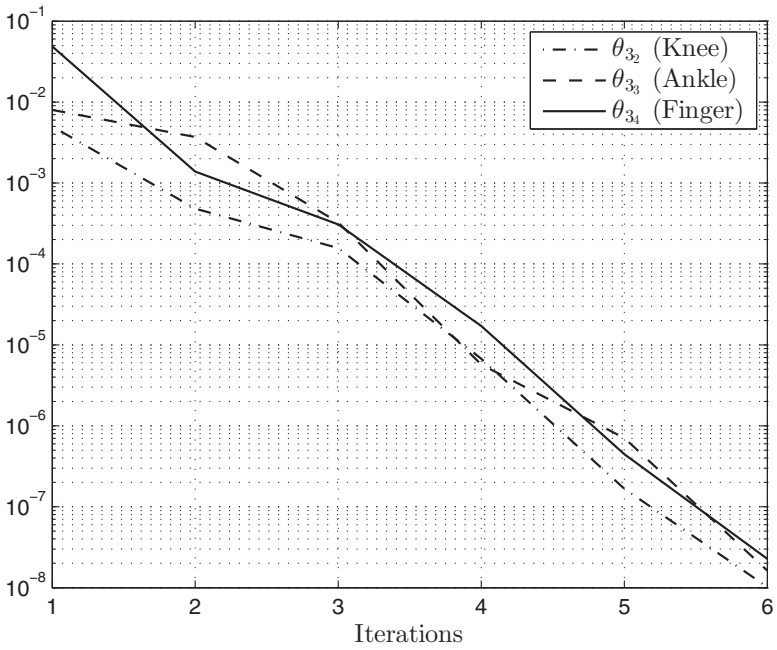

Fig. 10. Error of the control law between two consecutive iterations for independent joints in the full body movement.

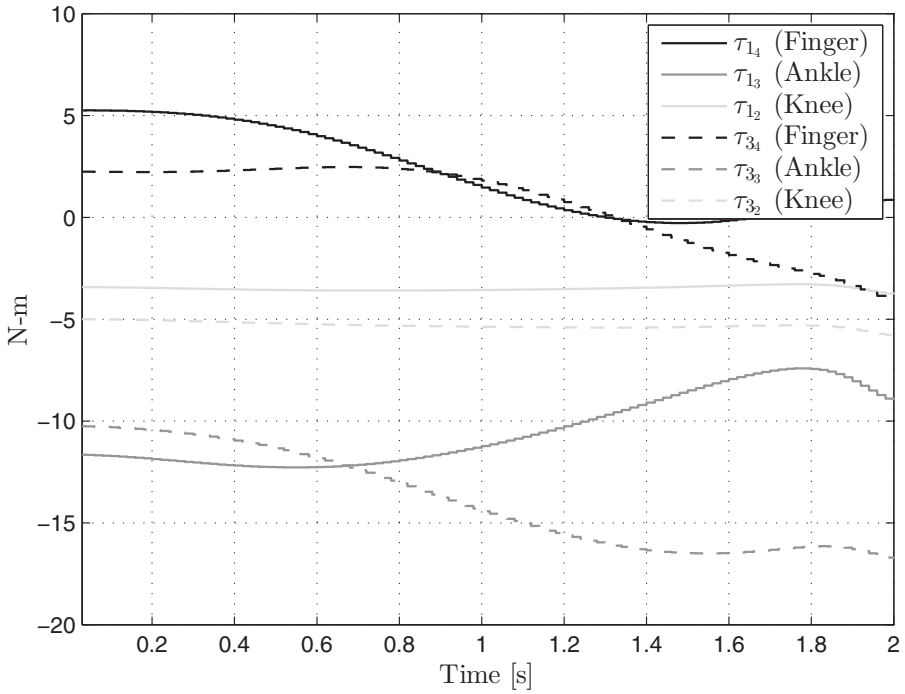

Fig. 11. Torque of active joints in full body movement.

θ_{2_3} (ankle) is almost negligible. This means that the ankle movement is due to the coupling between other joints rather than by an effective law control applied on it.

4.4. Analysis of results

It is worth noting that before running each implementation of the algorithm, the third condition of convergence of the theorem was tested. All the values obtained for α were lower than 1 (Table IV). For both body movement and single-leg movement, the process of convergence was quick. Figs. 10

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

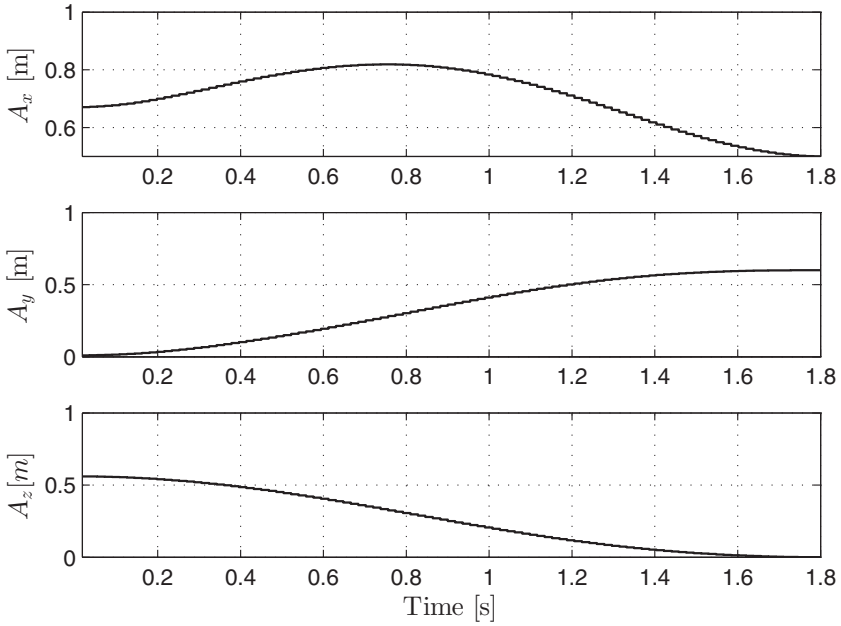

Fig. 12. Coordinates of the gripper relative to leg $l = 2$ (left front) in a single-leg movement.

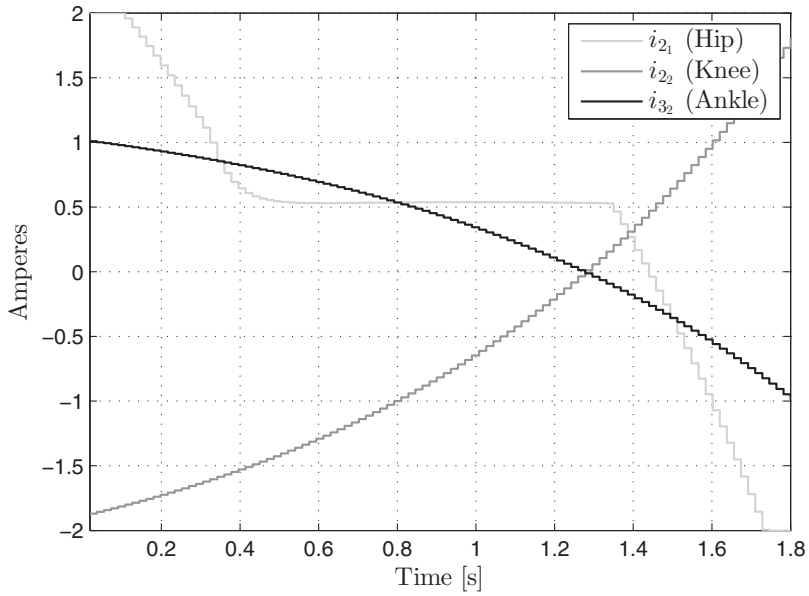

Fig. 13. Armature currents of active joints in leg $l = 2$ (left front) in a single-leg movement.

and 14 show that although the stopping criterion was very small, the optimal solutions were found at the sixth and ninth iteration.

An interesting point is the power consumption of body and leg movements; see Figs. 16 and 17. Note that the path toward the global minimum point is not necessarily monotonous.

Figure 18 shows the total power consumption of motors in LF leg obtained from three different starting points. Note that, whatever the starting point used, the algorithm converges to the same minimum. This is due to the third consequence of the Theorem of Contracting Maps.

Herein, the total power consumption of motors relative to the trajectory obtained by the algorithm proposed with two classical trajectories obtained by a cubic polynomial and a linear segment with parabolic blend (LSPB) is compared. Without losing generality, the comparison is made with the movement of one leg discussed above.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

Table IV. Results of optimization.

Stage	Body movement		Legs movement	
Legs	$l = 1, 3$	$l = 2, 4$	$l = 2$	$l = 4$
α	0.00045	0.00038	0.00021	0.00029
Iteration k	6	5	9	4
t_{proc}(s)	2.90	2.35	4.09	1.87
Power (W)	1108.45	1096.56	562.35	561.10

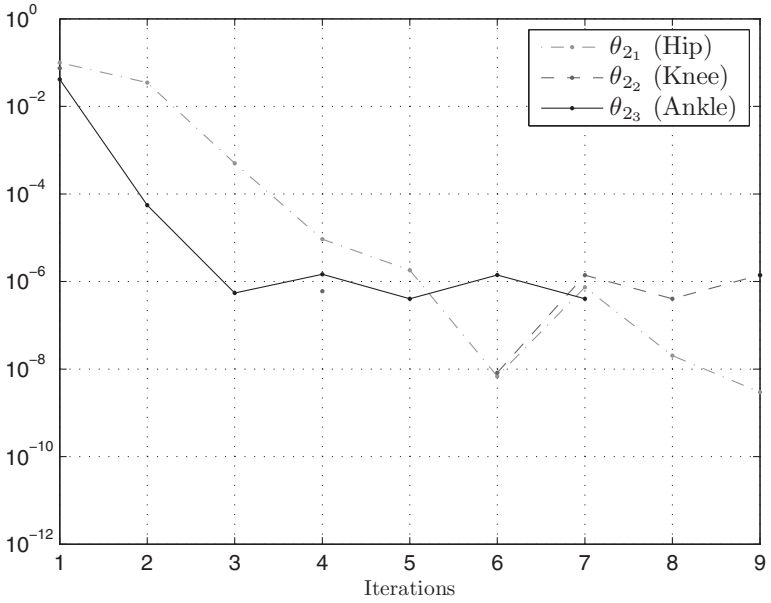

Fig. 14. Error of the control law between two consecutive iterations for independent joints in leg $l = 2$ (left front) in a single-leg movement.

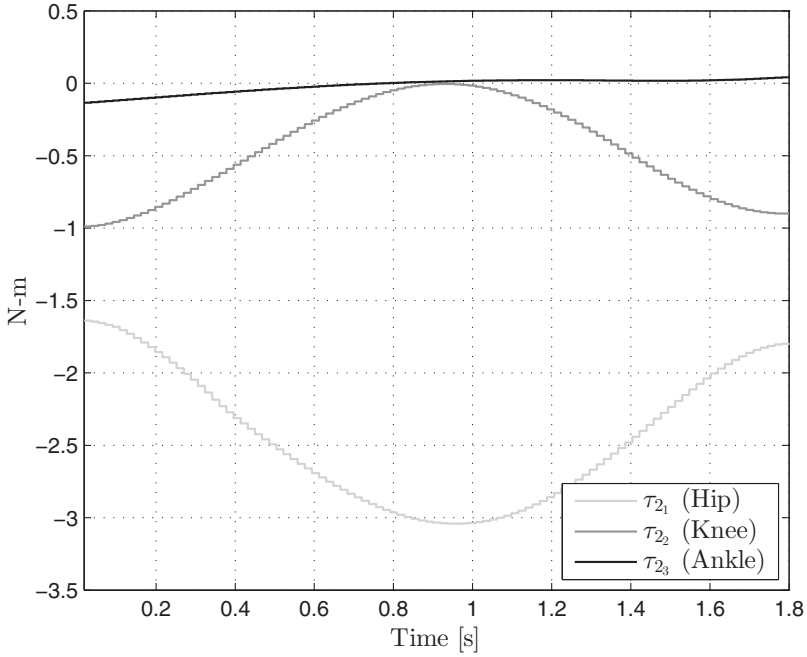

Fig. 15. Torques of the joints in leg $l = 2$ (left front) in a single-leg movement.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

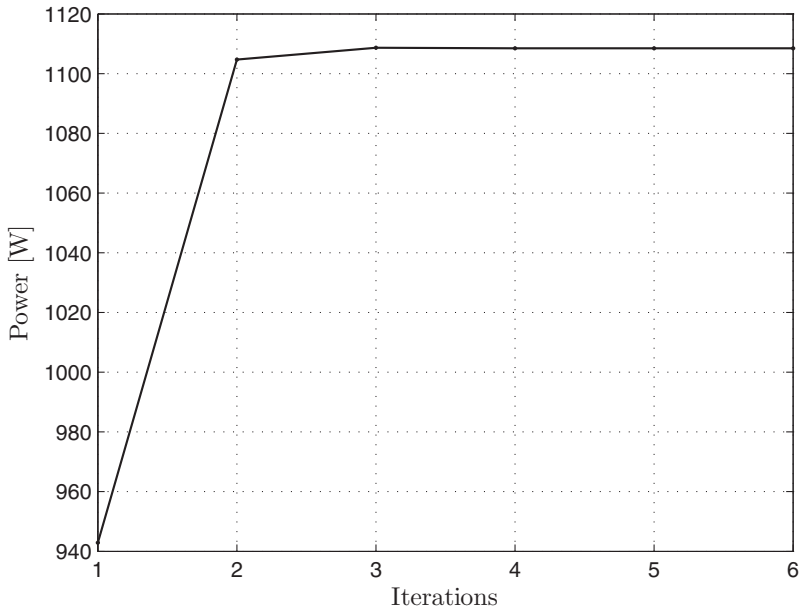

Fig. 16. Total power loss by the motors at the body movement.

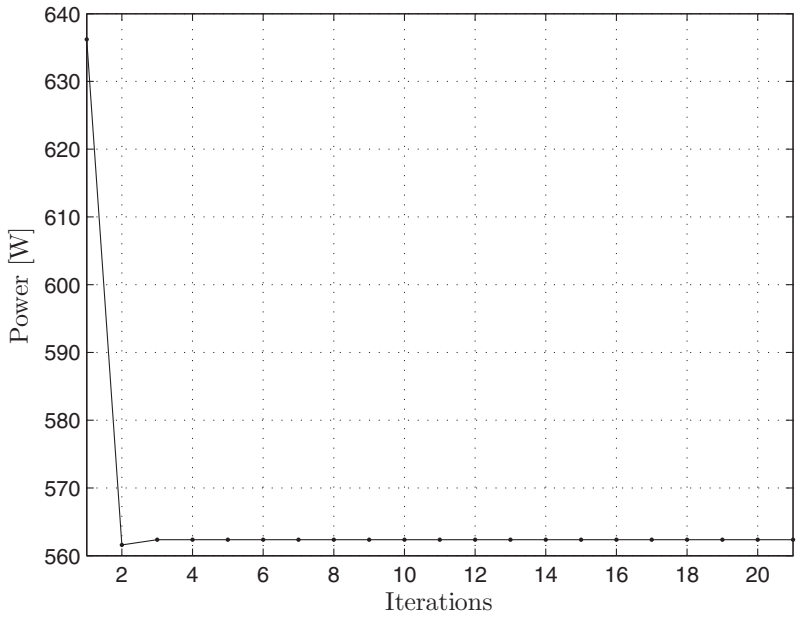

Fig. 17. Total power loss by the motors at the leg movement.

Although the trajectories obtained in Fig. 19 are almost equal, the main difference is observed in the acceleration of each joint (Fig. 20). In the case of optimal movement, the acceleration of the hip is close to zero most of the time. This performance of the acceleration causes a constant velocity of joints most of the time, while the acceleration of the joint in the cubic trajectory is always increasing in one direction or another, and in the case of the LSPB trajectory, it has a nonzero value, i.e. two-thirds of the total runtime.

Another difference in the performance of the algorithm is observed in the armature current and torques of joints (Figs 21 and 22). Note also that the cubic algorithm shows a performance similar to that obtained by the optimal algorithm in the torque curves; however, there is a great difference in the

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

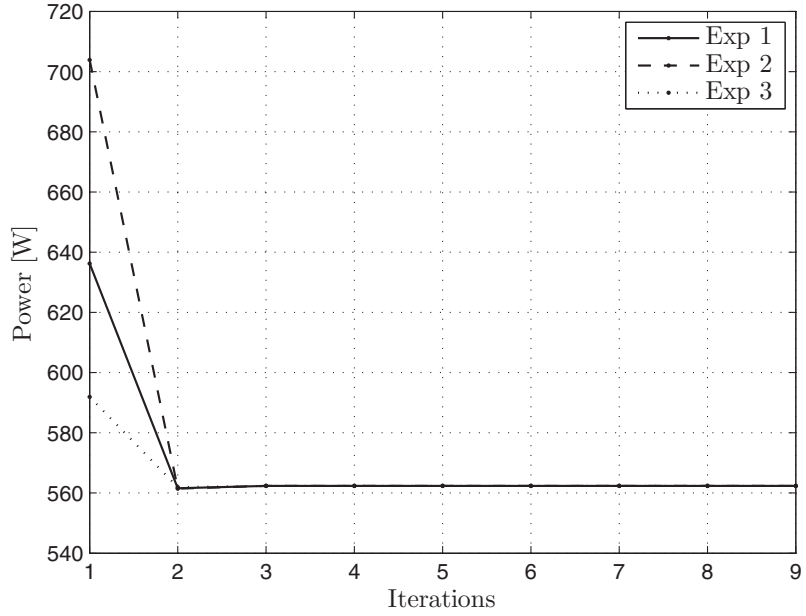

Fig. 18. Total power consumption obtained from three different starting points.

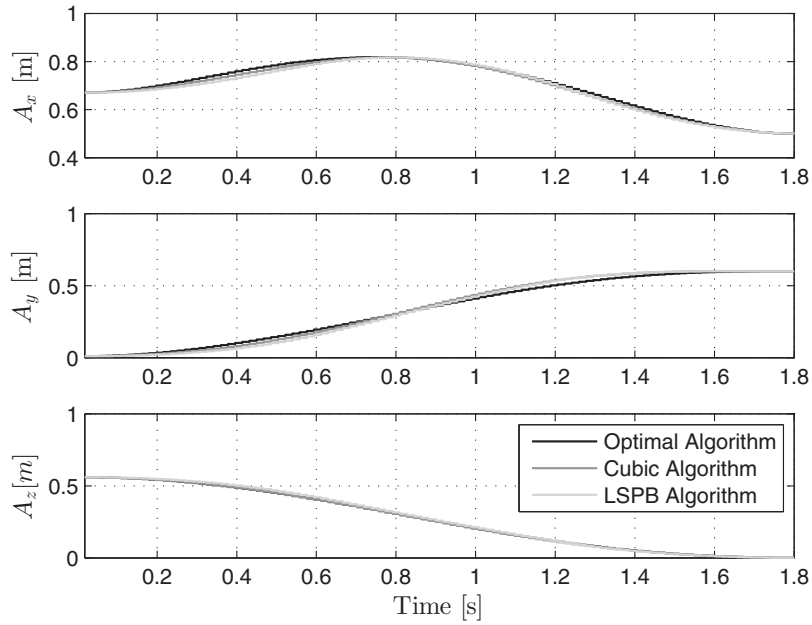

Fig. 19. Comparison of the Cartesian movement of the gripper obtained by the algorithms: LSPB, cubic polynomial, and optimal.

armature current of the motor driving the hip of leg (θ_{2_1}). In our algorithm, this armature current has a constant section in the middle of the movement, whereas the curve generated by the cubic algorithm shows a parabolic form. This constant section guarantees a more efficient power consumption.

On the other hand, the curves from the LSPB algorithm are completely different. Similar to the performance of the algorithm proposed, the armature current of the hip has a constant section. But, different from the last one, the armature current first increases and then decreases, while in the optimal algorithm the current always decreases. The torques are also somewhat different when compared with the torques generated by the algorithm proposed. Observe that for this movement, the key joint is

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

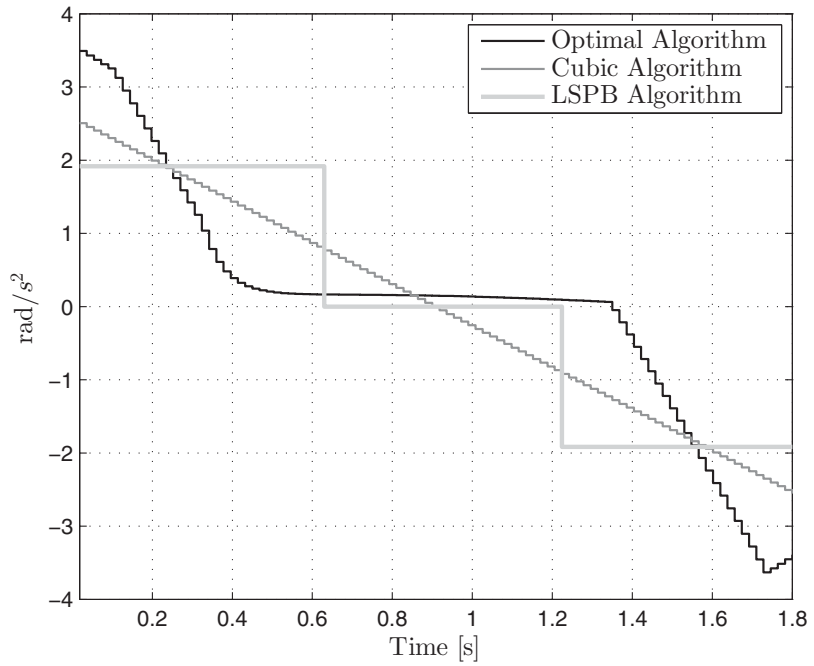

Fig. 20. Acceleration of joint θ_{2_1} (hip) in LSPB, cubic, and optimal trajectories.

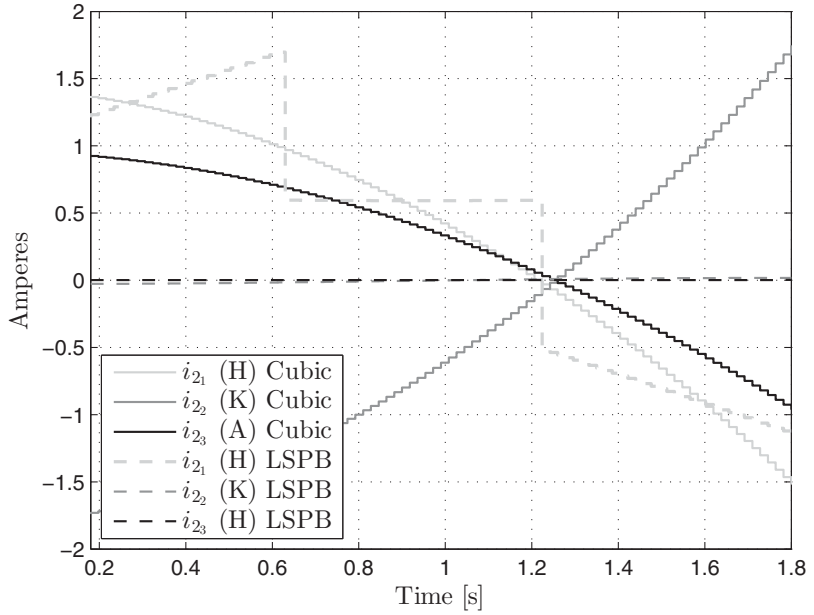

Fig. 21. Comparison of armature currents obtained by the algorithms: LSPB, cubic polynomial, and optimal.

the hip of the leg. Due to this performance, the power consumption of each joint in cubic and LSPB trajectories is larger than the power consumption in the optimal trajectory (Fig. 23).

Finally, Table IV shows the value of the term α, the number of iterations required for the convergence of the algorithm, the processing time, and the power loss of the robot at the two stages of gait when the stopping criterion is $\xi = 10^{-7}$.

The runtime depends on the value of ξ used as a stopping criterion. Thus, note that the relatively high runtime of the algorithm is due to the small value of the stopping criterion used in this simulation. In real applications, this time can be reduced by increasing the value of ξ (Fig. 24). The number of iterations and runtime is also related to the starting point of the algorithm. In this sense, techniques for a better starting point selection are being studied.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

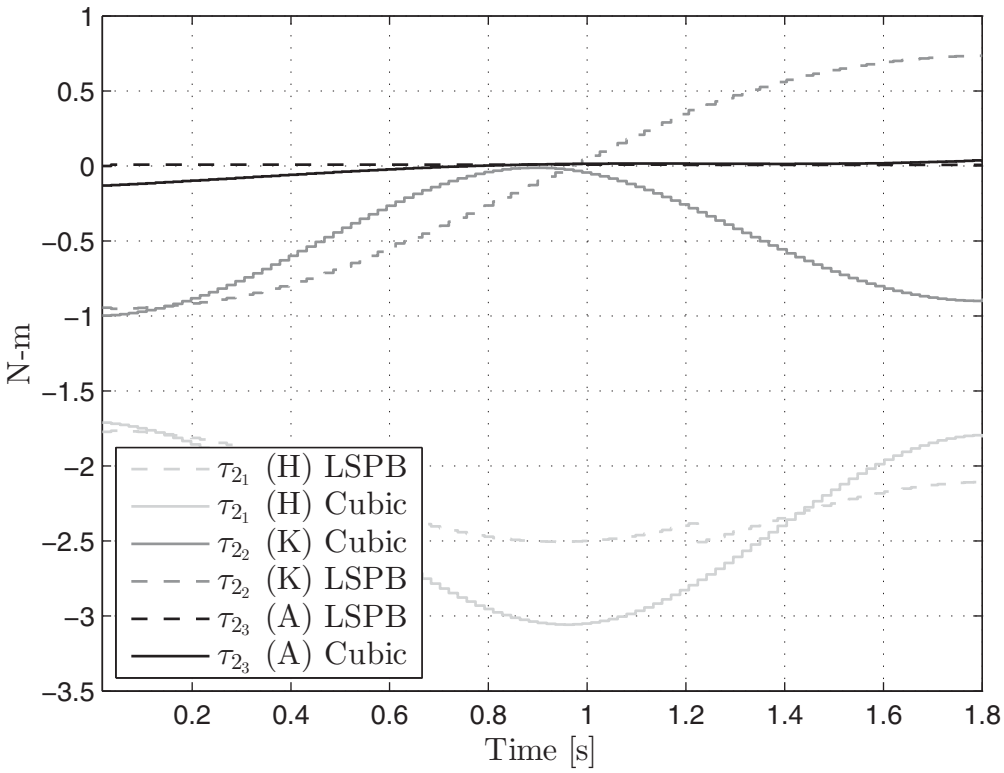

Fig. 22. Comparison of the torques obtained by the algorithms: LSPB, cubic polynomial, and optimal.

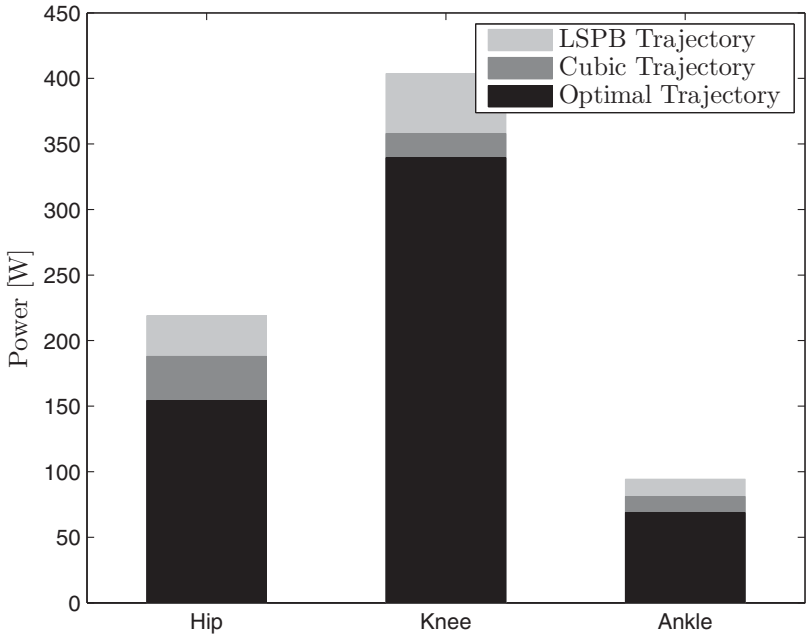

Fig. 23. Power consumption per joints. LSPB: 218.9 W, 403.5 W, 94.34 W; cubic: 187.8 W, 357.8 W, 81.03 W; and optimal: 154.26 W, 339.4 W and 68.68 W.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

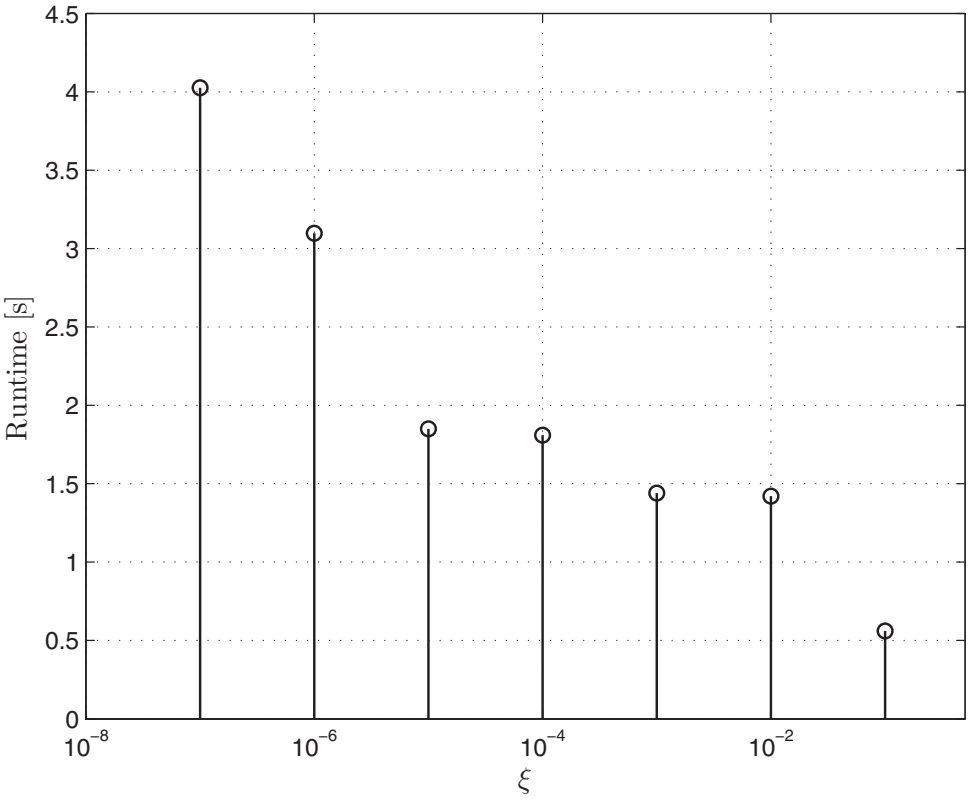

Fig. 24. Decrease of runtime of the algorithm for different ξ when the optimal trajectory of leg 2 (left front) is calculated.

In accordance with Table IV, the mean processing time by iteration was approximately 0.45 s.[2]

5. Conclusions

An OFMPP based on the minimization of energy loss in kinematic chains driven by DC motors was presented. Using the strategy of joint independent control, the original problem was transformed into a simple QPP subject to linear constraints. Note that in the algorithm proposed, the coupling effects between links are not completely eliminated but treated as a disturbance (which is reduced by the gear reduction that must be lower than 1); moreover, this coupling is updated at each iteration by (37) before the computation of next iteration. As the nonlinearities introduced by the vector of load torque are reduced by the gear reduction ratio, the system may be formulated as a QPP that can be solved efficiently. The objective of the algorithm is to obtain an optimal trajectory for each joint, hence this approach is useful because the trajectory computed is passed to the low level control as a reference while the effects of the disturbances are minimized by gear reduction. In this sense, controller's computational load may be reduced by low-cost hardware.

From the solution of the QPP obtained for each motor, the optimal solution to control the whole kinematic chain through an iterative algorithm was obtained. Proof of the convergence of the algorithm was also presented. Constraints such as on armature voltage andarmature current were used in the problem in addition to the joint range of movements.

The joints in the kinematic chain presented were grouped and treated separately at each iteration. To ensure solution to the constraints of the set of dependent joints, several criteria were used. Although in the example discussed all the motors are equal, the algorithm, in principle, could be applied to any mechatronic system driven by DC motors. As in the solution of the QPP, one joint does not affect any other joint; each QPP could be solved in parallel to reduce the runtime.

[2] All the simulations were made using the platform MATLAB R2011b® with a 64 bit processor Intel(R) CORE(TM) I7 CPU 930 @ 2.80 GHz 2.66 GHz, and 8 GB RAM.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

As legged robots are usually controlled by DC motors that reach their maximum torques at high speeds and using high gear ratios, we decided to employ the algorithm proposed for determining optimal trajectories for a quadruped legged robot.

As legged locomotion involves transition from open to closed kinematic chain and *vice versa*, and as the algorithm does not perform this kind of transition, two different versions of the Euler–Lagrange equation are necessary. For that the control was defined in two parts in accordance with a time-varying topology and a time-varying number of degrees of freedom of the system. Special attention was devoted to the parallel topology controlled. For that the algorithm was applied over a set of active and independent joints, and from these the movement of other joints was calculated. In both cases, for the control of open and closed kinematic chains, the convergence of the algorithm was very fast, reaching the stop criterion in few iterations. Besides, the results suggest that $\alpha < 1$ holds for a certain range of initial conditions and characteristics of the system.

Future works could include high levels of optimization techniques such as to re-optimize the phase after some states have been stored,[2] or analysis of trajectories where the sum of forces and torques across the body are minimized or amplified.

Acknowledgments

The authors are grateful to CAPES and CNPq for their partial support for this research.

References

1. J. Angeles, *Fundamentals of Robotic Mechanical Systems. Theory, Methods, and Algorithms* (Springer, New York, NY, 2007).
2. C. Atkeson and B. Stephens, "Random sampling of states in dynamic programming," *IEEE Trans. Syst. Man Cybern.* **38**(4), 924–929 (Aug. 2008).
3. K. Autumn, S. Hsieh, D. Dudek, J. Chen, C. Chitaphan and R. J. Full, "Dynamics of geckos running vertically," *J. Exp. Biol.* **209**(2), 260–272 (2006).
4. R. Bernardi and J. J. Da Cruz, "Kamanbaré: A tree-climbing biomimetic robotic platform for environmental research," *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO)* (2007).
5. J. Betts, "Survey of numerical methods for trajectory optimization," *Guid. Control Dyn.* **21**, 193–207 (1998).
6. J. Bobrow, F. Park and A. Sideris, "Recent Advances on the Algorithmic Optimization of Robot Motion," *Technical Report* (Departament of Mechanical and Aerospace Engineering, University of California, 2004).
7. A. Bryson and Y. Ho, *Applied Optimal Control. Optimization, Estimation and Control* (John Wiley, Hoboken, NJ, 1975).
8. T. Chettibi, M. Haddada, A. Labedb and S. Hanchi, "Generating optimal dynamic motions for closed-chain robotic systems," *Eur. J. Mech. Solids* **24**, 504–518 (2005).
9. P. Gregorio, M. Ahmadi and M. Buehler, "Design, control, and energetics of an electrically actuated legged robot," *IEEE Trans. Syst. Man Cybern.* **27**(4), 626–632 (Aug. 1997).
10. P. Henrici, *Elements of Numerical Analysis*, 1st ed. (John Wiley, Hoboken, NJ, 1964).
11. M. Hildebrand, "Vertebrate locomotion an introduction how does an animal's body move itself along?," *Bioscience* **39**, 764–765 (1989).
12. D. Hull, "Conversion of optimal control problems into parameter optimization problems," *Guid. Control Dyn.* **20**, 57–62 (1997).
13. M. Kahn and B. Roth, "The near minimum time control of open loop articulated kinematic chains," *ASME J. Dyn. Syst. Meas. Control* **11**, 164–172 (1971).
14. Kozlowski, K. (ed.), *Robot Motion and Control* (Springer, New York, NY, 2006).
15. J. Latombe, "Motion planning: a journey of robots, molecules, digital actors, and other artifacts," *Int. J. Robot. Res.* **18**, 1119–1128 (1999).
16. C. Liu and C. Atkeson, "Standing Balance Control Using a Trajectory Library," *Proceedings of the International Conference on Intelligent Robots and Systems* (2009).
17. Y. Liu, C. Wang, J. Li and L. Sun, "Time-optimal Trajectory Generation of a Fast-motion Planar Parallel Manipulator," *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct. 2006) pp. 754–759.
18. A. Majumdar and R. Tedrake, "Robust Online Motion Planning With Regions of Finite Time Invariance," **In**: *Algorithmic Foundations of Robotics X*, Springer Tracts in Advanced Robotics, Vol. 86 (Springer, New York, NY, 2013) pp 543–558.
19. F. Pfeiffer, J. Eltze and H.-J. Weidemann, "The TUM walking machine," *Intell. Autom. Soft Comput.* **1**, 307–323 (1995).
20. A. Potts, "Modelagem e Controle ótimo de um Robô Quadrúpede," *Ph.D. Thesis* (Escola Politécnica da Universidade de São Paulo, 2011).

21. A. Potts and D. J. Cruz, "A Kinematical and Dynamical Analysis of a Quadruped Robot," **In:** *Mobile Robots – Current Trends* (Zoran Gacovski, ed.) (Intech, Rijeka, Croatia, 2011) pp. 239–262.
22. M. Spong and M. Vidyasagar, *Robot Dynamics and Control*, 1st ed. (John Wiley, Hoboken, NJ, 1995).
23. O. Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Ann. Oper. Res.*, **37**(1), 357–373 (1992).
24. M. Tomizuka, "Robust digital motion controllers for mechanical systems," *Robot. Auton. Syst.* **19**(2), 143–149 (Dec. 1996).
25. H. Yang and J. Slotine, "Fast algorithms for nearminimum-time control of robot manipulators," *J. Robot. Syst.* **18**(7), 343–356 (2001).
26. M. Zhong and E. Todorov, "Moving least-squares approximations for linearly solvable stochastic optimal control problems," *J. Control Theory Appl.* **9**(3), 451–463 (2011).

Appendix A

To solve QPP numerically, it is necessary to express it in matrix form. In this sense, let us define the following matrices:

$$\boldsymbol{\Upsilon}_{l_i} = \begin{bmatrix} \mathbf{C}_{l_i}(e^{\mathbf{A}_{l_i}t_1}\mathbf{x}_{l_i}(0) - r\mathbf{H}_{l_i}(t_1)\Gamma_{l_i}) \\ \vdots \\ \mathbf{C}_{l_i}(e^{\mathbf{A}_{l_i}t_N}\mathbf{x}_{l_i}(0) - r\mathbf{H}_{l_i}(t_N)\Gamma_{l_i}) \end{bmatrix}, \tag{A1}$$

$$\boldsymbol{\Xi}_{l_i} = \begin{bmatrix} \mathbf{C}_{l_i}\mathbf{H}_{l_i}(t_1) \\ \vdots \\ \mathbf{C}_{l_i}\mathbf{H}_{l_i}(t_N) \end{bmatrix}, \tag{A2}$$

$$\bar{\boldsymbol{\Upsilon}}_{l_i} = \begin{bmatrix} [0\ 1]\,(e^{\mathbf{A}_{l_i}t_1}\mathbf{x}_{l_i}(0) - r\mathbf{H}_{l_i}(t_1)\Gamma_{l_i}) \\ \vdots \\ [0\ 1]\,(e^{\mathbf{A}_{l_i}t_N}\mathbf{x}_{l_i}(0) - r\mathbf{H}_{l_i}(t_N)\Gamma_{l_i}) \end{bmatrix}, \tag{A3}$$

and

$$\bar{\boldsymbol{\Xi}}_{l_i} = \begin{bmatrix} [0\ 1]\,\mathbf{H}_{l_i}(t_1) \\ \vdots \\ [0\ 1]\,\mathbf{H}_{l_i}(t_N) \end{bmatrix}. \tag{A4}$$

It is then possible to rewrite inequality constraints (31)–(33) in matrix form as follows:

$$\mathbf{T}_{l_i} \leq K_{m_{l_i}}\mathbf{I}_{\max_{l_i}}, \tag{A5}$$

$$\left(\frac{R_{l_i}}{K_{l_i}}\mathbf{E}(t_N) + K_{b_{l_i}}\bar{\boldsymbol{\Xi}}_{l_i}\right)\mathbf{T}_{l_i} \leq -K_b\bar{\boldsymbol{\Upsilon}}_{l_i} + \mathbf{V}_{\max_{l_i}}, \tag{A6}$$

$$\boldsymbol{\Xi}_{l_i}\mathbf{T}_{l_i} \leq \check{\boldsymbol{\Theta}}_{l_i} - \boldsymbol{\Upsilon}_{l_i} \tag{A7}$$

and

$$\boldsymbol{\Xi}_{l_i}\mathbf{T}_{l_i} \leq \hat{\boldsymbol{\Theta}}_{l_i} - \boldsymbol{\Upsilon}_{l_i}, \tag{A8}$$

where $\mathbf{I}_{\max_{l_i}} = [I_{\max_{l_i}}, \ldots, I_{\max_{l_i}}]$ and $\mathbf{V}_{\max_{l_i}} = [V_{\max_{l_i}}, \ldots, V_{\max_{l_i}}]$ are column vectors of dimension N. Finally, vectors $\check{\boldsymbol{\Theta}}_{l_i} = [\check{\theta}_{l_i}(t_1), \check{\theta}_{l_i}(t_2), \ldots, \check{\theta}_{l_i}(t_N)]^T$ and $\hat{\boldsymbol{\Theta}}_{l_i} = [\hat{\theta}_{l_i}(t_1), \hat{\theta}_{l_i}(t_2), \ldots, \hat{\theta}_{l_i}(t_N)]^T$. Then using Eqs. (A5) to (A8), it is possible to write all constraints in a single matrix inequality:

$$\Lambda_{l_i}\mathbf{T}_{l_i} \leq \Omega_{i_j}, \tag{A9}$$

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

where:

$$\mathbf{\Lambda}_{l_i} = \begin{bmatrix} \mathbf{E}(t_N) \\ \dfrac{R_{l_i}}{K_{l_i}}\mathbf{E}(t_N) + K_{b_{l_i}}\bar{\mathbf{\Xi}}_{l_i} \\ \mathbf{\Xi}_{l_i} \\ \mathbf{\Xi}_{l_i} \end{bmatrix} \tag{A10}$$

and

$$\mathbf{\Omega}_{l_i} = \begin{bmatrix} -K_{l_i}[I_{\max_{l_i}}]_N \\ K_b\bar{\mathbf{\Upsilon}}_{l_i} - [V_{\max_{l_i}}]_N \\ \left(\mathbf{\Upsilon}_{l_i} - \check{\mathbf{\Theta}}_{l_i}\right) \\ \left(\mathbf{\Upsilon}_{l_i} - \hat{\mathbf{\Theta}}_{l_i}\right) \end{bmatrix}. \tag{A11}$$

The equality constraint is written as:

$$\mathbf{H}_{l_i}(t_N)\mathbf{T}_{l_i} = \mathbf{F}_{l_i}, \tag{A12}$$

where $\mathbf{F}_{l_i} = r\mathbf{H}_{l_i}(t_N)\mathbf{\Gamma}_{l_i} + e^{\mathbf{A}_{l_i}t_N}\mathbf{x}_{l_i}(0) + \mathbf{x}_{l_i}(0) - \tilde{\mathbf{x}}_{l_i}$.

Appendix B

To verify the third condition of the Theorem of Contracting Map, the evaluation of the value of α is required. According to (60) it can be calculated from the Jacobian matrix of the recurrence law employed. Using the recurrence law (57) and from equations (50) to (52), we have the following:

$$\begin{aligned}\theta_{l_i}^k(t_j) = {} & \mathbf{C}_{l_i}e^{\mathbf{A}_{l_i}t_j}\mathbf{x}_{l_i}(0) + \mathbf{C}_{l_i}\mathbf{H}_{l_i}(t_j)\frac{\mathbf{F}_{l_i}^T\mathbf{H}_{l_i}(t_N)}{\mathbf{H}_{l_i}(t_N)\mathbf{H}_{l_i}^T(t_N)} \\ & - \mathbf{C}_{l_i}\mathbf{H}_{l_i}(t_j)\mu_{l_i}^{\mathcal{S}^*}\mathbf{\Lambda}_{l_i}^{\mathcal{S}}\mathbf{Q}_{l_i}^{-1} - r_{l_i}\mathbf{C}_{l_i}\mathbf{H}_{l_i}(t_j)\mathbf{\Gamma}_{l_i}^{k-1}.\end{aligned} \tag{B1}$$

Then without loss of generality it is assumed that there are no active inequality constraints in the feasible solution of QPP. Thus, $\mu_{l_i}^{\mathcal{S}^*} = \mathbf{0}$ and

$$\mathbf{\Theta}_{l_i}^k = \mathbf{K}_{l_i} + \mathbf{\Xi}_{l_i}\frac{\mathbf{F}_{l_i}^T\mathbf{H}_{l_i}(t_N)}{\mathbf{H}_{l_i}(t_N)\mathbf{H}_{l_i}^T(t_N)} - r_{l_i}\mathbf{\Xi}_{l_i}\mathbf{\Gamma}_{l_i}^{k-1}, \tag{B2}$$

where:

$$\mathbf{K}_{l_i} = \begin{bmatrix} \mathbf{C}_{l_i}e^{\mathbf{A}_{l_i}t_1}\mathbf{x}_{l_i}(0) \\ \vdots \\ \mathbf{C}_{l_i}e^{\mathbf{A}_{l_i}t_N}\mathbf{x}_{l_i}(0) \end{bmatrix}. \tag{B3}$$

Or in affine form,

$$\mathbf{\Theta}_{l_i}^k = \mathbf{X}_{l_i}\mathbf{\Gamma}_{l_i}^{k-1} + \mathbf{Y}_{l_i}, \tag{B4}$$

where

$$\mathbf{X}_{l_i} = -r_{l_i}\hat{\mathbf{\Xi}}_{l_i} \tag{B5}$$

and

$$\mathbf{Y}_{l_i} = \mathbf{K}_{l_i} + \mathbf{\Xi}_{l_i}\frac{\mathbf{F}_{l_i}^T\mathbf{H}_{l_i}(t_N)}{\mathbf{H}_{l_i}(t_N)\mathbf{H}_{l_i}^T(t_N)}. \tag{B6}$$

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press

Now using Definition 2 and Eq. (B2) we can express vector $\mathbf{\Theta}_l^k$ as:

$$\mathbf{\Theta}_l^k = \mathbf{X}_l \mathbf{\Gamma}_l^{k-1} + \mathbf{Y}_l, \tag{B7}$$

where $\mathbf{X}_l$ and $\mathbf{Y}_l$ are a matrix of dimension $MN \times MN$ and a column vector of dimension MN respectively. Since the load torque vector $\mathbf{\Gamma}_l$ can be written as a function of $\mathbf{\Theta}_l^k$, a recurrence law can be written in the form $\mathbf{\Theta}_l^{k+1} = f(\mathbf{\Theta}_l^k)$, where $f(\mathbf{\Theta}_l^k) = \mathbf{X}_l \mathbf{\Gamma}_l^{k-1}(\mathbf{\Theta}_l^{k-1}) + \mathbf{Y}_l$.

Note that the Jacobian matrix of (B7) is:

$$\mathfrak{F}_f = \mathbf{X}_l \mathfrak{F}_\Gamma, \tag{B8}$$

where $\mathfrak{F}_\Gamma$ is the Jacobian of the load torque vector defined in (37). Note also from (B5) that $\mathfrak{F}_f$ and the value of α depend directly on the gear reduction ratio r.

https://doi.org/10.1017/S0263574714001544 Published online by Cambridge University Press