Robotics and Computer-Integrated Manufacturing 25 (2009) 756–769

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Constrained multi-objective trajectory planning of parallel kinematic machines

Amar Khoukhi [a,*], Luc Baron [b], Marek Balazinski [b]

[a] Department of Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran, 31261, Saudi Arabia
[b] Mechanical Engineering Department, École Polytechnique de Montréal, C. P. 6079, Succ. CV, Montreal QC, Canada H3C 3A7

ARTICLE INFO

Article history:
Received 30 November 2006
Received in revised form
23 August 2007
Accepted 29 September 2008

Keywords:
Parallel kinematic machines
Constrained off-line programming
Nonlinear optimal control
Time–energy trajectory planning
Augmented lagrangean
Decoupling

ABSTRACT

This paper presents a new approach to multi-objective dynamic trajectory planning of parallel kinematic machines (PKM) under task, workspace and manipulator constraints. The robot kinematic and dynamic model, (including actuators) is first developed. Then the proposed trajectory planning system is introduced. It minimizes electrical and kinetic energy, robot traveling time separating two sampling periods, and maximizes a measure of manipulability allowing singularity avoidance. Several technological constraints such as actuator, link length and workspace limitations, and some task requirements, such as passing through imposed poses are simultaneously satisfied. The discrete augmented Lagrangean technique is used to solve the resulting strong nonlinear constrained optimal control problem. A decoupled formulation is proposed in order to cope with some difficulties arising from dynamic parameters computation. A systematic implementation procedure is provided along with some numerical issues. Simulation results proving the effectiveness of the proposed approach are given and discussed.

1. Introduction

The design of parallel kinematic machines (PKMs) dates back to the pioneer work by Gough [1], who established the basic principles of a manipulator in a closed loop structure. The machine was able to position and orientate an end-effector (EE), such that to test tire wear and tear. A decade later, Stewart [2] proposed a platform manipulator for the use as an aircraft simulator. Since then, extensive research efforts lead to the realization of several robots and machine tools with parallel kinematic structures [3]. PKMs have two basic advantages over conventional machines of serial kinematic structures. First, the connection between the base and the EE is made with several kinematic chains. This results in high structural stiffness and rigidity. Second, with such structure, it is possible to mount all drives on or near the base. This results in large payloads capability and low inertia. Indeed, the ratio of payload to the robot load is usually about 1/10 for serial robots, while only 1/2 for parallel ones. Despite these advantages, PKMs are still rare in the industry. Among the major reasons for this gap are the small workspace, complex transformations between joint and Cartesian space and singularities as compared to their serial counterparts. These facts lead to a tremendous amount of research in PKMs design and customization [3]. Another reason recently identified is the under consideration of the dynamics of these machines [4]. The mentioned architecture-dependent performance associated with the strong coupled nonlinear dynamics makes the trajectory planning and control system design for PKMs more difficult, as compared to serial machines. In fact, for serial robots, there is a plentiful literature published on the topics of off-line and online programming, from both stand points: computational geometry and kinematics, and optimal control including robots dynamics [5–8]. For PKMs, a relatively large amount of literature is devoted to the computational kinematics and workspace optimization issues. The overwhelm criteria considered for PKMs trajectory planning are essentially design-oriented. These include singularity avoidance and dexterity optimization [9–13]. In Ref. [14], the authors had developed a clustering scheme to isolate and avoid singularities and obstacles for a PKM path planning. A kinematic design and planning method had been described in Ref. [15] for a four-bar planar manipulator mechanism. Another related work was considered in Ref. [16], where it had been shown that a motion planning with singularity-free pose change is possible for PKMs. A variational approach is reported in Ref. [17] for planning a singularity-free minimum-energy path between two end-points for Gough–Stewart platforms. This method is based on a penalty optimization method. Penalty methods, however, have several

* Corresponding author.
E-mail addresses: amar@kfupm.edu.sa (A. Khoukhi), luc.baron@polymtl.ca (L. Baron), marek.balazinski@polymtl.ca (M. Balazinski).

0736-5845/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2008.09.002

Nomenclature

B	reference frame attached to the center of mass of the base
A	reference frame attached to the center of mass of the end-effector (EE)
A_i, B_i	ith attachment point of leg i on body A and B
$\mathbf{p} = [x\ y\ z]^{\mathrm{T}}$	position vector of the origin of A relative to B in B
$\dot{\mathbf{p}} = [\dot{x}\ \dot{y}\ \dot{z}]^{\mathrm{T}}$	velocity vector of the origin of A relative to B
$\boldsymbol{x}_1 = \mathbf{q} = [\mathbf{p}^{\mathrm{T}}\ \varphi\ \theta\ \psi]^{\mathrm{T}}$	position and orientation of A in B
$\dot{\mathbf{q}}_{\mathrm{E}} = [\dot{\mathbf{p}}^{\mathrm{T}}\ \ \dot{\varphi}\ \ \dot{\theta}\ \ \dot{\psi}]^{\mathrm{T}}$	time derivatives of $\boldsymbol{x}_1(t)$
$\boldsymbol{x}_2 = \dot{q} = [\mathbf{p}^{\mathrm{T}}\ \ \boldsymbol{\omega}^{\mathrm{T}}]^{\mathrm{T}}$	Cartesian and angular velocity of the EE
$\boldsymbol{x} = [\boldsymbol{x}_1\ \boldsymbol{x}_2]^{\mathrm{T}}$	continuous-time state of the PKM
$\boldsymbol{x}_k = [\boldsymbol{x}_{1k}\ \boldsymbol{x}_{2k}]^{\mathrm{T}}$	discrete-time state of the PKM
$\tau(t)$	Cartesian force/torques wrench
$\boldsymbol{i} = [i_1\ i_2 \ldots i_6]^{\mathrm{T}}$	vector of electric currents
$\boldsymbol{l} = [l_1\ l_2 \ldots l_6]^{\mathrm{T}}$	vector of the link lengths
$\boldsymbol{J}$	Jacobian matrix of the PKM
$\boldsymbol{M}_j(\boldsymbol{q})$, $\boldsymbol{M}_c(\boldsymbol{q})$	inertia matrix expressed in joint and Cartesian space
$\boldsymbol{N}_j(\boldsymbol{q},\dot{\boldsymbol{q}})$, $\boldsymbol{N}_c(\boldsymbol{q},\dot{\boldsymbol{q}})$	coriolis and centrifugal force/torque in joint and Cartesian space
$\boldsymbol{G}_j(\boldsymbol{q})$, $\boldsymbol{G}_c(\boldsymbol{q})$	gravity force in joint and Cartesian space
$\boldsymbol{M}_a$, M_a	actuator inertia matrix and its component
$\boldsymbol{V}_a$, V_a	actuator viscous damping coefficient matrix and its component
$\boldsymbol{K}_a$, K_a	actuator gain matrix and its component
$\boldsymbol{K}$	control law gain matrix
τ_m	joint torque vector produced by the DC motors
p	ballscrew pitch
n	gear ratio
J_s, J_m	ballscrew and motor mass moments of inertia
b_s, b_m	ballscrew and motor viscous damping coefficients
$\boldsymbol{\lambda}$	Lagrangian multipliers (or co-states) associated to state variables
$(\boldsymbol{\rho}, \boldsymbol{\sigma})$	Lagrangian multipliers associated to inequality and equality constraints
$(\boldsymbol{\mu}_g, \boldsymbol{\mu}_s)$	penalty coefficients associated to inequality and equality constraints
N	total number of discretisations of the trajectory
w^*, η^*, η^*_1	cost minimization, equality and inequality constraints optimal tolerances
w_t, η_t, η_{t1}	cost minimization, equality and inequality constraints feasible tolerances

drawbacks [18]. Another major issue for off-line programming and practical use of PKMs in industry (in a machining process, for example) is that for a prescribed tool path in the workspace, the control system should guarantee the prescribed task completion within the workspace, for a given set up of the EE (i.e., for which limitations on actuator lengths and physical dimensions are not violated). This problem has been recently considered in Refs. [19,20], with design methodologies involving workspace limitations and actuator forces optimization using optimization techniques.

In this context, we consider a new integrated multi-objective dynamic trajectory planning system for PKMs. Part of this work has been presented in Refs. [21–23]. The proposed approach considers PKM's dynamics, including actuators models as well as task and workspace requirements, as a unique entity. It can be encapsulated into two levels (see Fig. 1): the modeling and approach level and the simulation and testing level [7,23]. The former consists to select according to performance targets related to the robot, task and workspace interactions, the appropriate models and control approaches in order to optimize an overall performance of the robot–task–workspace system. The second level is devoted to coding, testing and validation. Criteria to be optimized in this study are time, energy and a measure of manipulability necessary for a task execution. The optimization procedure is performed within a proper balance between time and energy minimization, singularity avoidance, actuators, sampling periods, link lengths and workspace limitations, and task constraints satisfaction. From a state-space representation by a system of differential equations in the phase plane, the trajectory planning is formulated within a variational calculus framework. The resulting constrained nonlinear programming problem is solved using an augmented Lagrangian (AL) with decoupling technique. AL algorithms have proven to be robust and powerful to cope with difficulties related to none strictly convex constraints [24–27] as compared to optimization methods employing only penalty. The decoupling technique is introduced in order to solve some difficult computations in the original nonlinear and coupled formulation. Another advantage of the proposed method is that one might introduce several criteria and constraints to satisfy in the trajectory planning process.

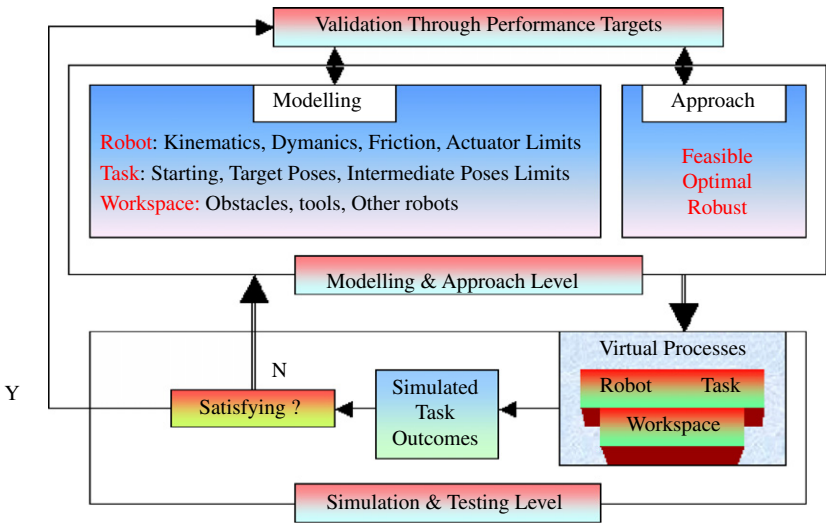

Fig. 1. Overall off-line programming framework of PKMs.

Section 2 introduces the kinematic and dynamic models using Euler–Lagrange formulation, and gives the associated discrete-time state-space model. In Section 3, the constrained nonlinear optimal control problem is formulated. In Section 4, the AL with decoupling technique is developed to solve the resulting linear and decoupled optimal control problem. In Section 5, an implementation on a case study 2-degrees-of-freedom (DOF) planar PKM is provided. Finally, Section 6 concludes this work.

2. Modeling

2.1. Kinematic model

The PKM shown in Fig. 2 represents a full 6-DOF motion of its EE from its articulated motion of its six leg lengths. The *pose*, i.e., position and orientation, of the EE, namely, $\mathbf{q} = [\mathbf{p}^{\mathrm{T}}\ \varphi\ \theta\ \psi]^{\mathrm{T}}$ can be expressed in B by $\mathbf{p}$ the position vector of the origin of A relative to B, and the set of Euler angles (ϕ, θ, ψ) defining the orientation of the EE in B, namely, ${}^{\mathrm{B}}\boldsymbol{R}_{\mathrm{A}}$, as used to uniquely determine the EE orientation noted as $\boldsymbol{R}_{\mathrm{A}}$.

This rotation matrix is given as

$$
\begin{aligned}
{}^{\mathrm{B}}\boldsymbol{R}_{\mathrm{A}} &= (\boldsymbol{R}_x(\varphi), \boldsymbol{R}_y(\theta),\ \boldsymbol{R}_z(\psi)) \\
&= \begin{bmatrix} c\psi c\phi - c\theta s\phi s\psi & -s\psi c\phi - c\theta s\phi c\psi & s\theta s\phi \\ c\psi s\phi + c\theta c\phi s\psi & -s\psi s\phi + c\theta c\phi c\psi & -s\theta c\phi \\ s\psi s\phi & c\psi s\theta & c\theta \end{bmatrix},
\end{aligned} \tag{1}
$$

where c and s stand for the cosine and sine functions, respectively. Clearly, the orientation of A is described with respect to B by a rotation matrix ${}^{\mathrm{B}}\boldsymbol{R}_{\mathrm{A}} = [r_1\ \ r_2\ \ r_3]$, where r_1, r_2 and r_3 are, respectively, 3×1 unit vectors along the axes of A.

The velocity of the EE, namely, $\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix}$, can be obtained as a function of the time derivative of $\mathbf{q}$, i.e.,

$$
\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3\times 3} & O_{3\times 3} \\ O_{3\times 3} & \mathbf{R}(\varphi, \theta, \psi) \end{bmatrix} \dot{\mathbf{q}} \tag{2}
$$

where the transformation between the angular velocity $\boldsymbol{\omega}$ and the time derivatives of the Euler angles $\dot{\varphi}\ \dot{\theta}\ \dot{\psi}$ is given as [28,29]

$$
\boldsymbol{R}(\varphi, \vartheta, \psi) = \begin{bmatrix} 0 & c\varphi & s\varphi c\theta \\ 0 & s\varphi & -c\varphi s\theta \\ 1 & 0 & c\theta \end{bmatrix} \tag{3}
$$

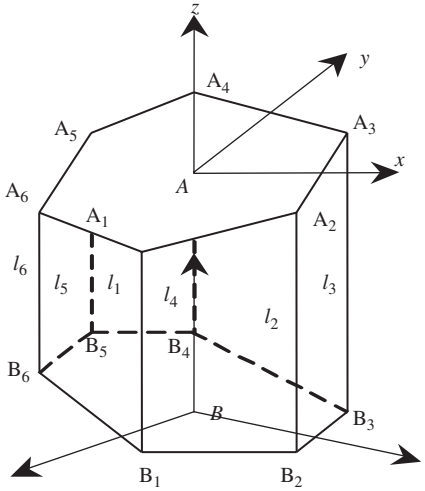

Fig. 2. Geometry of a PKM.

There are two basic problems in kinematic modeling; the *forward kinematics* is the determination of the EE motion from a given motion of the leg lengths, while the *inverse kinematics* is the determination of the leg lengths motion from a given EE motion. Below these two kinematic problems are formulated at the velocity level.

2.1.1. Inverse rate kinematics

The closure of each kinematic loop passing through the origin of A and B, and through the six attachment points B_i on the base and the hip attachment points A_i on the EE is given as

$$
\boldsymbol{a}_i = \mathbf{p} + {}^{\mathrm{B}}\boldsymbol{R}_{\mathrm{A}}\,{}^{\mathrm{A}}\boldsymbol{a}_i, \quad i = 1, \ldots, 6, \tag{4}
$$

where ${}^{\mathrm{A}}\boldsymbol{a}_i$ is the constant position vector of A_i in A. By differentiating Eq. (4) with respect to time, projecting along the joint axis and grouping in a matrix form, one gets the inverse kinematic model as

$$
\dot{\boldsymbol{l}} = \boldsymbol{J}^{-1}\dot{\boldsymbol{q}} \tag{5}
$$

where $\dot{\boldsymbol{l}}$ is the actuated leg length velocity, $\boldsymbol{J}^{-1}$ is the inverse Jacobian matrix given as

$$
J^{-1} = \begin{pmatrix} e_1^{\mathrm{T}} & ({}^{\mathrm{B}}\boldsymbol{R}_{\mathrm{A}}\,{}^{\mathrm{A}}a_1 \times e_1)^{\mathrm{T}} \\ \\ e_6^{\mathrm{T}} & ({}^{\mathrm{B}}\boldsymbol{R}_{\mathrm{A}}\,{}^{\mathrm{A}}a_6 \times e_6)^{\mathrm{T}} \end{pmatrix} \tag{6}
$$

and $\boldsymbol{e}_i$ is a unit vector along the ith leg axis.

2.1.2. Forward kinematic model

Unlike the inverse kinematic problem, the forward kinematics is more challenging for general PKMs. The number of solutions depends on the number of configurations the mechanism can be assembled into, for a given set of link lengths. Eq. (5) representing the inverse kinematic solution cannot be inverted to find $\boldsymbol{q}$ for a given $\boldsymbol{l}$, because $\boldsymbol{q}$ does not explicitly occur in Eq. (5). Numerical methods are generally used to solve the forward kinematic problem [28,29]. In this paper, a Newton method is used [26].

2.2. Dynamic model

Likewise to the kinematic modeling, there are two basic problems in dynamic modeling [28,29]: the *forward and inverse dynamics*. The latter consists to find the joint force/torque from a given EE motion. While the former is to find the EE motion from a given joint input force/torque and initial position, velocity and acceleration conditions. For optimal control and trajectory planning purposes, the dynamic equations are derived using Euler–Lagrange formalism. In Cartesian space, the inverse dynamic model is given a canonical form as

$$
\boldsymbol{\tau} = \boldsymbol{M}_{\mathrm{c}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{N}_{\mathrm{c}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}_{\mathrm{c}}(\boldsymbol{q}) \tag{7}
$$

where $\boldsymbol{M}_{\mathrm{c}}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{N}_{\mathrm{c}}(\mathbf{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{G}_{\mathrm{c}}(\boldsymbol{q})$ are the Coriolis and centrifugal, and gravitational forces, respectively.

2.2.1. Actuators model

It has been shown that actuator dynamics are significant and cannot be neglected for simulation and control [28]. The dynamic equations of actuators yield

$$
\boldsymbol{M}_{\mathrm{a}}\ddot{\boldsymbol{l}} + \boldsymbol{V}_{\mathrm{a}}\dot{l} + \boldsymbol{K}_{\mathrm{a}}\boldsymbol{F} = \boldsymbol{\tau}_{\mathrm{m}} \tag{8}
$$

Where

$$\boldsymbol{M}_a = \mathrm{M_a}\boldsymbol{I}_{6\times6} = \frac{2\pi}{np}(J_s + n^2 J_m)\boldsymbol{I}_{6\times6}$$
$$\boldsymbol{V}_a = \mathrm{V}_a\boldsymbol{I}_{6\times6} = \frac{2\pi}{np}(b_s + n^2 b_m)\boldsymbol{I}_{6\times6}$$
$$\boldsymbol{K}_a = K_a\boldsymbol{I}_{6\times6} = \frac{p}{2\pi n}\boldsymbol{I}_{6\times6} \tag{9}$$

with the parameters described in the nomenclature.

Since the actuator's dynamics is written in joint space, it is necessary to transform the motion Eq. (7) to joint space, using Eq. (5), in order to include it

$$\boldsymbol{M}_j(\boldsymbol{q})\ddot{\boldsymbol{l}} + \boldsymbol{N}_j(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}_j(\boldsymbol{q}) = \boldsymbol{F} \tag{10}$$

where

$$\boldsymbol{M}_j(\boldsymbol{q}) = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{c}}(\boldsymbol{q})\boldsymbol{J},$$
$$\boldsymbol{N}_j(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{N}_{\mathrm{c}}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{c}}(\boldsymbol{q})\boldsymbol{J}\frac{\mathrm{d}(\boldsymbol{J}^{-1})}{\mathrm{d}t}\dot{\boldsymbol{q}}, \text{ and}$$
$$\boldsymbol{G}_j(\boldsymbol{q}) = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{G}_{\mathrm{c}}(\boldsymbol{q}) \tag{11}$$

with $\boldsymbol{J}$ being the Jacobian, and $\boldsymbol{F}$ the six-directional column of the input forces.

The combination of Eqs. (8) and (10) gives the dynamic model including the actuators in joint space as

$$\bar{\boldsymbol{M}}_j(\boldsymbol{q})\ddot{\boldsymbol{l}} + \bar{\mathrm{N}}_j(\boldsymbol{q},\dot{\boldsymbol{q}}) + \bar{\boldsymbol{G}}_j(\boldsymbol{q}) = \boldsymbol{\tau}_m \tag{12}$$

with

$$\bar{\boldsymbol{M}}_j(\boldsymbol{q}) = \boldsymbol{K}_{\mathrm{a}}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{c}}(\boldsymbol{q})\boldsymbol{J} + \boldsymbol{M}_{\mathrm{a}},$$
$$\bar{\boldsymbol{N}}_j(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{K}_{\mathrm{a}}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{N}_{\mathrm{c}}(\boldsymbol{q},\boldsymbol{q}) + \left(\boldsymbol{V}_{\mathrm{a}} - \boldsymbol{K}_{\mathrm{a}}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{c}}(\boldsymbol{q})\boldsymbol{J}\frac{\mathrm{d}(\boldsymbol{J}^{-1})}{\mathrm{d}t}\boldsymbol{J}\right)\boldsymbol{J}^{-1}\boldsymbol{q},$$
$$\bar{\boldsymbol{G}}_j(\boldsymbol{q}) = \boldsymbol{K}_{\mathrm{a}}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{G}_{\mathrm{c}}(\boldsymbol{q}) \tag{13}$$

The bars over uppercase boldface letters are used to include both manipulator and actuator elements. Going back to Cartesian space, the overall PKM dynamic model is given as

$$\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{q}) = \boldsymbol{\tau}_{\mathrm{m}} \tag{14}$$

with

$$\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{q}) = \boldsymbol{K}_{\mathrm{a}}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{c}}(\boldsymbol{q}) + \boldsymbol{M}_{\mathrm{a}}\boldsymbol{J}^{-1},$$
$$\bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{q},\boldsymbol{q}) = \boldsymbol{K}_{\mathrm{a}}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{N}_{\mathrm{c}}(\boldsymbol{q},\boldsymbol{q}) + \left(\boldsymbol{V}_{\mathrm{a}}\boldsymbol{J}^{-1} + \boldsymbol{M}_{\mathrm{a}}^{\mathrm{T}}(\boldsymbol{q})\frac{\mathrm{d}(\boldsymbol{J}^{-1})}{\mathrm{d}t}\right)\boldsymbol{q},$$
$$\bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{q}) = \boldsymbol{K}_{\mathrm{a}}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{G}_{\mathrm{c}}(\boldsymbol{q}) \tag{15}$$

It is noteworthy that in the proposed approach one might include contact effort models. Among such models, there are friction and other application-specific forces. Such inclusion is very useful in many practical cases as deflashing and screwing, as it allows avoiding actuator saturation and improving the trajectory planning performance.

2.2.2. Discrete-time dynamic model

The approximate state-space discrete-time model of the PKM is deducted from a state-space form of the continuous-time dynamic model. Without loss of generality and for the sake of writing simplicity, the time index and the contact forces are omitted. So, one might write Eq. (14) as

$$\ddot{\boldsymbol{q}} = \bar{\boldsymbol{M}}_{\mathrm{c}}^{-1}(\boldsymbol{q})\boldsymbol{\tau}_{\mathrm{m}} - \bar{\boldsymbol{M}}_{\mathrm{c}}^{-1}(\boldsymbol{q})[\bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{q})] \tag{16}$$

By using state $\boldsymbol{x}_1$, and its time derivative $\boldsymbol{x}_2$, i.e., $\boldsymbol{x} = [\boldsymbol{x}_1^{\mathrm{T}}, \boldsymbol{x}_2^{\mathrm{T}}]^{\mathrm{T}}$ (defined in the nomenclature), Eq. (16) is transformed as

$$\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_1,\boldsymbol{x}_2)\boldsymbol{x}_2 + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_1) = \boldsymbol{\tau}_{\mathrm{m}} \tag{17}$$

In turns, one might rewrite Eq. (17) as follows:

$$\dot{\boldsymbol{x}} = \begin{bmatrix}\boldsymbol{O}_{6\times6} & \boldsymbol{I}_{6\times6}\\ \boldsymbol{O}_{6\times6} & \boldsymbol{O}_{6\times6}\end{bmatrix}\boldsymbol{x} - \begin{bmatrix}0_{6\times1}\\ \bar{\boldsymbol{M}}_{\mathrm{c}}^{-1}(\boldsymbol{x}_1)[\bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_1,\boldsymbol{x}_2)\boldsymbol{x}_2 + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_1)]\end{bmatrix} + \begin{bmatrix}\boldsymbol{O}_{6\times6}\\ \bar{\boldsymbol{M}}_{\mathrm{c}}^{-1}(\boldsymbol{x}_1)\end{bmatrix}\boldsymbol{\tau}_{\mathrm{m}} \tag{18}$$

with

$$\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_1) = \boldsymbol{K}_{\mathrm{a}}\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{x}_1)\boldsymbol{M}_{\mathrm{c}}(\boldsymbol{x}_1) + \boldsymbol{M}_{\mathrm{a}}\boldsymbol{J}^{-1}(\boldsymbol{x}_1)$$

and

$$\bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_1,\boldsymbol{x}_2) = \boldsymbol{K}_{\mathrm{a}}\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{x}_1) + \left(\boldsymbol{V}_{\mathrm{a}}\boldsymbol{J}^{-1}(\boldsymbol{x}_1) + \boldsymbol{M}_{\mathrm{a}}\frac{\mathrm{d}(\boldsymbol{J}^{-1}(\boldsymbol{x}_1))}{\mathrm{d}t}\right)\boldsymbol{x}_2$$

In order to derive the discrete dynamic model of the robot, Eq. (18) is written in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{F}\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{\tau} \tag{19}$$

with

$$\boldsymbol{F} = \begin{bmatrix}\boldsymbol{O}_{6\times6} & I_{6\times6}\\ \boldsymbol{O}_{6\times6} & \boldsymbol{O}_{6\times6}\end{bmatrix}, \quad \boldsymbol{D}(\boldsymbol{x}) = \begin{bmatrix}0_{6\times1}\\ \bar{\boldsymbol{M}}_{\mathrm{c}}^{-1}(\boldsymbol{x})[\bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}) + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x})]\end{bmatrix},$$

and

$$\boldsymbol{B}(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{O}_{6\times6}\\ \bar{\boldsymbol{M}}_{\mathrm{c}}^{-1}(\boldsymbol{x})\end{bmatrix} \tag{20}$$

Now, let us define the sampling period h_k, such that $h_k < t < h_{k+1}$ and $\sum_{k=1}^{\mathrm{N}} h_k = T$, with T being the total traveling time and the robot state being defined between two sampling points k and $k+1$ as

$$\boldsymbol{x}(t) = \boldsymbol{x}(h_k), \quad \text{for } k = 1, 2, \ldots, N. \tag{21}$$

The equivalent discrete-time model to Eq. (19) is given as [8]

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_{\mathrm{d}}(h_k)\boldsymbol{x}_k - \boldsymbol{D}_{\mathrm{d}}(\boldsymbol{x}_k,h_k) + \boldsymbol{B}_{\mathrm{d}}(\boldsymbol{x}_k,h_k)\boldsymbol{\tau}_k \tag{22}$$

where $\boldsymbol{F}_{\mathrm{d}}$, $\boldsymbol{D}_{\mathrm{d}}$, $\boldsymbol{B}_{\mathrm{d}}$ are the discrete equivalents to $\boldsymbol{F}$, $\boldsymbol{D}$, $\boldsymbol{B}$ matrices. The relationships between these pair of matrices are

$$\boldsymbol{F}_d(h_k) = \boldsymbol{F}_d(k+1,k) = e^{\boldsymbol{F}h_k} \cong \begin{bmatrix}\boldsymbol{I}_{6\times6} & h_k\boldsymbol{I}_{6\times6}\\ \boldsymbol{O}_{6\times6} & \boldsymbol{I}_{6\times6}\end{bmatrix}$$
$$\boldsymbol{D}_d(\boldsymbol{x}_k,h_k) = \int_0^{h_k}\boldsymbol{F}_d(h_k-t)\boldsymbol{G}(h_k-t)(\boldsymbol{D}(\boldsymbol{x}_k))\mathrm{d}t \cong \bar{\boldsymbol{M}}_{\mathrm{c}}^{-1}(\boldsymbol{x}_{1k})\begin{bmatrix}\frac{h_k^2}{2}\boldsymbol{I}_{6\times6}\\ h_k\boldsymbol{I}_{6\times6}\end{bmatrix}[\bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k},\boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k})]$$
$$\boldsymbol{B}_d(\boldsymbol{x}_k,\boldsymbol{h}_k) = \int_0^{h_k}\boldsymbol{F}_d(h_k-t)\boldsymbol{B}(x_k)\mathrm{d}t = \begin{bmatrix}\frac{h_k^2}{2}\mathrm{I}_{6\times6}\\ h_k\mathrm{I}_{6\times6}\end{bmatrix}\bar{\boldsymbol{M}}_{\mathrm{c}}^{-1}(\boldsymbol{x}_{1k}) \tag{23}$$

Hence, the discrete-time state-space dynamic model of the PKM is developed with a second order of accuracy for the position and one order for the velocity, to finally be written in the following form:

$$\boldsymbol{x}_{k+1} = \begin{bmatrix}\boldsymbol{I}_{6\times6} & h_k\boldsymbol{I}_{6\times6}\\ \boldsymbol{O}_{6\times6} & \boldsymbol{I}_{6\times6}\end{bmatrix}x_k - \begin{bmatrix}\frac{h_k^2}{2}\boldsymbol{I}_{6\times6}\\ h_k\boldsymbol{I}_{6\times6}\end{bmatrix}[\bar{\boldsymbol{M}}_{\mathrm{c}}^{-1}(\boldsymbol{x}_{1k})[\bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k},\boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k})]] + \begin{bmatrix}\frac{h_k^2}{2}\boldsymbol{I}_{6\times6}\\ h_k\boldsymbol{I}_{6\times6}\end{bmatrix}\bar{\boldsymbol{M}}_{\mathrm{c}}^{-1}(\boldsymbol{x}_{1k})\boldsymbol{\tau}_k \tag{24}$$

3. Optimal time-energy trajectory planning problem

3.1. Constraints modeling

Simulating a robotics task requires taking into account several constraints; structural and geometric constraints, kinematic and dynamic parameter nominal values, such as limits on link lengths, velocities, accelerations and nominal torques supported by actuators. Some of these constraints are defined in joint space, while others are in task space.

3.1.1. Robots constraints

- Dynamic state equations: Eq. (24) can be rewritten for later easy use as

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \tag{25}$$

- Link intermediate length limits

$$\boldsymbol{l}_{\mathrm{Min}} < \boldsymbol{l}_k < \boldsymbol{l}_{\mathrm{Max}}, \quad \text{with } k = 0, 2, \ldots, N, \quad \text{and } l_{\mathrm{Max}} = \boldsymbol{\Theta}_{\mathrm{Max}}(\boldsymbol{x}) \tag{26}$$

- Singularity avoidance
Singularities are particular poses in which the robot becomes uncontrollable. Therefore, they are crucial for a successful trajectory planning system. The conditions characterizing singularities are difficult to find analytically for a general PKM, since an analytical expression for the determinant of $\boldsymbol{J}^{-1}$ is not available. Several studies had been dealt with this problem and many singularity avoidance algorithms were proposed [9–13,16,17,31–33]. Common kinematic performance index related singularity avoidance is the manipulability measure [27]. Accordingly, by defining the manipulability measure as

$$\mathbf{w} = \sqrt{\det(\boldsymbol{J}(\boldsymbol{x}_{1k})\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{x}_{1k}))} \tag{27}$$

The following singularity avoidance function can be used as

$$\varpi(\boldsymbol{x}_{1k}) = \frac{1}{\sqrt{\det(\boldsymbol{J}(\boldsymbol{x}_{1k})\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{x}_{1k}))}} \tag{28}$$

- Torque limits
Non violation of control torque limits is another major issue for trajectory planning. The required leg forces must continuously be checked for possible violation of the limits as the manipulator moves close to a singular pose. As soon as any leg actuator crosses its limit, the optimal planning procedure has to determine an alternate leg actuation strategy leading to another path on which the actuator forces would be constrained within the limits. In this paper, the robot torques are assumed to belong to a compact and bounded set $\mathbf{C} \subset \Re^{6N}$, expressed as

$$\mathbf{C} = \{\boldsymbol{\tau}_k \in \Re^{6N}, \quad \text{such that} : \boldsymbol{\tau}_{\min} \leqslant \boldsymbol{\tau}_k \leqslant \boldsymbol{\tau}_{\max}, \quad k = 0, \ldots, N-1\} \tag{29}$$

- Sampling period limits
If the overall robot traveling time T is too small, there may be no admissible solution to the optimal control problem, qsince the torque constraints bound indirectly the path traversal time. On the other hand, the sampling period h_k must be smaller than the system smallest time constant in order to prevent the system from being uncontrollable between two control times. In this paper, a tradeoff is made through variation of the sampling period within an admissible domain H defined as

$$H = \{h_k \in \Re^{+}, \quad \text{such that} : h_{\min} \leqslant h_k \leqslant h_{\max}\} \tag{30}$$

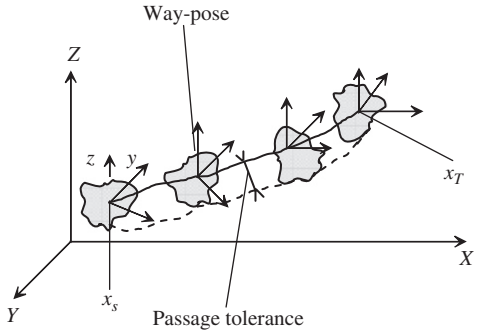

Fig. 3. Illustration of EE passage through imposed poses (positions and orientations). —— optimal path, ---- feasible path, $\boldsymbol{x}_s$ Starting pose, $\boldsymbol{x}_T$ Target pose

3.1.2. Task and workspace constraints

Task and workspace constraints are basically geometric and kinematic, and allow the determination of the size and shape of the manipulator workspace, which defines the set of poses that can be reached by the EE without singularity or link interference [19,20]. In this paper, these constraints are expressed by imposing the EE to pass through a set of specified poses (Fig. 3). These poses are quantified by a set of L pairs $(\mathbf{p}_l, \boldsymbol{R}_l)$ with $\mathbf{p}_l$ referring to the Cartesian position, and $\boldsymbol{R}_l$ to the orientation of the lth imposed pose on the EE, such that

$$||\mathbf{p} - \mathbf{p}_l|| - T_{\mathrm{PassTh}l_p} = 0, \tag{31}$$

and

$$||\mathbf{vect}(\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}_l)|| - T_{\mathrm{PassTh}l_R} = 0 \tag{32}$$

where $(\mathbf{p},\boldsymbol{R})$ describes the current computed pose of the EE, while **vect**(.) is the axial vector of its 3×3 matrix argument, and measures the absolute value of the angle of rotation between $\boldsymbol{R}$ and $\boldsymbol{R}_l$. These constraints represent equality constraints and are written for simplicity as

$$\begin{aligned} &\boldsymbol{s}_1^l(\boldsymbol{x}) = ||p - p_l|| - T_{\mathrm{PassTh}lp} = 0, \\ &\boldsymbol{s}_2^l(\boldsymbol{x}) = ||\mathbf{vect}(\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}_l)|| - T_{\mathrm{PassTh}l_R} = 0, \quad l = 1, \ldots, L \end{aligned} \tag{33}$$

The above inequality constraints are written in the following simplified forms:

$$\begin{aligned} &g_1(x) = \boldsymbol{l}_{\mathrm{Min}} - \boldsymbol{\Theta}(x) \leqslant 0, \quad g_2(x) = \boldsymbol{\Theta}(x) - \boldsymbol{l}_{\mathrm{Max}} \leqslant 0, \\ &\boldsymbol{g}_3(\tau) = \boldsymbol{\tau}_{\mathrm{Min}} - \boldsymbol{\tau} \leqslant 0, \quad \boldsymbol{g}_4(\tau) = \boldsymbol{\tau} - \boldsymbol{\tau}_{\mathrm{Max}} \leqslant 0 \end{aligned} \tag{34}$$

For the sake of development simplicity, all inequality constraints will be noted as $\mathbf{g}_j(\boldsymbol{x},\boldsymbol{\tau},h) \leqslant 0$, $j = 1,\ldots,4$, regardless if they depend only on state, control variables or both. Hence, we turn up with $J = 24$ inequality constraints, $2L$ equality constraints (imposed passages) and 12 equality constraints representing state dynamics equations.

3.2. Performance index

In general, it is possible to optimize any cost function that has a physical sense. It can be specified according to task and performance targets. The performance index considered in this paper, relates energy consumption, traveling time and singularity avoidance. For energy criterion, both electric and kinetic energies are optimized. For time criterion, there are two basic ways to perform optimization: the first one assumes a fixed sampling period h and searches for a minimum number N of discretisations. This is equivalent to bring the robot from an initial pose $\boldsymbol{x}_{\mathrm{S}}$ to a final pose $\boldsymbol{x}_{\mathrm{T}}$, within a minimum number N of switching steps. For the unconstrained case, the time optimal control is basically

bang–bang with singularities occurring at the vicinity of the switching function. In serial robots literature, there are several publications following this approach [30]. For strongly nonlinear and coupled mechanical systems (like for the PKM at hand), this is simply impractical, even by using symbolic calculation. The second approach fixes the number of discretisations N and varies the sampling periods h_k. This is equivalent to bring the robot from an initial pose $\boldsymbol{x}_S$ to a final pose $\boldsymbol{x}_T$, within a fixed number of steps N while varying (minimizing) the sampling periods. In this paper, the number of sampling periods is guessed from an initial feasible kinematic solution. Then the sampling periods and the actuator torques are taken as control variables. For singularity avoidance, it is included through maximization of Eq. (28). In continuous-time, the constrained optimal control problem can be stated as follows: among all admissible control sequences $(\boldsymbol{\tau}(t), h) \in C \times H$, that allow the robot to move from an initial state $\boldsymbol{x}(t_0) = \boldsymbol{x}_S$ to a final state $\boldsymbol{x}(t_f) = \boldsymbol{x}_T$, find those which minimize the cost function E:

$$E = \underset{\substack{\boldsymbol{\tau}(t)\in C \\ t_0, t_T \in H}}{\mathrm{Min}} \int_{t_0}^{t_T} \left\{ \left[\boldsymbol{\tau}(t)\boldsymbol{U}\boldsymbol{\tau}^{\mathrm{T}}(t) + \boldsymbol{\iota}_1 + \frac{1}{2} x_2(t)\boldsymbol{Q}\boldsymbol{x}_2^{\mathrm{T}}(t) + \boldsymbol{\delta}\varpi(\boldsymbol{x}_1(t)) \right] \mathrm{d}t \right\} \tag{35}$$

subject to constraints Eqs. (25), (26), (29)–(32), with C, H $\boldsymbol{U}$, $\boldsymbol{Q}$, $\boldsymbol{\iota}_1$ and $\boldsymbol{\delta}$ are, respectively, the set of admissible torques, the set of admissible sampling periods, the electric energy, the kinetic energy, the time level-headedness, and a weight factor for singularity avoidance. The corresponding discrete-time optimal control problem consists of finding the optimal sequences ($\boldsymbol{\tau}_1$, $\boldsymbol{\tau}_2$, …, $\boldsymbol{\tau}_N$) and (h_1, h_2, …, h_N), allowing the robot to move from an initial state $\boldsymbol{x}_0 = \boldsymbol{x}_S$ to a target state $\boldsymbol{x}_N = \boldsymbol{x}_T$, while minimizing the cost E_d

$$\underset{\substack{\boldsymbol{\tau}\in C \\ h\in H}}{\mathrm{Min}} E_d = \left\{ \sum_{k=1}^{N} [\boldsymbol{\tau}_k \boldsymbol{U} \boldsymbol{\tau}_k^{\mathrm{T}} + \boldsymbol{\iota}_1 + x_{2k}\boldsymbol{Q}\boldsymbol{x}_{2k}^{\mathrm{T}} + \boldsymbol{\delta}\varpi(\boldsymbol{x}_{1k})] h_k \right\} \tag{36}$$

subject to

$$\begin{aligned} &\boldsymbol{x}_{k+1} = f_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k), \quad k = 0, \ldots, N-1 \\ &\boldsymbol{g}_j(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \leqslant 0, \quad j = 1, \ldots, 4, \quad k = 0, \ldots, N-1 \\ &\boldsymbol{s}_i(\boldsymbol{x}_k) = 0, \quad i = 1, \ldots, 2L, \quad k = 0, \ldots, N \end{aligned}$$

4. Nonlinear programming formulation

4.1. AL approach

In the course of solving the constrained nonlinear multi-objective optimal control problem Eq. (36), the AL function is used to transform it into a non-constrained one, where the degree of penalty for violating the constraints is regulated by penalty parameters. This method was originated independently by Powell and Hestens [34,35]. It was subsequently improved by several authors [24–26]. It basically relies on quadratic penalty methods, but reduces the possibility of ill conditioning of the sub-problems that are generated with penalization by introducing explicit Lagrange multipliers estimates at each step into the function to be minimized, which results in a super linearly convergence iterates. Furthermore, while the ordinary Lagrangean is generally nonconvex (in the presence of nonconvex constraints like for the considered problem), AL might be convexified to some extent with a judicious choice of the penalty coefficients [26]. An outline of AL implementation procedure for the case at hand is given at the end of Section 5, and a flowchart diagram implementation appears in the appendix. The AL function transforming the constrained optimal control problem to an unconstrained one is written as

$$\begin{aligned} L_{\boldsymbol{\mu}}(x, \boldsymbol{\tau}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \boldsymbol{\sigma}) = &\sum_{k=1}^{N} [\boldsymbol{\tau}_k^{\mathrm{T}} \boldsymbol{U} \boldsymbol{\tau}_k + \boldsymbol{\iota}_1 + x_{2k}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{x}_{2k} + \boldsymbol{\delta}\varpi(\boldsymbol{x}_{1k})] h_k \\ &+ \sum_{k=0}^{N-1} \{\boldsymbol{\lambda}_{k+1}^{\mathrm{T}} (\boldsymbol{x}_{k+1} - \boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k))\} \\ &+ \sum_{k=0}^{N-1} h_k \left[\sum_{l=1}^{L-1} \sum_{i=1}^{2} \boldsymbol{\Psi}_{\boldsymbol{\mu}_s}(\boldsymbol{\sigma}_k^i, \boldsymbol{s}_i^l(\boldsymbol{x}_k)) \right. \\ &\left. + \sum_{j=1}^{\mathrm{J}} \boldsymbol{\Phi}_{\boldsymbol{\mu}_g}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j(x_k, \boldsymbol{\tau}_k, h_k)) \right] \\ &\times \sum_{i=1}^{2} h_N \boldsymbol{\Psi}_{\boldsymbol{\mu}_s}(\boldsymbol{\sigma}_N^i, \boldsymbol{s}_i^L(\boldsymbol{x}_N)) \end{aligned} \tag{37}$$

where the function $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$is defined by the discrete state Eq. (25) at the sampling time k, N is the sampling number, $\boldsymbol{\lambda} \in R^{12N}$ designates the adjoint (or co-state) obtained from the adjunct equations associated to state equations, $\boldsymbol{\rho}$, $\boldsymbol{\sigma}$ are Lagrange multipliers with appropriate dimensions, associated to equality and inequality constraints and $\boldsymbol{\mu}_g\boldsymbol{\mu}_S$ are the corresponding penalty coefficients. The penalty functions adopted here combine penalty and dual methods. This allows relaxation of the inequality constraints as soon as they are satisfied. Typically, these penalty functions are defined by

$$\begin{aligned} &\Psi_{\boldsymbol{\mu}_s}(\boldsymbol{a}, \boldsymbol{b}) = \left(\boldsymbol{a} + \frac{\boldsymbol{\mu}_s}{2}\boldsymbol{b}\right)^{\mathrm{T}} \boldsymbol{b} \text{ and} \\ &\boldsymbol{\Phi}_{\boldsymbol{\mu}_g}(\boldsymbol{a}, \boldsymbol{b}) = \frac{1}{2\boldsymbol{\mu}_g} \{||\mathrm{Max}(0, \boldsymbol{a} + \boldsymbol{\mu}_g \boldsymbol{b})||^2 - ||\boldsymbol{a}||^2\} \end{aligned} \tag{38}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ refer, respectively, to Lagrange multipliers and the left hand side of equality and inequality constraints.

The Karush–Kuhn–Tucker first-order optimality necessary conditions require that for $\boldsymbol{x}_k$, $\boldsymbol{\tau}_k$, h_k, $k = 0, \ldots, N$ to be solution to the problem, there must exist some positive Lagrange multipliers ($\boldsymbol{\lambda}_k$,$\boldsymbol{\rho}_k$), unrestricted sign multipliers $\boldsymbol{\sigma}_k$, and finite positive penalty coefficients ($\boldsymbol{\mu}_g$,$\boldsymbol{\mu}_s$) such that

$$\begin{aligned} &\frac{\partial L_{\boldsymbol{\mu}}}{\partial \boldsymbol{x}} = 0, \quad \frac{\partial L_{\boldsymbol{\mu}}}{\partial \boldsymbol{\tau}} = 0, \quad \frac{\partial L_{\boldsymbol{\mu}}}{\partial h} = 0, \quad \frac{\partial L_{\boldsymbol{\mu}}}{\partial \boldsymbol{\lambda}} = 0, \quad \frac{\partial L_{\boldsymbol{\mu}}}{\partial \boldsymbol{\rho}} = 0, \quad \frac{\partial L_{\boldsymbol{\mu}}}{\partial \boldsymbol{\sigma}} = 0, \quad \text{and} \\ &\boldsymbol{\rho}_k^{\mathrm{T}} g(\boldsymbol{x}, \boldsymbol{\tau}, h) = 0, \quad \boldsymbol{\sigma}_k^{\mathrm{T}} s(\boldsymbol{x}) = 0, \quad g(\boldsymbol{x}, \boldsymbol{\tau}, h) \leqslant 0 \end{aligned} \tag{39}$$

The development of these conditions enables us to derive the iterative formulas to solve the optimal control problem by adjusting control variables, Lagrange multipliers as well as penalty coefficients. However, in Eq. (25), $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$ contains the inverse of the total inertia matrix $\bar{\boldsymbol{M}}_c^{-1}(\boldsymbol{x})$ of the PKM, including struts and actuators, as well as their Coriolis and centrifugal wrenches $\bar{\boldsymbol{N}}_c(\boldsymbol{x}_1, \boldsymbol{x}_2)$. These might take several pages long to display. In developing the first necessary optimality conditions and computing the co-states $\boldsymbol{\lambda}_k$, one has to determine the inverse of the mentioned inertia matrix and its derivatives with respect to state variables. This results in an intractable complexity even by using symbolic calculation.

4.2. Constrained linear-decoupled formulation

The major computational difficulty mentioned earlier cannot be solved by performing with the original nonlinear formulation. Instead, it is solved using a linear-decoupled formulation [36].

4.2.1. Theorem

Under the invertibility condition of the inertia matrix, the control law defined in the Cartesian space as

$$\boldsymbol{u} = \bar{\boldsymbol{M}}_c(\boldsymbol{x}_1)\boldsymbol{v} + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{x}_2 + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_1) \tag{40}$$

allows the robot to have a linear and decoupled behavior with a dynamic equation

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{v}, \tag{41}$$

where $\boldsymbol{v}$ is an auxiliary input.

4.2.2. Proof

It follows simply by substituting the proposed control law Eq. (40) into the dynamic model Eq. (14). One gets

$$\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 = \bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_1)\boldsymbol{v}$$

Since $\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x})$ is invertible, it follows that $\dot{\boldsymbol{x}}_2 = \boldsymbol{v}$

This brings the robot to have the decoupled and linear behavior described by the following linear dynamic equation written in discrete form as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_{\mathrm{d}k}\boldsymbol{x}_k + \boldsymbol{B}(h_k)(\boldsymbol{v}_k) = \boldsymbol{f}_{d_k}^{\mathrm{D}}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k) \tag{42}$$

with

$$\boldsymbol{f}_{d_k}^{\mathrm{D}}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k) = \begin{bmatrix} \boldsymbol{I}_{6\times6} & h_k\boldsymbol{I}_{6\times6} \\ \boldsymbol{O}_{6\times6} & \boldsymbol{I}_{6\times6} \end{bmatrix}\boldsymbol{x}_k + \begin{bmatrix} \frac{h_k^2}{2}\boldsymbol{I}_{6\times6} \\ h_k\boldsymbol{I}_{6\times6} \end{bmatrix}\boldsymbol{v}_k$$

Notice that this formulation reduces drastically the computations, by alleviating us the calculation at each iteration of the inertia matrix inverse and its derivatives with respect to state variables, which results in easy calculation of the co-states. The nonlinearity is, however, transferred to the objective function.

The decoupled formulation transforms the discrete optimal control problem into finding optimal sequences of sampling periods and acceleration inputs (h_1, h_2, …, h_N), ($\boldsymbol{v}_1$, $\boldsymbol{v}_2$, …, $\boldsymbol{v}_N$), allowing the robot to move from an initial state $\boldsymbol{x}_0 = \boldsymbol{x}_{\mathrm{S}}$ to a final state $\boldsymbol{x}_N = \boldsymbol{x}_{\mathrm{T}}$, while minimizing the cost function

$$\begin{aligned} E_d^{\mathrm{D}} = \min_{\substack{v\in V \\ h_k\in \mathrm{H}}} \Bigg\{ \sum_{k=0}^{N-1} [&[\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_{1k})v_k + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} \\ &+ \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k})]^T \boldsymbol{U}[\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} \\ &+ \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k})] + \imath_1 + \boldsymbol{x}_{2k}^T\boldsymbol{Q}\boldsymbol{x}_{2k} + \delta\varpi(\boldsymbol{x}_{1k})]h_k \Bigg\} \end{aligned} \tag{43}$$

and satisfying the above-mentioned constraints, which mainly remain the same, except for actuator torques, which become

$$\boldsymbol{\tau}_{\mathrm{Min}} \leqslant \bar{\boldsymbol{M}}_c(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k}) \leqslant \boldsymbol{\tau}_{\mathrm{Max}} \tag{44}$$

Moreover, inequality constraints $\boldsymbol{g}_3$ and $\boldsymbol{g}_4$ can be rewritten as

$$\begin{aligned} \boldsymbol{g}_3^{\mathrm{D}}(\boldsymbol{x}_k, \boldsymbol{v}_k) &= \boldsymbol{\tau}_{\mathrm{Min}} - [\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k})] \leqslant 0 \\ \boldsymbol{g}_4^{\mathrm{D}}(\boldsymbol{x}_k, \boldsymbol{v}_k) &= [\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k})] - \boldsymbol{\tau}_{\mathrm{Max}} \leqslant 0 \end{aligned} \tag{45}$$

Similarly to the non decoupled case, the decoupled problem might be written in the following form:

$$\min_{\substack{v\in V \\ h\in \mathrm{H}}} E_d^{\mathrm{D}}$$

subject to

$$\begin{aligned} &\boldsymbol{x}_{k+1} = f_{d_k}^{\mathrm{D}}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k), \quad k = 0, \ldots, N-1 \\ &\boldsymbol{g}_j^{\mathrm{D}}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k) \leqslant 0, \quad j \in \{1, 2, \ldots, J\} \\ &\boldsymbol{s}_i^{\mathrm{D}}(\boldsymbol{x}_k) = 0, \quad i \in \{1, \ldots, I\}, \quad k = 0, \ldots, N \end{aligned} \tag{P}$$

4.3. AL for the decoupled formulation

Now, *mutatis mutandis*, the AL associated to the decoupled formulation (P) is (after removing bars and *c* indexes for writing simplicity)

$$\begin{aligned} L_{\boldsymbol{\mu}}^{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{v}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \boldsymbol{\sigma}) = &\sum_{k=0}^{N-1} \{[[[\boldsymbol{M}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + N(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \boldsymbol{G}(\boldsymbol{x}_{1k})]^{\mathrm{T}}\boldsymbol{U} \\ &[\boldsymbol{M}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \boldsymbol{N}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \boldsymbol{G}(\boldsymbol{x}_{1k})]] + \imath_1 + \boldsymbol{x}_{2k}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}_{2k} + \delta\varpi(\boldsymbol{x}_{1k})]h_k\} \\ &+ \sum_{k=0}^{N-1} \{\boldsymbol{\lambda}_{k+1}^{\mathrm{T}}(\boldsymbol{x}_{k+1} - \boldsymbol{f}_{d_k}^{\mathrm{D}}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k))\} \\ &\sum_{k=0}^{N-1} h_k \left[\sum_{l=1}^{L-1} \sum_{i=1}^{2} \Psi_{\boldsymbol{\mu}_{\mathrm{s}}}(\boldsymbol{\sigma}_k^i, \boldsymbol{s}_i^{Dl}(\boldsymbol{x}_k)) + \sum_{j=1}^{J} \boldsymbol{\Phi}_{\boldsymbol{\mu}_{\mathrm{g}}}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j^{\mathrm{D}}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)) \right] \\ &+ \sum_{i=1}^{2} h_N \boldsymbol{\Psi}_{\boldsymbol{\mu}_{\mathrm{s}}}(\boldsymbol{\sigma}_N^i, \boldsymbol{s}_i^{DL}(\boldsymbol{x}_N)) \end{aligned} \tag{46}$$

where the function $f_{d_k}^{\mathrm{D}}(x_k, \boldsymbol{\tau}_k, h_k)$ is defined by Eq. (42) at time k, N is the total sampling number, other parameters appearing in Eq. (46) are defined above.

Again, the development of the first-order Karush–Kuhn–Tucker optimality necessary conditions require that for $\boldsymbol{x}_k$, $\boldsymbol{v}_k$, h_k, $k = 0, \ldots N$ to be solution to the problem (P), there must exist some positive Lagrange multipliers ($\boldsymbol{\lambda}_k$,$\boldsymbol{\rho}_k$), unrestricted sign multipliers $\boldsymbol{\sigma}_k$, and finite positive penalty coefficients $\boldsymbol{\mu} = (\boldsymbol{\mu}_{\mathrm{g}}, \boldsymbol{\mu}_{\mathrm{s}})$, such that Eq. (39) are satisfied for the decoupled formulation.

The co-states $\boldsymbol{\lambda}_k$ are determined by backward integration of the adjunct state equation yielding

$$\begin{aligned} \boldsymbol{\lambda}_{k-1} = &-2h_k \frac{\partial[\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k})]}{\partial x_k} \\ &\times \boldsymbol{U}[\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k})] \\ &- 2\boldsymbol{Q}\boldsymbol{x}_{2k}h_k - \delta\nabla_{x_{1k}}\varpi(x_{1k}) - \boldsymbol{F}_{\mathrm{d}}^{\mathrm{T}}\boldsymbol{\lambda}_k \\ &- h_k \left[\sum_{l=1}^{L-1} \sum_{i=1}^{2} \nabla_{x_k} \Psi_{\boldsymbol{\mu}_{\mathrm{s}}}(\boldsymbol{\sigma}_k^i, s_i^{Dl}(x_k)) \right] \\ &- h_k \left[\sum_{j=1}^{j} \nabla_{x_k} \Phi_{\boldsymbol{\mu}_{\mathrm{g}}}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j^{D}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k)) \right], \quad k = N-1, \ldots, 1 \end{aligned} \tag{47}$$

The gradient of the Lagrangian with respect to sampling period variables is

$$\begin{aligned} \nabla_{hk} L_{\boldsymbol{\mu}}^{\mathrm{D}} = &[[\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_k)\boldsymbol{v}_k + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k})]^{\mathrm{T}} \\ &\times \boldsymbol{U}[\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k})] \\ &+ \boldsymbol{x}_{2k}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}_{2k} + \imath_1 + \delta\varpi(\boldsymbol{x}_{1k})] + \sum_{l=1}^{L-1} \sum_{i=1}^{2} \boldsymbol{\Psi}_{\boldsymbol{\mu}_{\mathrm{s}}}(\boldsymbol{\sigma}_k^i, \boldsymbol{s}_i^{Dl}(\boldsymbol{x}_k)) \\ &+ \sum_{j=1}^{j} \boldsymbol{\Phi}_{\boldsymbol{\mu}_{\mathrm{g}}}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j^{D}(\boldsymbol{x}_k, \boldsymbol{v}_k)) \end{aligned} \tag{48}$$

The gradient of the Lagrangian with respect to acceleration variables is

$$\begin{aligned} \nabla_{vk} L_{\boldsymbol{\mu}}^{\mathrm{D}} = &2\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_{1k})\boldsymbol{U}^{\mathrm{T}}[\bar{\boldsymbol{M}}_{\mathrm{c}}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_{\mathrm{c}}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_{\mathrm{c}}(\boldsymbol{x}_{1k})]h_k + \boldsymbol{Z}_k^{\mathrm{T}}\boldsymbol{\lambda}_k \\ &+ h_k \left[\sum_{j=1}^{j} \nabla_{v_k} \Phi_{\boldsymbol{\mu}_{\mathrm{g}}}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j^{\mathrm{D}}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k)) \right] \end{aligned} \tag{49}$$

where

$$\boldsymbol{Z}_k = \begin{bmatrix} \boldsymbol{I}_{6\times6} & h_k\boldsymbol{I}_{6\times6} \\ \boldsymbol{O}_{6\times6} & \boldsymbol{I}_{6\times6} \end{bmatrix}[x_k] + \begin{bmatrix} \frac{h_k^2}{2}\boldsymbol{I}_{6\times6} \\ h_k\boldsymbol{I}_{6\times6} \end{bmatrix}[\boldsymbol{v}_k], \quad k = 0, 2, \ldots, N-1$$

The development of various related expressions are quite long and not given here. They are detailed in Ref. [37].

4.4. Implementation issues

4.4.1. Initial solution

To fasten convergence of AL–although it converges even if it starts from an unfeasible solution–a kinematic-feasible solution is defined. It is based on a velocity profile. This solution is divided into three zones, acceleration zone with duration T_1. In this zone, the actuators are assumed to supply an initial force to accelerate the EE until the maximum velocity is reached. Then a constant velocity zone of duration T_2 is achieved. Finally a deceleration zone of duration $T_3 = T_1$ finishes the cycle. The initial time discretisation is assumed an equidistant grid for convenience, i.e.,

$$h_k = t_{k+1} - t_k = \frac{t_f - t_0}{N}, \quad k = 1, 2, \ldots, N-1 \tag{50}$$

4.4.2. Search direction update

Because the considered problem is of large scale type, to solve for the minimization step at the primal level of AL, a limited-memory Quasi-Newton-like method is used at each iteration of the optimization process. This method allows the computing of the approximate Hessian matrix by using only the first derivative information, and without need to storing of this approximated Hessian matrix. It performs the second-order BFGS (Broyden–Fletcher–Goldfarb–Shano) search technique. It is briefly outlined below. For more details, the reader is referred to Ref. [18]. At the (k+1)th iteration, set $\boldsymbol{\alpha}_k = \boldsymbol{\nu}_{k+1} - \boldsymbol{\nu}_k$ as the update of the control variable $\boldsymbol{\nu}$, $\boldsymbol{\beta}_k = \nabla_{\boldsymbol{\nu}} L_{k+1} - \nabla_{\boldsymbol{\nu}} L_k$ the update of the gradient and $\mathbf{H}_k{}^{-1}$ the approximation of the inverse of the Hessian. The inverse of the approximate Hessian $\mathbf{H}_k{}^{-1}$ can be obtained using the BFGS update formula

$$\mathbf{H}_k^{-1} = \mathbf{V}_k \mathbf{H}_k^{-1} \mathbf{V}_k^{\mathrm{T}} + \frac{\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^{\mathrm{T}}}{\boldsymbol{\beta}_k^{\mathrm{T}} \boldsymbol{\alpha}_k}, \quad \text{with } \mathbf{V}_k = \mathbf{I} - (\boldsymbol{\beta}_k \boldsymbol{\alpha}_k^{\mathrm{T}})/(\boldsymbol{\beta}_k^{\mathrm{T}} \boldsymbol{\alpha}_k) \tag{51}$$

The following pseudo-code describes the BFGS two-loop iterative procedure used to compute the search $\mathbf{H}_k{}^{-1} \nabla_{\boldsymbol{\nu}} L_k$ efficiently by using the last m pairs of ($\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}_k$)

$\mathbf{s} \leftarrow \nabla_{\boldsymbol{\nu}} L_k$
for $i = k-1, k-2, \ldots, k-m$
$\gamma_i \leftarrow \boldsymbol{\alpha}_i^{\mathrm{T}} \mathbf{s} / \boldsymbol{\beta}_i^{\mathrm{T}} \boldsymbol{\alpha}_i$;
$\mathbf{s} \leftarrow \mathbf{s} - \gamma_i \boldsymbol{\beta}_i$;
End (for)
$\mathbf{r} \leftarrow (\mathbf{H}_k^0)^{-1} \mathbf{s}$;
For $i = k-m, k-m, \ldots, k-1$
$\boldsymbol{\delta}_i \leftarrow \boldsymbol{\beta}_i^{\mathrm{T}} \mathbf{r} / \boldsymbol{\beta}_i^{\mathrm{T}} \boldsymbol{\alpha}_i$;
$\mathbf{r} \leftarrow \mathbf{r} + (\gamma_i - \boldsymbol{\delta}_i) \boldsymbol{\alpha}_i$;
End (for)
Stop with result $\mathbf{H}_k^{-1} \nabla_v L_k = \mathbf{r}$ (52)

where $(\mathbf{H}_k{}^0)^{-1}$ is the initial approximation of the inverse of the Hessian matrix. One can set it as: $(\mathbf{H}_k{}^0)^{-1} = \varsigma_k \mathbf{I}$, with $\mathbf{I}$ is the identity matrix of appropriate dimension, and $\varsigma_k = (\boldsymbol{\alpha}_{k-1}^{\mathrm{T}} \boldsymbol{\beta}_{k-1}) / (\boldsymbol{\beta}_{k-1}^{\mathrm{T}} \boldsymbol{\beta}_{k-1})$.

4.4.3. Overall solution procedure

A systematic procedure flowchart for the AL implementation appears in the appendix (Fig. A1). In this procedure, after selecting robot parameters, task definition, (such as starting, intermediate and final poses), workspace limitations and simulation parameters, the kinematic unit defines a feasible solution satisfying initial and final poses. Then the inner optimization loop solves for the AL minimization with respect to sampling periods and actuator torques control variables to give the robot dynamic state. This state is then tested within a singularity test unit. If singular, the state is recalculated by going back to the inner optimization loop. If non-singular, a feasibility test is performed. The feasibility is done by testing the norms of all equality and inequality constraints against given tolerances. If the feasibility test fails, restart inner optimization unit. Otherwise, if the feasibility test succeeds, i.e., the current values of penalty are good in maintaining feasibility of iterates, a convergence test is made against optimal tolerances. If convergence holds, display optimal results and end the program. If non convergence, go further to the dual part of AL to test for constraints satisfaction and update multipliers, penalty and tolerance parameters. If the constraints are satisfied with respect to a first tolerance level (judged as good, though not optimal), then the multipliers are updated without decreasing penalty. If the constraints are violated with respect to a second tolerance level, then one keeps unchanged multiplier values and decreases penalty to ensure that the next sub-problem will place more emphasis on reducing the constraints violations. In both cases the tolerances are decreased to force the subsequent primal iterates to be increasingly accurate solutions of the primal problem.

5. Simulation case study

5.1. Description of the 2-DOF parallel manipulator case study

A simulation program has been implemented using Matlab [38] to test the proposed multi-objective trajectory planning approach on a PKM case study reported in Ref. [39]. Preliminary results are encouraging. This PKM consists of 2-DOF planar parallel manipulator. The robot kinematic and dynamic models considered have been developed in Ref. [37]. A schematic of the manipulator is depicted in Fig. 4, where the base is labeled 1 and the EE is labeled 2. The EE is connected to the base by two identical legs. Each leg consists of a planar four-bar parallelogram: links 2–5 for the first leg and links 2, 6–8 for the second leg. The link 3 and 8 are actuated by prismatic actuators, respectively. Motions of the EE are achieved by combination of movements of links 3 and 8 that can be transmitted to the EE by the system of the two parallelograms. Due to its structure, the manipulator can position a rigid body in a two-dimensional (2D) space with a constant orientation.

5.2. Kinematic and dynamic analysis

As illustrated in Fig. 4, a reference frame A:(O', x', y') is attached to the EE, and a reference frame B:(O, x, y) is attached to the robot

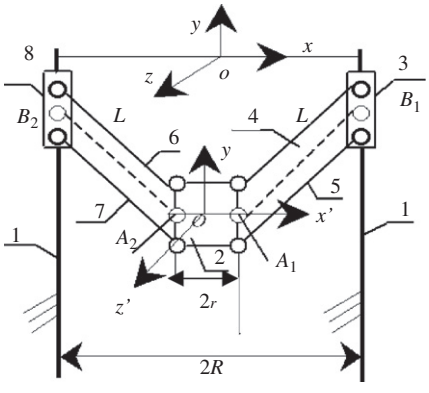

Fig. 4. A schematic representation of the planar parallel manipulator.

base, where O' is the origin of frame A and O the origin of B. To characterize the planar four-bar parallelogram, the chains A_1B_1 and A_2B_2 are considered, as shown in Fig. 4. Vectors A_{iA} and A_{iB} $(i = 1, 2)$ define the positions of points A_i in frames A and B, respectively. Vectors b_{iB} $(i = 1, 2)$ define the position of B_i points in frame B. The geometric parameters of the manipulator are

$$A_iB_i = L(i = 1, 2), \quad A_1A_2 = 2r \text{ and } B_1B_2 = 2R \tag{53}$$

The position of point O' in the fixed frame B is defined by the vector $(x, y)^T$. The kinematic equations of this manipulator are given by

$$\boldsymbol{J}_l \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \boldsymbol{J}_x \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{54}$$

Table 1
Limits of workspace, actuator torques and sampling periods

Parameter	x-coordinate (m)	y-coordinate (m)	τ_1 (N)	τ_2 (N)	h (s)
Maximum	0.8	−0.720	550	700	0.7
Minimum	−0.8	−1.720	−550	−700	0.005

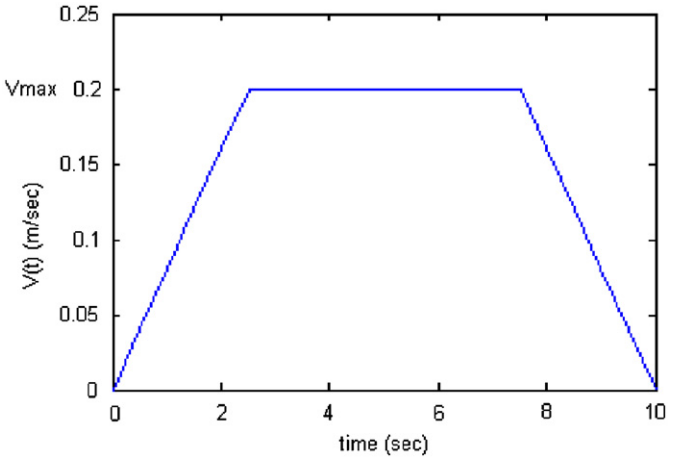

Fig. 5. Initial solution, velocity profile.

where $\boldsymbol{J}_l$ and $\boldsymbol{J}_x$ are, respectively, the 2×2 inverse and forward Jacobian matrices of the manipulator, which can be expressed as

$$\boldsymbol{J}_l = \begin{bmatrix} y - y_1 & 0 \\ 0 & y - y_2 \end{bmatrix}, \quad \boldsymbol{J}_x = \begin{bmatrix} r + x + R & y - y_1 \\ x - r + R & y - y_2 \end{bmatrix} \tag{55}$$

If $\boldsymbol{J}_l$ is non-singular, the Jacobian matrix of the manipulator can be obtained as

$$\boldsymbol{J} = \boldsymbol{J}_l^{-1}\boldsymbol{J}_x = \begin{bmatrix} (r + x - R)/(y - y_1) & 1 \\ (x - r + R)/(y - y_2) & 1 \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \tag{56}$$

Accordingly, it is clear that singularity occurs when one of the following cases holds

1st case: $|\boldsymbol{J}_l| = 0$ and $|\boldsymbol{J}_x| \neq 0$. This case is known as the first type singularity, and corresponds to the situation where $y = y_1$ or $y = y_2$, i.e., the first or the second leg is parallel to the x-axis.

2nd case: $|\boldsymbol{J}_l| \neq 0$ and $|\boldsymbol{J}_x| = 0$. This case is known as the second type singularity. It corresponds to the pose where four bars of the parallelogram in one of the two legs are parallel to each other. It is analytically expressed by the equality $x+r = R$ for the first leg when x is positive and $x+R = r$ for the second leg when x is negative.

3rd case: $|\boldsymbol{J}_l| = 0$ and $|\boldsymbol{J}_x| = 0$. This corresponds to the third type of singularity for which the two legs are both parallel to the x-axis. This is mainly a design issue as it is characterized by a geometric parameters condition given by: $L+r = R$.

The robot dynamic model in the task space of the PKM is obtained through Lagrange formalism as follows:

$$\begin{aligned}
\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} &= \begin{bmatrix} \left(m_p + \frac{4}{3}m_l\right) + 2\left(m_s + \frac{m_l}{3}\right)(J_{11}^2 + J_{21}^2) & \left(m_s + \frac{2}{3}m_l\right)(J_{11} + J_{21}) \\ \left(m_s + \frac{2}{3}m_l\right)(J_{11} + J_{21}) & m_p + 2m_s + \frac{8}{3}m_l \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \\
&+ \left[L^2 \left[\frac{\dot{x}^2}{\sqrt{\dot{x}^2 + \dot{y}^2}} \left(m_s + \frac{2}{3}m_l\right) - \sqrt{\dot{x}^2 + \dot{y}^2} \right] \left[\frac{J_{11}\sqrt{J_{11}}}{(x + r - R)^{3/2}} + \frac{J_{21}\sqrt{J_{21}}}{(x - r + R)^{3/2}} \right] \right. \\
&\left. L^2 \left[\dot{x}^2 \left(m_s + \frac{2}{3}m_l\right) + \frac{\dot{x}\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \right] \left[\frac{J_{11}\sqrt{J_{11}}}{(x + r - R)^{3/2}} + \frac{J_{21}\sqrt{J_{21}}}{(x - r + R)^{3/2}} \right] \right] \\
&+ \left[\left[\frac{\dot{y}(\ddot{x}\dot{y} - (\dot{x}\ddot{y})}{(\dot{x}^2 + \dot{y}^2)^{3/2}} + g\left(m_s + \frac{m_l}{2}\right) \right] (J_{11} + J_{21}) \right. \\
&\left. \left[\frac{\ddot{y}\dot{x}^2 - \ddot{x}\dot{y}\dot{x})}{(\dot{x}^2 + \dot{y}^2)^{3/2}} (J_{11} + J_{21}) + g(m_p + m_l) \right] \right]
\end{aligned} \tag{57}$$

More details on the derivation of the dynamic model might be found in Ref. [37].

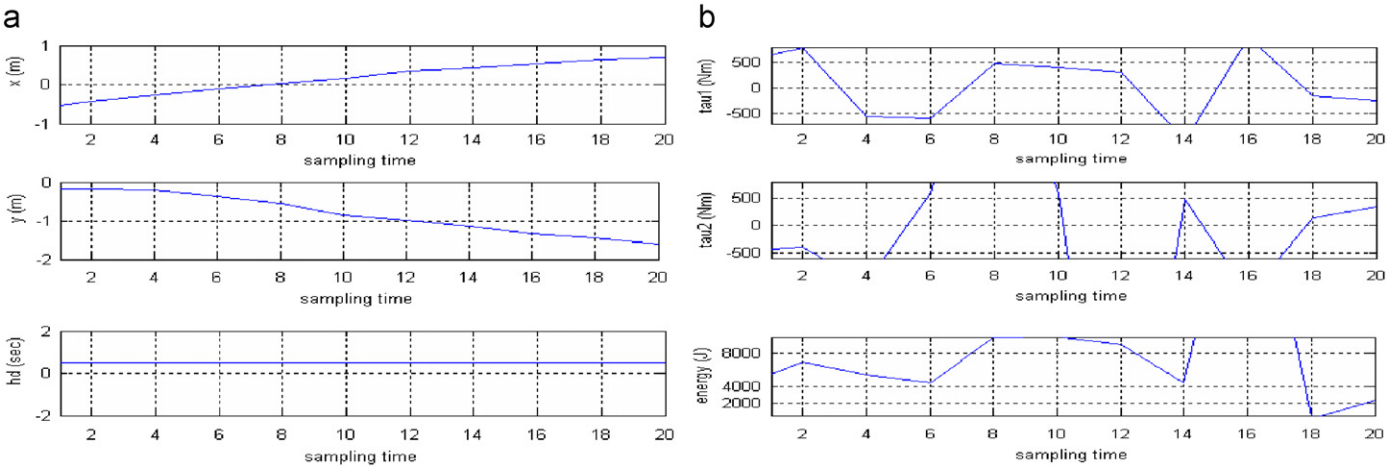

Fig. 6. Kinematic simulation results: (a) variations of x, y, coordinates of the EE, and sampling periods, (b) variations of torques τ_1, τ_2, and energy.

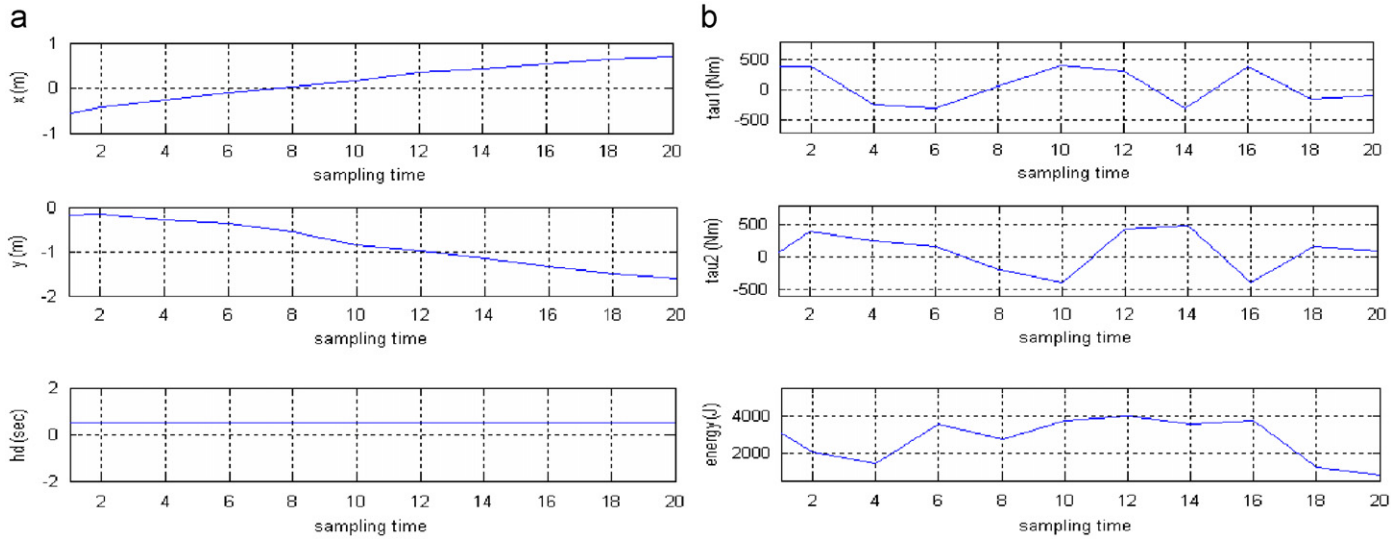

Fig. 7. Simulation results with the AL (minimum time–energy): (a) variations of x, y, coordinates of the EE, and sampling periods, (b) variations of torques τ_1, τ_2, and energy.

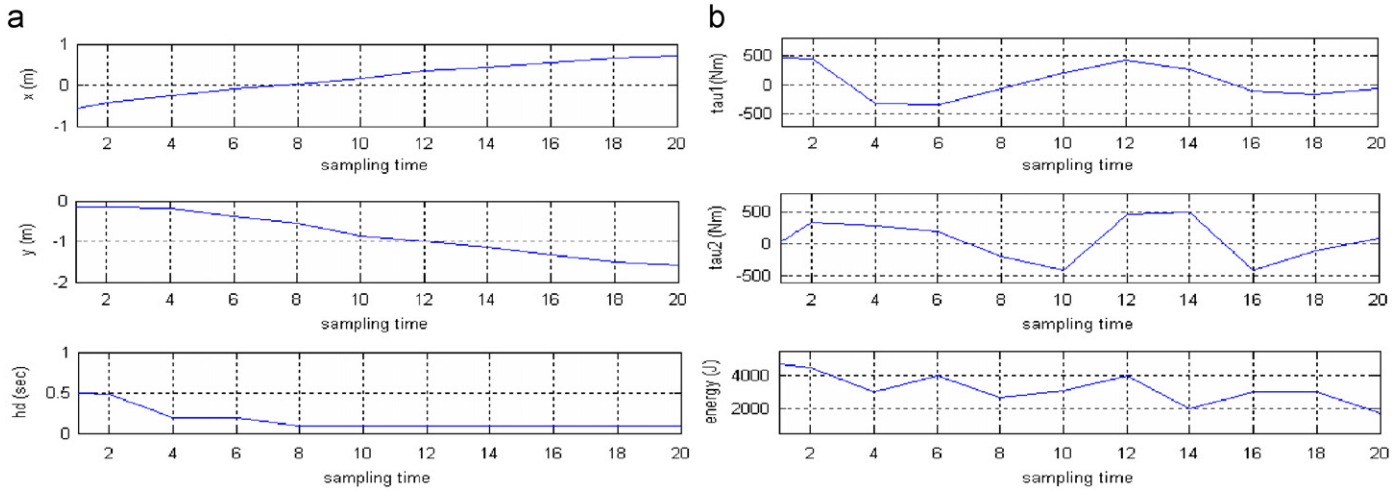

Fig. 8. AL (minimum energy): (a) variations of x, y, coordinates of the EE, and sampling periods, (b) variations of torques τ_1, τ_2, and energy.

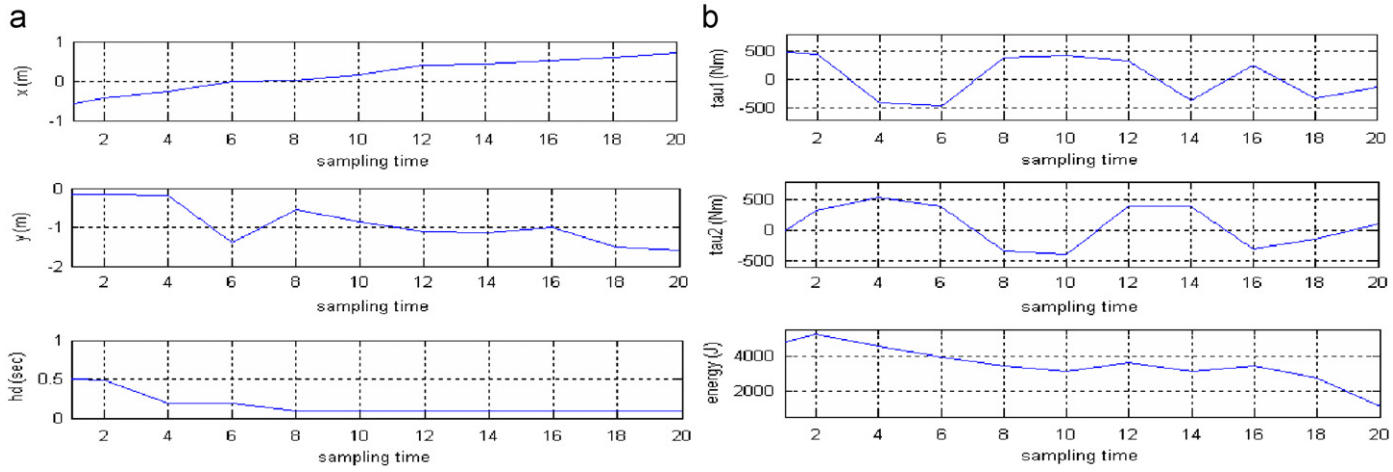

Fig. 9. AL minimum time–energy with imposed passage through Cartesian positions: (a) variations of x, y, coordinates of the EE, and sampling periods, (b) variations of torques τ_1, τ_2, and energy.

Following the streamline developed in previous sections, a discrete-time state-space model associated to the state Eq. (57) is:

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} \boldsymbol{I}_{2\times2} & h_k\boldsymbol{I}_{2\times2} \\ \boldsymbol{O}_{2\times2} & \boldsymbol{I}_{2\times2} \end{bmatrix}\boldsymbol{x}_k - \begin{bmatrix} \frac{h_k^2}{2}\boldsymbol{I}_{2\times2} \\ h_k\boldsymbol{I}_{2\times2} \end{bmatrix}[\bar{\boldsymbol{M}}_c^{-1}(\boldsymbol{x}_{1k})[\bar{\boldsymbol{N}}_c(\boldsymbol{x}_{1k},\boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k})]] + \begin{bmatrix} \frac{h_k^2}{2}\boldsymbol{I}_{2\times2} \\ h_k\boldsymbol{I}_{2\times2} \end{bmatrix}\bar{\boldsymbol{M}}_c^{-1}(\boldsymbol{x}_{1k})\boldsymbol{\tau}_k \quad (58)$$

The optimal control problem consists to minimize criterion Eq. (35) subject to dynamic Eq. (58) and equality and inequality constraints Eqs. (29)–(31), and the following specific constraints.

- Workspace limitations:

$$x_{\text{Min}} \leqslant x_k \leqslant x_{\text{Max}}, \quad y_{\text{Min}} \leqslant y_k \leqslant y_{\text{Max}} \quad \text{for } k = 1,2,\ldots,N \quad (59)$$

- Singularity avoidance:
  In our considered case study, the first type singularity constraint might be expressed by:

$$|(y_k - y_{1k}()y_k - y_{2k})| \geqslant \varepsilon_1 \quad (60)$$

  whereas the second it is given as

$$|(x_k + \text{sgn}(x_k)(r - R))| \geqslant \varepsilon_2 \quad (61)$$

  ε_1 and ε_2 represent small positive tolerances.

Table 2
Convergence history of minimum time–energy planning with AL

Ndisc	Nprimal	Ndual	CPU (s)	t_T (s)	Energy (J)	AP	
						EqPre	IneqPre
10	4	7	12.62	9.50	6874.37	3.10^{-3}	3.10^{-3}
20	4	7	29.25	8.01	5156.28	10^{-3}	10^{-3}
20	5	10	35.54	7.31	4910.39	5.10^{-4}	4.10^{-4}
30	5	10	69.83	6.07	4210.23	2.10^{-5}	3.10^{-5}

The third singularity type concerns mainly the geometric parameters L, r and R. These parameters are chosen at the design level, such that the equality $L+r = R$ does not hold. The required passage poses is reduced to positioning ones, insofar a constant orientation is assumed during task execution. Typically, one might have

$$s(\boldsymbol{x}) = ||\mathbf{p} - \mathbf{p}_l|| - \mathbf{T}_{\text{PassThl}_p} = 0 \quad (62)$$

The AL and the associated decoupled formulation are obtained along with various gradients. These calculations are quite long. The reader is referred to Ref. [37] for further details.

5.3. Simulation data and scenario

The following numeric values are used: the EE mass is $m_{\text{EE}} = 200.0\,\text{kg}$ that of each leg is $m_l = 570.5\,\text{kg}$ and that of the slider is $m_s = 100\,\text{kg}$. The platform radius is $r = 0.75\,\text{m}$, $R = 1.2030\,\text{m}$ and the strut length $L = 1.9725\,\text{m}$. Table 1 shows the limits of the workspace, actuator torques and sampling periods. For the AL, the following parameter values had been taken $\omega_s = 0.5$, $\eta_s = 0.5$, $\alpha_w = \alpha_\eta = 0.4$, $\beta_w = \beta_\eta = 0.4$, $w_0 = \eta_0 = \eta_{10} = 10^{-2}$, $w^* = \eta^* = \eta^*_1 = 10^{-5}$, $\gamma_1 = 0.25$, $\gamma_2 = 1.2$, $\nu = 0.01$, $\bar{\nu} = 0.3$. The initial Lagrange multipliers $\boldsymbol{\rho}_0$, $\boldsymbol{\sigma}_0$ components are set to zero. The singularity weight is $\delta = 1$. The maximum value for $\boldsymbol{\delta}_{\text{Max}} = 10^{42}$, and the minimum value for $\boldsymbol{\delta}_{\text{Max}} = 10^{-42}$. The scenario consists of a straight line trajectory from an initial Cartesian state position $x_0 = -0.7$, $y_0 = -0.1$ to a final position $x_T = 0.7$, $y_T = -1.6$ (in meters). The initial and final linear and angular velocities are equal to zero.

The maximum velocity is 0.2 m/s and maximum acceleration is $2\,\text{m/s}^2$. The maximum allocated time for this trajectory is 10 s. In the presented simulations, the focus is on time–energy constrained trajectory planning by the AL, more kinematic-related performance evaluation and design for a similar case study might be found in Refs. [37,39]. Typically, four simulation objectives are considered: (1)–Compare robot trajectories for different values of the weights $\boldsymbol{U}$, $\boldsymbol{Q}$, $\boldsymbol{\iota}_1$ and $\boldsymbol{\delta}$. (2)–Assess the effects of the dynamic parameters changes on the AL sensitivity and on the behavior of the PKM. (3)–At which precision vs. time consumption, the augmented Lagrangian achieves passage

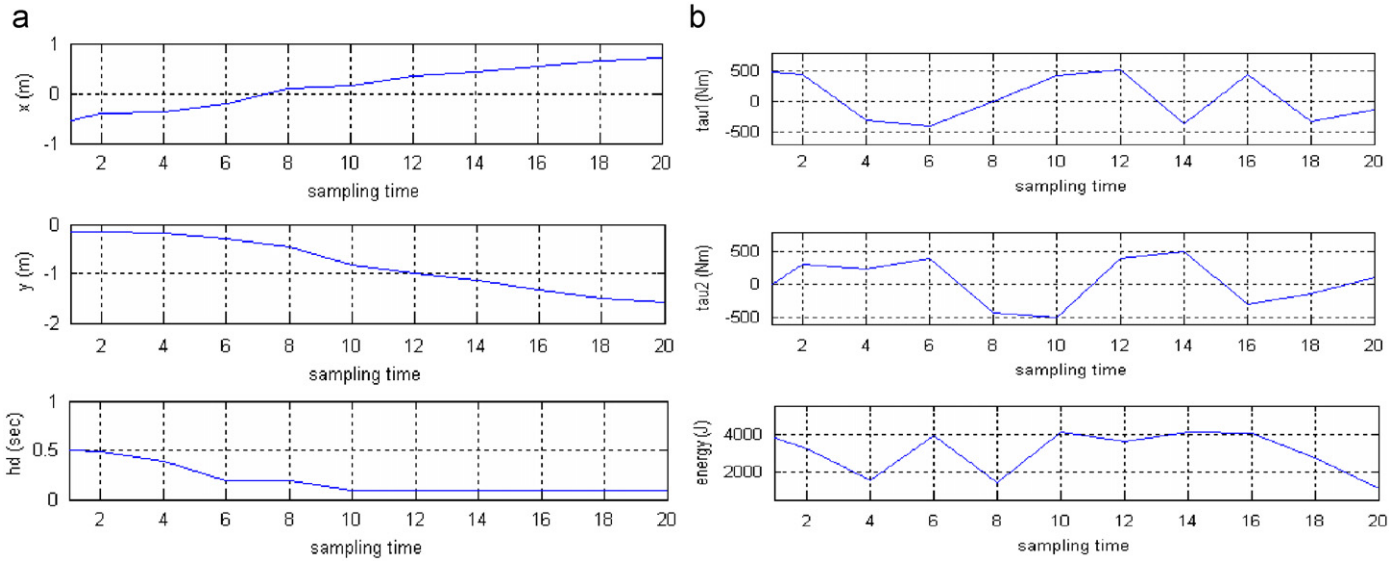

Fig. 10. Disturbed AL (minimum tme–energy) (modified mass of EE $m_{\text{EE}} = 300.0\,\text{kg}$): (a) variations of x, y, coordinates of the EE, and sampling periods, (b) variations of torques τ_1, τ_2, and energy.

satisfaction through imposed poses? (4)–To what extent the number of inner and outer iterations of AL impacts PKM performance vs. CPU Time?

To start, Fig. 5 shows the velocity profile used to initialize the AL. Figs. 6 and 7 show the simulation results for both initial kinematic and AL solutions. In part (a) of these figures, the first plot from the top shows the displacement along the x-axis of the EE point of operation. The second plot shows the displacement along the y-axis of the EE point of operation. The third shows the instantaneous values of the consumed time to achieve the trajectory. In part (b), the first and second plots from the top show the instantaneous variations of joint torques, while the third one shows the instantaneous values of the consumed energy. It is noteworthy that although the initial solution is kinematically feasible, when the corresponding torques is computed considering the dynamic model and forces, one gets shortly torque values outside the admissible domain resulting in high values for the energy cost. With the AL, however, with four inner and seven outer iterations, the variations of the energy consumption are increasing smoothly and monotonously. Fig. 8 displays the simulation outcomes for the only energy criterion (i.e., the time weight is set to zero, so the sampling period is kept constant). One gets a 34% faster trajectory with time–energy criterion (Fig. 7), as compared to a trajectory computed with only minimum-energy criterion (Fig. 8). As for imposed passages through pre-specified poses, the same scenario as above is simulated, while constraining the EE to pass through the following positions: (0.0, −1.4), (0.4, −1.1) and (0.5, −1.0), all in meters. Fig. 9 shows the trajectory corresponding to passage through imposed poses. With 10 dual iterations and 7 primal ones, one gets a precision of 7.10^{-4}, which confirms the well-known constraints satisfaction performance of AL for constrained optimization problems, as compared to its counterparts like penalty methods. Furthermore, we observe that the proposed variational approach has not only been successful in finding singularity-free trajectory, but also the obtained trajectories are optimal in time and energy minimization. To analyze with respect to AL parameters, Table 2 shows comparison of results for different simulation parameters of AL, where NDisc is the number of discretisations, NPrimal is the number of inner optimization loops, NDual is the number of outer optimization loops, $t_T = \sum_{k=1}^{N} h_k$ is the total traveling time, Energy $= \sum_{k=1}^{N} [\boldsymbol{\tau}_k \boldsymbol{U} \boldsymbol{\tau}_k^T + \boldsymbol{x}_{2k} \boldsymbol{Q} \boldsymbol{x}_{2k}^T)]$ is the consumed electric and kinetic energy, and AP for achieved precision for, respectively, equality (EqPre) and inequality (IneqPre) constraints satisfaction. The values shown for the total traveling time t_T, Energy, and AP correspond to those computed for the last outer iteration.

5.4. Sensitivity analysis

The optimal time–energy control considered so far is dependent on the values of the dynamic parameters of the PKM. As PKMs are strongly nonlinear and coupled mechanical systems, several of these parameters such as inertial parameters are known only approximately or may change. So, a sensitivity analysis [40] is necessary to know how robust the proposed approach to the parameter changes is.

This is performed through varying the value of the EE mass. On Fig. 10, it is shown that the AL simulation with modified EE mass as $m_{EE} = 300.0$ kg. One notices that the needed actuator torques and necessary energy and time to achieve the same task are higher, especially at the beginning.

6. Conclusions and discussions

The basic contribution of this paper is the formulation and resolution of the trajectory planning problem of PKMs using a variational calculus framework. This is performed by considering robot kinematic and dynamic models, while optimizing time and energy necessary to achieve the trajectory, avoiding singularities and satisfying several constraints related to the robot, task and workspace. The robot dynamic model includes the EE, struts and actuators models. The AL algorithm is used to solve the resulting nonlinear and nonconvex optimal control problem. This optimization technique is used along with a decoupled and linearized formulation of the original problem, permitting the ultimate benefit of easing the computation of the co-states and other variables necessary to perform optimization. Although, it is task and algorithm parameter settings dependent, the computational time is drastically reduced when the decoupled and linearized formulation is used. It has been shown that the proposed approach performs better in optimizing traveling time and actuator torques than kinematic only based schemes. Furthermore, the proposed trajectory planning is robust to dynamic parameters changes. This in fact is due to the ability of the AL to cope with numerical ill-conditioning problems, as compared to other optimization techniques like penalty methods. Moreover, a major advantage of this approach is that one can introduce any type of constraints related to the robot, task or environment, like obstacles or link interference avoidance, by deriving the corresponding constraint expressions and adding them naturally in the Lagrangian in order to have them included in the trajectory planning system. These issues are now being incorporated in an ongoing work.

Acknowledgment

The authors gratefully thank the Natural Science and Engineering Research Council of Canada (NSERC) for supporting this work under Grants ES D3-317622, RGPIN-203618, RGPIN-105518 and STPGP-269579. The first author thanks also King Fahd University of Petroleum and Minerals for general support.

Appendix A. Augmented Lagrangian algorithm

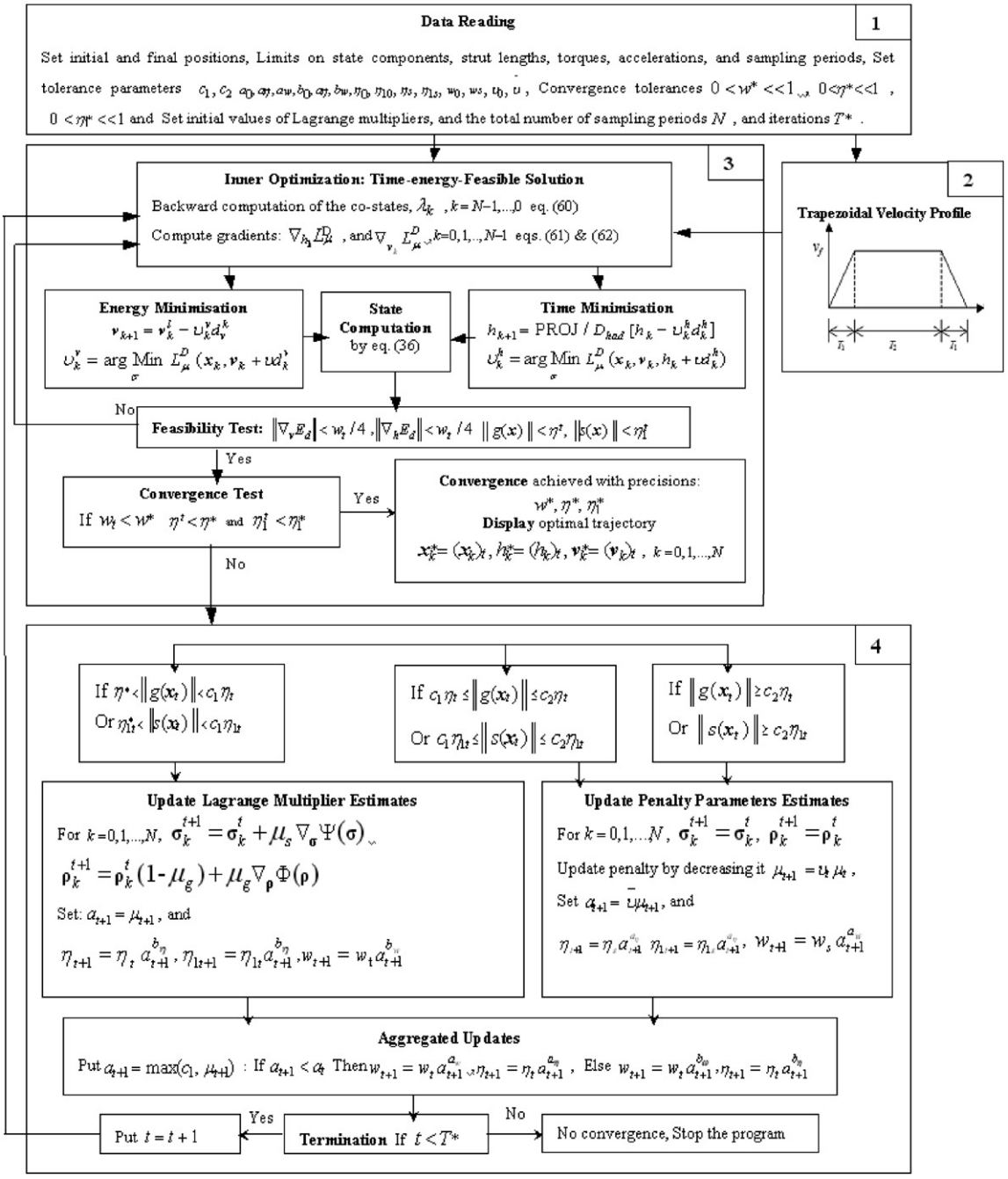

Fig. A1. Flowchart for AL algorithm function and operation.

References

[1] Gough VE. Contribution to discussion of papers on research in automobile stability, control and type performance. In: Proc Auto Div Inst Mech Eng Part D (J Automob Eng). 1956. p. 392–395.

[2] Stewart D. A platform with six degree-of-freedom. In: Proc Inst Mech Eng. vol. 180. 1965. p. 371–386.

[3] Merlet JP. Parallel robots. Dordrecht, The Netherlands: Kluwer Academic Publisher; 2000.

[4] Pietsch IT, Krefft M, Becker OT, Bier CC, Hesselbach J. How to reach the dynamic limits of parallel robots? An autonomous control approach. IEEE Trans Autom Sci Eng 2005;2(4):369–80.

[5] Ahrikencheikh C, Seireg A. Optimized-motion planning. New York: Wiley; 1994.

[6] Khoukhi A, Ghoul A. Maximum dynamic stress search for a robot manipulator. Robotica 2004;22(5):513–22.

[7] Khoukhi A. An optimal time–energy control design for a prototype educational robot. Robotica 2002;20(6):661–71.

[8] Vukobratovic M, Stotic D. Is dynamic control needed in robotics system? And if so to what extent? Int J Robotics Res 1985;2(2):18–35.

[9] Dasgupta B, Mruthyunjaya TS. The stewart platform manipulator, a review. Mech Mach Theory 2000;35(1):15–40.

[10] Baron L, Angeles J. The kinematic decoupling of parallel manipulators under joints-sensor redundancy. IEEE Trans Robotics Autom 2000;16(1):12–9.

[11] Baron L, Angeles J. The direct kinematics of parallel manipulators under joints-sensor data. IEEE Trans Robotics Autom 2000;16(6):644–51.

[12] Ma O, Angeles J. Architecture singularities of parallel manipulators. Int J Robotics Autom 1992;7(1):23–9.

[13] Gosselin C. Parallel computational algorithms for the kinematics and dynamics of plannar and spatial parallel manipulators. ASME, J Dyn Syst, Meas Control 1996;118(1):22–8.
[14] Kumar DA, Ming CI, Huat YS, Guilin Y. Workspace generation and planning singularity-free path for parallel manipulators. Mech Mach Theory 2005; 40(7):776–805.
[15] Mermertas V. Optimal design of manipulator with four-bar mechanism. Mech Mach Theory 2004;39(5):545–54.
[16] Bhattacharya S, Hatwal H, Ghosh A. Comparison of an exact and approximate method of singularity avoidance in platform-type parallel manipulators. Mech Mach Theory 1998;33(7):965–74.
[17] Sen S, Dasgupta B, Mallik AK. A variational approach for singularity-free path-planning of parallel manipulators. Mech Mach Theory 2003;38(11):1165–83.
[18] Polak E. Optimization, algorithms and consistent approximation. New York: Springer; 1997.
[19] Hay M, Snyman JA. Methodologies for the optimal design of parallel manipulators. Inl J Numer Methods Eng 2004;59(1):131–52.
[20] Hay M, Snyman JA. A multi-level optimization methodology for determining the dexterous workspaces of planar parallel manipulator. Struct Multidisciplinary Optim 2005;30(6):422–7.
[21] Khoukhi A, Baron L, Balazinski M. Programmation dynamique time-énergie minimum des robots parallèles par lagrangien augmenté. CCToMM Symposium on Mechanics, Machines and Mechatronics. Canadian Space Agency: Saint Hubert, Canada; May 26–27, 2005. p. 147–8.
[22] Khoukhi A, Baron L, Balazinski M. A decoupled approach to time-energy trajectory planning of parallel kinematic machines. In: Proc sixteenth CISM-IFToMM Robotics and Manipulators Symposium Warsaw. Poland: June 20–24, 2005. p. 179–86.
[23] Khoukhi A. Dynamic modelling and optimal time-energy off-line programming for mobile robots, a cybernetic problem. Kybernetes 2002;31(5–6): 731–5.
[24] Rockafellar T. Lagrange multipliers and optimality. SIAM Review 1993;35(2): 183–238.
[25] Bertsekas DP. Non linear programming. Athena Scientific; 1995.
[26] Hiriart-Urruty JB, Lemaréchal C. Convex analysis and minimization algorithms. Berlin, New York: Springer; 1993.
[27] Nakamura Y. Advanced robotics: redundancy and optimization. Reading MA: Addison-Wesley; 1991.
[28] Angeles J. Fundamentals of robotic mechanical systems, theory, methods and algorithms. 2nd ed. Berlin: Springer; 2003.
[29] Harib K, Srinivasan K. Kinematic and dynamic analysis of stewart platform-based machine tool structures. Robotica 2003;21(5):541–51.
[30] Jurgen H, John Mc. Determination of minimum-time maneuvers for a robotic manipulator using numerical optimization methods. Mech Struct Mach 1999; 27(2):185–201.
[31] Yu H, Wang J, Duan G, Sun L. Approach for singularity avoidance path planning of parallel robots. In: Proc eleventh world congress in mechanism and machine science. 2004. p. 1961–1965.
[32] Nenchev DN, Bhattacharya S, Uchiyama et M. Dynamic analysis of parallel manipulators under singularity-consistent parameterization. Robotica 1997; 15(4):375–84.
[33] Innocenti C, Parenti-Castelli V. Singularity-free evolution from one pose to another in serial and fully parallel manipulators. ASME J Mech Des 1998; 120(1):73–9.
[34] Powell MJD. A method for nonlinear constraints in minimization problems. In: Fletcher R, editor. Optimization. London: Academic Press; 1969. p. 283–98.
[35] Hestens MR. Multiplier and gradient methods. J Opt Theory Appl 1969;4: 303–20.
[36] Isidori A. Nonlinear control systems. 3rd ed. London, UK: Springer; 1995.
[37] Khoukhi A, Baron L, Balazinski M. Constrained multi-objective off-line programming of parallel kinematic machines. Technical Report, École Polytechnique de Montréal, CDT-P2877-05, 70 Pages; May 2007.
[38] Available: 〈http://www.mathworks.com/products/matlab/〉.
[39] Liu XJ, Wang QM, Wang J. Kinematics, dynamics and dimensional synthesis of a novel 2-DOF translational manipulator. J Intell Robotic Syst 2004;41: 205–24.
[40] Saltelli Andrea, Chan K, Scott EM, editors. Sensitivity analysis. New York: Wiley; 2000.

Amar Khoukhi received his Ph.D. degree in Mechanical Engineering from Mechanical Engineering Dept, University of Montreal in 2007, an Engineering Doctorate from Software Engineering Dept. of National School of Telecommunication Engineers, Paris, France, in 1991 and an Advanced Diploma of University Studies in Operation Research from Decision Mathematics Institute of University Paris IX Dauphine in 1985. Over the last several years, he had been actively involved in several research projects in the fields of robotics, soft computing and applied optimization and funded by NSERC Canada, CNRS France and Electricity of France. Since February 2008, he joined Systems Engineering Dept. King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. He received the Best Paper Award at NAFIPS'2007 Conference held at San Diego, CA, June 25–27, 2007 and has published more than 40 international peer-reviewed journal papers and over 90 conference papers. His research interests include robotics, applied optimization and intelligent systems.

Luc Baron received his B.Eng. and M.A.Sc. degrees from Ecole Polytechnique, Montreal, Canada, in 1983 and 1985, respectively, and Ph.D. degree from McGill University, Montreal, Canada, in 1997, all in Mechanical Engineering. From 1986 to 1993, he held cross appointments as faculty lecturer at the Ecole Polytechnique, the engineering faculty of University of Montreal, and as Research Engineer at the Walsh Automation Inc. Since 1997, he is Professor in the Department of Mechanical Engineering at Ecole Polytechnique. His current research interests are in the field of robotics and industrial automation, kinematics, synthesis and control of manipulators. He is a Member of IEEE, an Associate Member of ASME and a registered engineer in Quebec.

Marek Balazinski is a specialist in the area of manufacturing, fuzzy logic and genetic algorithms. He has obtained his bachelor, master and doctoral degrees from the Technical University of Carcow in Poland. Over the last several years, he has focused on metal cutting of hard and difficult to machine materials and on application of fuzzy logic and genetic algorithms in manufacturing. He is currently Professor and Head of the Manufacturing Section in the Department of Mechanical Engineering at the École Polytechnique of Montréal. Over the last 10 years, he has published over 100 research papers and he is an active Member of CIRP (The International Academy for Production Engineering).