Constrained Multi-Objective Trajectory Planning of Parallel Kinematic Machines

Amar Khoukhi, Luc Baron and Marek Balazinski

Mechanical Engineering Department, École Polytechnique de Montréal, C. P. 6079, Succ. CV, Montreal QC, Canada H3C 3A7
(amar.khoukhi, luc.baron, marek.balazinski)@polymtl.ca

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 models, (including actuators) are first developed. Then the proposed trajectory planning system is introduced. It minimizes electrical and kinetic energy, robot travel time separating two sampling periods and maximizes a measure of manipulability allowing singularity avoidance. Several technological constraints such as actuator limits, link length and workspace bounds, and certain task requirements, such as passing through imposed poses are simultaneously satisfied. The discrete augmented Lagrangian technique is used to solve the resulting non-linear and non-convex 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 on a 2-degrees of freedom PKM are given and discussed.

Keywords: Parallel Kinematic Machines, Constrained Off-line Programming, Nonlinear Optimal Control, Time-Energy Trajectory Planning, Augmented Lagrangian, Decoupling.
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)

\(A_i, B_i\) : \(i^{th}\) attachment point of leg \(i\) on body \(A\) and \(B\)

\(e_i\) : Unit vector along the \(i^{th}\) joint axis

\(p = [x\ y\ z]^T\) : Position vector of the origin of frame \(A\) relative to frame \(B\) in \(B\)

\(a_i\) : Constant position vector of attachment point \(A_i\) in frame \(A\).

\(\dot{p} = [\dot{x}\ \dot{y}\ \dot{z}]^T\) : Velocity vector of the origin of frame \(A\) relative to frame \(B\)

\(aR_{A} \) : Rotation matrix of frame \(A\) with respect to frame \(B\)

\(x_i = q = [p^T\ \phi\ \theta\ \psi]^T\) : Position and orientation of \(A\) in \(B\)

\(\dot{q}_e = [\dot{p}^T\ \dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T\) : Time derivatives of \(x_i(t)\)

\(x = [x_i x_2]^T\) : Cartesian and angular velocity of the EE

\(x = [x_1 x_2]^T\) : Continuous-time state of the PKM

\(x_k = [x_{1k} x_{2k}]^T\) : Discrete-time state of the PKM

\(\tau = [\tau_1 \tau_2 \ldots \tau_6]^T\) : Cartesian force/torques wrench

\(l = [l_1 l_2 \ldots l_6]^T\) : Vector of the link lengths

\(J\) : Jacobian matrix of the PKM

\(J_x\) : Forward Jacobian matrix of the PKM

\(J_j\) : Inverse Jacobian matrix of the PKM

\(M_j(q), M_c(q)\) : Inertia matrix expressed in joint and Cartesian space

\(N_j(q, \dot{q}), N_c(q, \dot{q})\) : Coriolis and centrifugal force/torque in joint and Cartesian space

\(G_j(q), G_c(q)\) : Gravity force in joint and Cartesian space

\(M_a, M_a\) : Actuator inertia matrix and its component

\(V_a, V_a\) : Actuator viscous damping coefficient matrix and its component

\(K_a, K_a\) : Actuator gain matrix and its component
\( K \) : Control law gain matrix
\( \tau_m \) : Joint torque vector produced by the DC motors
\( F \) : 6-directional column of the generalized forces
\( p \) : 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
\( \lambda \) : Lagrange multipliers associated with state variables
\( f_{d_k} \) : Approximated function of the discretized robot dynamic model
\( f_{d_k}^D \) : Approximated function of the discretized robot decoupled dynamics
\( (p_l, R_l) \) : Imposed \( l^{th} \) position and orientation
\( \Theta_{\text{max}} \) : Maximum limit of the joint position
\( L_\mu \) : Augmented Lagrangian function
\( L_\mu^D \) : Augmented Lagrangian function of the decoupled problem
\( E_d \) : Discrete objective function
\( E_d^D \) : Discrete objective function of the decoupled problem
\( (\sigma, p) \) : Lagrange multipliers associated with equality and inequality constraints
\( (\Psi, \Phi) \) : Penalty functions associated with equality and inequality constraints
\( (\mu_s, \mu_g) \) : Penalty coefficients associated with equality and inequality constraints
\( (s, g) \) : Generic functions defining equality and inequality constraints
\( C \) : Admissible domain of input torques
\( H \) : Admissible domain of sampling periods
\( U, Q \) : Weight matrices associated with electric and kinetic energy criteria
\( (v, \delta) \) : Weight factors for traveling time and singularity avoidance criteria
\( w \) : Measure of manipulability function
\( \sigma \) : Singularity avoidance function
\( N \) : Total number of discretizations of the trajectory
\( T^* \): Total number of iterations of the optimization algorithm

\( w_t \): Feasible tolerance for objective function minimization

\( \eta_{i,t}, \eta_t \): Feasible tolerances for equality and inequality constraint satisfaction

\( w^* \): Optimal tolerance for objective function minimization

\( \eta^*_i, \eta^*_t \): Optimal tolerances for equality and inequality constraint satisfaction

\( T_{\text{PassTh}_p} \): Tolerance on passage satisfaction through the \( P^p \) position

\( T_{\text{PassTh}_b} \): Tolerance on passage satisfaction through the \( P^b \) orientation

1. **Introduction**

The design of Parallel Kinematic Machines (PKMs) dates back to 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 orient an end-effector (EE), such that to test tire wear and tear. A decade later, Stewart [2] proposed a platform manipulator for use as an aircraft simulator. Since then, extensive research efforts have led 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 PKM structures, 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 ½ for parallel ones. Despite these advantages, PKMs are still rare in industrial applications. 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 have led to a tremendous amount of research in PKMs design and customization [3]. Another reason recently identified is the under estimation of the dynamics of these machines [4]. The previously mentioned architecture-dependent performance associated with strong-coupled non-linear dynamics makes the trajectory planning and control system design for PKMs more difficult compared to serial machines. In fact, for serial robots, there is a plentiful literature published on the topics of offline 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 overwhelming criteria considered for PKMs trajectory planning are essentially design-oriented. These include singularity avoidance and dexterity optimization [9-13]. In reference [14], the authors had developed a clustering scheme to isolate and avoid singularities and obstacles for PKMs path planning. A kinematic design and planning method had been described in [15] for a four-bar planar manipulator mechanism. Another related work was considered in [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 [17] for planning a singularity-free minimum-energy path between two end-points for Gough-Stewart platforms. The approach used is based on penalty optimization methods. These methods transform the constrained problem in an unconstrained one by augmenting the objective function with additional functions that reflect the violation of the constraints. Penalty methods, however, have several 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 [19, 20] where design methodologies involving workspace and actuator force limitations were proposed, using iterative optimization techniques.

In this paper we consider a new integrated multi-objective dynamic trajectory planning system for PKMs. Part of this work has been presented in [21-22]. The proposed approach considers PKM dynamics including actuators models as well as task and workspace requirements as a unique entity. It can be encapsulated into two levels (see Figure 1): the modeling and approach level, and the simulation and testing level [7, 23]. The former consists of selecting the appropriate models and control approaches in order to optimize the overall performance of robot-task-workspace interactions. The second level is devoted to coding, testing, and validation. Criteria to be optimized in this study are travel time, energy necessary for a task execution and a measure of manipulability. 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, trajectory planning is formulated within a variational calculus framework. The resulting constrained non-linear programming problem is solved using an augmented Lagrangian (AL) technique implemented on a decoupled dynamics.
AL algorithms have proven to be robust and powerful enough to cope with difficulties related to constraints which are not strictly convex [24-27] as compared to optimization methods employing only penalties. The decoupling technique is introduced to solve some difficult computations in the original non-linear and coupled formulation. Another advantage of the proposed method is that it enables the introduction of several criteria and constraints to satisfy in the trajectory planning process.

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 non-linear optimal control problem is formulated. In section 4, the augmented Lagrangian 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-DOF planar PKM is provided. Finally, Section 6 concludes this work.

2. Modeling

2.1. Kinematic Model

The PKM shown in Figure 2 represents a full 6-Degrees-Of-Freedom (DOF) motion of its EE through articulated motion of its six leg lengths. The *pose*, i.e., position and orientation of the EE, namely, \( \mathbf{q} = [\mathbf{p} \quad \phi \quad \theta \quad \psi]^T \) can be expressed in frame \( B \) by \( \mathbf{p} \), the position vector of the origin of frame \( A \) relative to frame \( B \), and the set of Euler angles \((\phi, \theta, \psi)\) defining the orientation of the EE in frame \( B \). An example of the expression of this orientation matrix may be found in [28].

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

\[
\dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{3 \times 3} & \mathbf{R}(\phi, \theta, \psi) \end{bmatrix} \dot{\mathbf{q}}
\]

(1)

where the transformation between the angular velocity \( \mathbf{w} \) and the time derivatives of the Euler angles \( \phi, \theta, \psi \) is given as [28, 29]:
There are two basic problems in kinematic modeling; *forward kinematics* is the determination of EE motion from a given motion of the leg lengths, while *inverse kinematics* is the determination of leg length motion from a given EE motion.

### 2.1.1 Inverse Kinematics

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

$$a_i = p + {}^b R_i^A a_i, \quad i=1,...,6,$$

where $^A a_i$ is the constant position vector of $A_i$ in frame $A$. By differentiating Equation (3) with respect to time, projecting along the joint axis and grouping in a matrix form, one gets the inverse rate kinematic model as:

$$\dot{i} = J^{-1}(q) \dot{q}$$

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

$$J^{-1} = \begin{pmatrix} e_1^T \\ \vdots \\ e_6^T \\ {}^b R_a^A a_1 \times e_1 \\ \vdots \\ {}^b R_a^A a_6 \times e_6 \\ \end{pmatrix}$$

and $e_i$ is a unit vector along the $i^{th}$ joint axis.
2. 1. 2. Forward kinematics

Unlike the inverse kinematic problem, 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. Equation (4) representing the inverse kinematic
solution cannot be inverted to find $\dot{q}$ for a given $\dot{I}$, because $q$ does not explicitly occur in (4). 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

As was the case in kinematic modeling, there are two basic problems in dynamic modeling [28, 29]: forward and inverse dynamics. The latter consists to find the joint force/torque from a given EE motion. The former involves finding 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:

$$\tau = M_c(q)\ddot{q} + N_c(q, \dot{q})\dot{q} + G_c(q)$$  \hspace{1cm} (6)$$

where $M_c(q)$ is the inertia matrix, $N_c(q, \dot{q})$ and $G_c(q)$ 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:

$$M_a \ddot{I} + V_a \dot{I} + K_a F = \tau_m$$  \hspace{1cm} (7)$$

where

$$M_a = M_a I_{6x6} - \frac{2\pi}{np} (J_a + n^2 J_m) I_{6x6}$$  \hspace{1cm} (8)$$

$$V_a = V_a I_{6x6} - \frac{2\pi}{np} (b_a + n^2 b_m) I_{6x6}$$  \hspace{1cm} (9)$$

$$K_a = K_a I_{6x6} = \frac{P}{2\pi I} I_{6x6}$$  \hspace{1cm} (10)$$

where the various parameters are described in the nomenclature.

In order to include the actuator’s dynamics, written in joint space, it is necessary to transform the motion Equation (6) into Cartesian space, using Equation (4) as:

$$M_j(q)\ddot{I} + N_j(q, \dot{q})\dot{q} + G_j(q) = F$$  \hspace{1cm} (11)$$

where
\[ M_j(q) = J^T M_c(q) J, \]  
\[ N_j(q, \dot{q}) = J^T N_c(q, \dot{q}) - J^T M_c(q) J \frac{d(J^{-1})}{dt} \] and
\[ G_j(q) = J^T G_c(q) \]

with \( J \) being the Jacobian, and \( F \) a 6-directional column of the generalized forces. Combination of Equations (7) and (11) produces the dynamic model including the actuators in joint space as

\[ \overline{M}_j(q) \ddot{q} + \overline{N}_j(q, \dot{q}) \dot{q} + \overline{G}_j(q) = \tau_m \]  
with
\[ \overline{M}_j(q) = K_a J^T M_c(q) J + M_a, \]
\[ \overline{N}_j(q, \dot{q}) = K_a J^T N_c(q, \dot{q}) + (V_a - K_a J^T M_c(q) J \frac{d(J^{-1})}{dt}) J J^{-1} \] and
\[ \overline{G}_j(q) = K_a J^T G_c(q) \]

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:

\[ \overline{M}_c(q) \ddot{q} + \overline{N}_c(q, \dot{q}) \dot{q} + \overline{G}_c(q) = \tau_m \]

with
\[ \overline{M}_c(q) = K_a J^T M_c(q) + M_a J^{-1}, \]
\[ \overline{N}_c(q, \dot{q}) = K_a J^T N_c(q, \dot{q}) + (V_a J^{-1} + M_a J^{-1} \frac{d(J^{-1})}{dt}) \] and
\[ \overline{G}_c(q) = K_a J^T G_c(q) \]

It is noteworthy that in the proposed approach one might include contact effort models. These models generally include friction and other application-specific forces. Such inclusion is very useful in many practical cases as de-flashing 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. First, one might express Equation (19) as:

$$\ddot{q} = \overline{M_e}^{-1}(q)\tau_m - \overline{M_e}^{-1}(q)\left[\overline{N}_e(q, \dot{q})\dot{q} + \overline{G}_e(q)\right]$$  \hspace{1cm} (23)

By noting the robot state as \(x = [x_1^T, x_2^T]^T\), (with \(x_1\) and \(x_2\) being defined in the nomenclature), Equation (23) is transformed as:

$$\overline{M}_e(x_1)\dot{x}_2 + \overline{N}_e(x_1, x_2)x_2 + \overline{G}_e(x_1) = \tau_m$$  \hspace{1cm} (24)

In turns, one might rewrite Equation (24) as follows:

$$\dot{x} = \begin{bmatrix} O_{6\times6} & I_{6\times6} \\
O_{6\times6} & 0_{6\times6} \end{bmatrix}x - \begin{bmatrix} \theta_{6\times1} \\
\overline{M}_e^{-1}(x_1) \left[\overline{N}_e(x_1, x_2)x_2 + \overline{G}_e(x_1)\right] \end{bmatrix} + \begin{bmatrix} O_{6\times6} \\
\overline{M}_e^{-1}(x_1) \end{bmatrix}\tau_m$$  \hspace{1cm} (25)

with

$$\overline{M}_e(x_1) = K_oJ^T(x_1)M_e(x_1) + M_eJ^{-1}(x_1)$$  \hspace{1cm} (26)

and

$$\overline{N}_e(x_1, x_2) = K_oJ^T(x_1) + (V_a - J^{-1}(x_1) + M_a\frac{d(J^{-1}(x_1))}{dt})$$  \hspace{1cm} (27)

In order to derive the discrete-time dynamic model of the robot, Equation (25) is written in the form:

$$\dot{x} = Fx - D(x) + B(x)\tau$$  \hspace{1cm} (28)

with

$$F = \begin{bmatrix} O_{6\times6} & I_{6\times6} \\
O_{6\times6} & 0_{6\times6} \end{bmatrix}, \quad D(x) = \begin{bmatrix} \theta_{6\times1} \\
\overline{M}_e^{-1}(x_1) \left[\overline{N}_e(x_1, x_2)x_2 + \overline{G}_e(x_1)\right] \end{bmatrix}, \quad \text{and}$$  \hspace{1cm} (29)

$$B(x) = \begin{bmatrix} O_{6\times6} \\
-1 \end{bmatrix}$$  \hspace{1cm} (30)

Now, let’s define the sampling time \(h_k\), such that \(h_k \leq t \leq h_{k+1}\), and \(\sum_{k=0}^{N-1}h_k = T\), with \(T\) being the total traveling time and the robot state being assumed invariable between two sampling points \(k\) and \(k+1\), and defined as:

$$x(t) = x(h_k), \text{ for } k = 0, 1, \ldots, N - 1, \quad h_k \leq t \leq h_{k+1}$$  \hspace{1cm} (31)

The equivalent discrete-time model to (25) is given as [8]:

$$x_{k+1} = F_d(h_k)x_k - D_d(x_k, h_k) + B_d(x_k, h_k)\tau_k$$  \hspace{1cm} (32)
where \( F_d, D_d, B_d \) are the discrete equivalents to \( F, D, B \) matrices. By assuming the system invariant between two sampling times \( h_k \) and \( h_{k+1} \), the relationships between these pair of matrices are developed through a first order approximation as:

\[
F_d(h_k) = F_d(k+1,k) = e^{Fh_k} \approx I + Fh_k = \begin{bmatrix} I_{6\times6} & h_k I_{6\times6} \\
O_{6\times6} & I_{6\times6} \end{bmatrix}
\]

(33)

\[
D_d(x_k, h_k) = \int_0^{h_k} F_d(h_k-t)D(x_k) \, dt = \begin{bmatrix} h_k^2 I_{6\times6} \\
h_k I_{6\times6} \end{bmatrix} \begin{bmatrix} x_{2k} \\
x_{2k} \end{bmatrix}
\]

(34)

\[
B_d(x_k, h_k) = \int_0^{h_k} F_d(h_k-t)B(x_k) \, dt = \begin{bmatrix} h_k^2 I_{6\times6} \\
h_k I_{6\times6} \end{bmatrix} \begin{bmatrix} \tilde{M}^{-1}(x_{1k}) \\
\tilde{M}^{-1}(x_{1k}) \end{bmatrix}
\]

(35)

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 as:

\[
x_{k+1} = \begin{bmatrix} I_{6\times6} & h_k I_{6\times6} \\
O_{6\times6} & I_{6\times6} \end{bmatrix} x_k - \begin{bmatrix} h_k^2 I_{6\times6} \\
h_k I_{6\times6} \end{bmatrix} \begin{bmatrix} \tilde{N}_c(x_{1k}, x_{2k}) \\
\tilde{N}_c(x_{1k}, x_{2k}) \end{bmatrix} + \begin{bmatrix} \tilde{G}_c(x_{1k}) \end{bmatrix} - \begin{bmatrix} \tau \end{bmatrix}
\]

(36)

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 the actuators. Some of these constraints are defined in joint space while others are in task space.

3.1.1. Robot Constraints

- Dynamic state equations: These consists of Equation (36), which is rewritten for later ease of use as:

\[
x_{k+1} = f_{d_k}(x_k, \tau_k, h_k),
\]

(37)

where \( f_{d_k} \) encapsulates the second member of Equation (36)
- Link intermediate length limits:
  \[ l_{\min} \leq l_k \leq l_{\max}, \text{ with } k = 0, 2, \ldots, N, \text{ and } l_{\max} = \Theta_{\max}(x) \]  
  (38)

- Singularity avoidance:
  Singularities are particular poses in which the robot becomes uncontrollable. It is crucial that they are compensated for in 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 \( J^{-1} \) is not available. Several studies had been dealt with the problem and many singularity avoidance algorithms were proposed [9-13, 16, 17, 31-33]. A common kinematic performance index related to singularity avoidance is the manipulability measure [27]. Accordingly, by defining the manipulability measure as
  \[ w(x_{1k}) = \frac{1}{\text{det}(J(x_{1k})J^T(x_{1k}))} \]  
  (39)

  The following singularity avoidance function can be used:
  \[ \sigma(x_{1k}) = \frac{1}{\sqrt{\text{det}(J(x_{1k})J^T(x_{1k}))}} \]  
  (40)

- 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/torques would be constrained within the limits. In this paper, the robot torques is assumed to belong to a compact and bounded set \( C \subset \mathbb{R}^{6N} \), expressed as:
  \[ C = \{ \tau_k \in \mathbb{R}^{6N}, \text{ such that } \tau_{\min} \leq \tau_k \leq \tau_{\max}, \ k = 0, \ldots, N - 1 \} \]  
  (41)

- Sampling period limits:
  If the overall robot travel time \( T \) is too small, there may be no admissible solution to the optimal control problem since the torque constraints place indirect boundaries on the path traverse time. On the other hand, the sampling period \( h_k \) must be smaller than the system’s smallest time mechanical constant in order to prevent the system from being uncontrollable between two control times. The determination of such a time-mechanical constant and limits of sampling periods is beyond the scope of this paper. They are assumed to be available through a
prior study and analysis of the system’s dynamics. From these limits, a tradeoff is made in this paper through variation of the sampling period within an admissible domain $H$ defined as:

$$H = \left\{ h_k \in \mathbb{R}^+ \text{, such that } h_{\text{min}} \leq h_k \leq h_{\text{max}} \right\}$$

(42)

### 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]. These constraints are expressed by imposing the EE to pass through a set of specified poses (Figure 3). These poses are quantified by a set of $L$ pairs $(p, R)$ with $p_i$ referring to the Cartesian position, and $R_i$ to the orientation of the $i^{th}$ imposed pose on the EE, such that:

$$\|p_i - p\|_{T_{\text{PassThl}}} = 0,$$

(43)

and

$$\|\text{vect}(R_i^T R_i)\|_{T_{\text{PassThk}}} = 0$$

(44)

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

$$s_{l}(x) = \left\| p_i - p \right\|_{T_{\text{PassThl}}} = 0, \quad s_{l}(x) = \left\| \text{vect}(R_i^T R_i) \right\|_{T_{\text{PassThk}}} = 0, \quad l = 1, \ldots, L$$

(45)

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

$$g_1(x) = l_{\text{min}} - \Theta(x) \leq 0, \quad g_2(x) = \Theta(x) - l_{\text{max}} \leq 0,$$

$$g_3(\tau) = \tau_{\text{min}} - \tau \leq 0, \quad g_4(\tau) = \tau - \tau_{\text{max}} \leq 0$$

(46)

For the sake of development simplicity, all inequality constraints will be noted as $g_j(x, \tau, h) \leq 0, \quad j=1,\ldots,4$, regardless whether 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, simultaneously relates energy consumption, travel 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 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 $x_s$ to a final pose $x_f$, within a minimum number $N$ of discretizations. 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 that apply this approach [30]. For strongly non-linear and coupled mechanical systems (including the PKM at hand) this is simply impractical, even if symbolic calculation is used. The second approach uses a fixed number of discretisations $N$ and varies the sampling periods $h_k$. This is equivalent to bring the robot from an initial pose $x_s$ to a final pose $x_f$, 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. Singularity avoidance is included through minimization of Equation (40). In continuous-time, the optimal control problem can be stated as follows:

\[
E = \min_{\tau(t) \in \mathbb{C}, h \in \mathbb{H}} \int_{t_0}^{t_T} \left( \tau(t)^T U \tau(t) + \mathbf{1}_1^T + \frac{1}{2} x_2^T(t) Q x_2(t) + \delta \sigma(x(t)) \right) dt \quad (47)
\]

Subject to constraints (37), (38), (41), (42), (43) and (44).

In the optimization criterion (47), $U$, $Q$, $\mathbf{1}_1$ and $\delta$ are, respectively, the weight factors associated to the electric energy, kinetic energy, travel time and singularity avoidance. The corresponding discrete-time optimal control problem consists of finding the optimal sequences $(\tau_0, \tau_1, ..., \tau_{N-1})$ and $(h_0, h_1, ..., h_{N-1})$, allowing the robot to move from an initial state $x_0 = x_s$ to a target state $x_N = x_f$, while minimizing the cost $E_d$:
\[
\begin{align*}
\text{Min} E_d & = \left\{ \sum_{k=1}^{N} \left[ \mathbf{T}_k \mathbf{U} \mathbf{T}_k^T + \mathbf{t}_1 + x_{2k} \mathbf{Q} x_{2k} + \delta \mathbf{S} (x_{1k}) \right] \mathbf{h}_k \right\} \\
\text{Subject to:} & \\
\mathbf{x}_{k+1} &= f_{d_i} (x_k, \tau_k, h_k), \quad k = 0, \ldots, N-1 \\
\mathbf{g}_j (x_k, \tau_k, h_k) & \leq 0, \quad j = 1, \ldots, 4, \quad k = 0, \ldots, N-1 \\
\mathbf{s}_i (x_k) &= 0, \quad i = 1, \ldots, 2L, \quad k = 0, \ldots, N
\end{align*}
\]

4. Nonlinear Programming Formulation

4.1. Augmented Lagrangian Approach

During the course of solving the constrained non-linear multi-objective optimal control problem in Equation (48), the Augmented Lagrangian (AL) function is used to transform it into a non-constrained problem 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], and subsequently improved by several authors [24-26]. It 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 Lagrangian is generally non convex (in the presence of non convex constraints as in the case for the problem considered here), AL may be convexified to some extent with a judicious choice of penalty coefficients [26]. An outline of the AL implementation procedure for the case at hand is given at the end of section 5 and an implementation flowchart diagram appears in the appendix. The AL function transforming the constrained optimal control problem into an unconstrained one is written as:

\[
L_{d}(x, \tau, h, \lambda, \rho, \sigma) = \sum_{k=1}^{N} \left[ \mathbf{T}_k \mathbf{U} \mathbf{T}_k^T + \mathbf{t}_1 + x_{2k} \mathbf{Q} x_{2k} + \delta \mathbf{S} (x_{1k}) \right] \mathbf{h}_k + \sum_{k=0}^{N-1} \left\{ \lambda_{k+1} (x_{k+1} - f_{d_i} (x_k, \tau_k, h_k)) \right\} + \\
\sum_{k=0}^{N-1} \left[ \sum_{l=1}^{2} \sum_{j=1}^{2} \Psi_{p_i} (\sigma_l, s'(x_l)) + \sum_{j=1}^{2} \Phi_{p_i} (\rho_j, g_j (x_j, \tau_j, h_j)) \right] + \sum_{i=1}^{2} \sum_{j=1}^{2} \Psi_{p_i} (\sigma_l, s'(x_l)) \quad (49)
\]

where the function \( f_{d_i} (x_k, \tau_k, h_k) \) is defined by the discrete state Equation (37) at the sampling time \( k, N \) is the sampling number, \( \lambda \in \mathbb{R}^{12N} \) designates the adjoin (or co-state) obtained from the adjunct equations associated with state equations, \( \sigma, \rho \) are Lagrange multipliers with appropriate dimensions, associated with equality and inequality constraints and \( \mu_s, \mu_g \) 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:

\[ \Psi_{\mu}(a, b) = (a + \frac{\mu}{2} b)^2 \quad \text{and} \quad \Phi_{\mu}(a, b) = \frac{1}{2\mu} \left\{ \max(0, a + \mu b) \right\} - \|b\|^2 \]  

(50)

where \( a \) and \( b \) refer respectively to Lagrange multipliers and the left hand side of equality and inequality constraints. The Karush-Kuhn-Tucker first order optimality necessary conditions \[25\] require that for \((x_0, \tau, h_1, x_1, \tau_2, h_2, \ldots, x_N, \tau_{N-1}, h_{N+1})\) to be a solution to the problem, the following parameters must exist: positive Lagrange multipliers \((\lambda, \rho)\), unrestricted sign multipliers \(\sigma\), and finite positive penalty coefficients \((\mu, \mu_p)\) such that:

\[ \frac{\partial L_u}{\partial x} = 0, \quad \frac{\partial L_u}{\partial \tau} = 0, \quad \frac{\partial L_u}{\partial h} = 0, \quad \frac{\partial L_u}{\partial \lambda} = 0, \quad \frac{\partial L_u}{\partial \rho} = 0, \quad \frac{\partial L_u}{\partial \sigma} = 0, \quad \text{and} \quad \rho_i (g(x, \tau, h)) = 0, \quad \sigma_i (s(x)) = 0, \quad g(x, \tau, h) \leq 0 \]  

(51)

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 Equation (37), \( f_{d_i}(x_i, \tau_i, h_k) \) contains the inverse of the total inertia matrix \( \overline{M}^{-1}(x) \) of the PKM, including struts and actuators, as well as their Coriolis and centrifugal wrenches \( \overline{N}_{\tau_i}(x_i, x) \). These would take several pages long to display. In developing the first order optimality conditions and computing the co-states \( \lambda \), 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 when symbolic calculation is used.

4.2. Constrained Linear-Decoupled Formulation

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

**Theorem:**

Under the invertibility condition of the inertia matrix, the control law defined in the Cartesian space as (after removing bars and \( c \) indexes for writing simplicity):
allows the robot to have a linear and decoupled behavior with a dynamic equation:

$$\dot{x}_2 = v$$  \hspace{1cm} (53)$$

where $v$ is an auxiliary input.

This was derived by simply substituting the proposed control law (52) into the dynamic model (19). One gets

$$M(x_1)\dot{x}_2 = M(x_1)v$$

Since $M(x)$ is invertible, it follows that $\dot{x}_2 = v$.

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

$$x_{k+1} = F_{d_k}x_k + Z_{h_k}v_k = f_{d_k}^o(x_k, v_k, h_k)$$  \hspace{1cm} (54)$$

with

$$F_{d_k} = \begin{bmatrix} I_{6\times 6} & h_k I_{6\times 6} \\ O_{6\times 6} & I_{6\times 6} \end{bmatrix}$$ and $$Z_{h_k} = \begin{bmatrix} h_k^2 I_{6\times 6} \\ 2h_k I_{6\times 6} \\ h_k I_{6\times 6} \end{bmatrix}$$

Notice that this formulation reduces drastically the number of computations by alleviating us the need for calculation of the inertia matrix inverse and its derivatives with respect to state variables at each iteration, which eases calculation of the co-states. The non-linearity 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 $(h_0, h_2, ..., h_{N-1})$, $(v_0, v_2, ..., v_{N-1})$, (v_0, v_2, ..., v_{N-1})
allowing the robot to move from an initial state $x_0 = x_s$ to a final state $x_N = x_f$, while minimizing the cost function:

$$E_D = \min_{x_0, x_N, \tau} \left\{ \sum_{k=0}^{N-1} \left[ M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \right]^T U \right\}$$

$$\left[ M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \right] + t_i + x^T Q x_{2k} + \delta \sigma (x_{1k}) \right\} h_k \right\}$$

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

$$\tau_{\min} \leq M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \leq \tau_{\max}$$

Moreover, inequality constraints $g_3$ and $g_4$ can be rewritten as:

$$g_3^D(x_k, v_k) = \tau_{\min} - \left[ M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \right] \leq 0$$

$$g_4^D(x_k, v_k) = \left[ M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \right] - \tau_{\max} \leq 0$$

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

$$\min_{x_0, x_N, \tau} E_D$$

Subject to:

$$x_{k+1} = f^D(x_k, \tau_k, h_k), \quad k = 0, \ldots, N-1$$

$$g^D(x_k, v_k, h_k) \leq 0, \quad j \in \{1, 2, \ldots, J\}$$

$$s^D(x_k) = 0, \quad i \in \{1, \ldots, I\}$$

4.3. Augmented Lagrangian for the Decoupled Formulation (ALD)

Now, mutatis mutandis, the augmented Lagrangian associated with the decoupled form (P) is:

$$L(x, v, h, \lambda, p, \sigma) = \sum_{k=0}^{N-1} \left\{ \left[ M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \right]^T U \right\}$$

$$\left[ M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \right] + t_i + x^T Q x_{2k} + \delta \sigma (x_{1k}) \right\} h_k \right\} +$$

$$\sum_{k=0}^{N-1} \left[ \lambda_{k+1}^T (x_{k+1} - f^D_{a_k}(x_k, v_k, h_k)) \right] + \sum_{k=0}^{N-1} h_k \left[ \sum_{i=1}^{I} \sum_{j=1}^{J} \psi_i (\sigma_i, s^D(x_k)) \right]$$

19
\[
\sum_{j=1}^{2} \Phi_{p_j} (p_j^i, g_j^P(x_{i_k}, \tau_{i_k}, h_i)) + \sum_{j=1}^{2} h_j \Psi_{r_j} (\sigma_j^i, s_j^P(x_j))
\]

(59)

where the function \( f_{d_j}^P(x_{i_k}, \tau_{i_k}, h_i) \) is defined by Equation (54) at time \( k \), and other parameters appearing in (59) are defined above.

Again, the development of the first order Karush-Kuhn-Tucker optimality necessary conditions require that for \((x_o, v_i, h_1, x_1, v_2, h_2, \ldots, x_N, v_{N-1}, h_{N-1})\) to be solution to the problem \((P)\), the following parameters must exist: positive Lagrange multipliers \( (\lambda_i, \rho_i) \), unrestricted sign multipliers \( \sigma_k \), and finite positive penalty coefficients \( (\mu_i, \mu_x) \) such that equations (51) are satisfied for the decoupled formulation.

The co-states \( \lambda_k \) are determined by backward integration of the adjunct state equation yielding, for \( k = N-1, \ldots, 1 \):

\[
\lambda_{k-1} = -2 h_k \frac{\partial \left[ M(x_{ik}) v_k + N(x_{ik}, x_{ik2}) x_{ik2k} + G(x_{ik}) \right]}{\partial x_k} U \left[ M(x_{ik}) v_k + N(x_{ik}, x_{ik2}) x_{ik2k} + G(x_{ik}) \right] \\
-2Qx_{ik2k} h_k - \delta V x_{ik} \sigma (x_{ik}) - \sum_{j=1}^{2} h_j \left[ \sum_{i=1}^{N} \sum_{j=1}^{2} \nabla x_i \Psi_{r_j} (\sigma_j^i, s_j^P(x_j)) \right] - \\
h_k \left[ \sum_{j=1}^{2} \nabla x_i \Phi_{p_j} (p_j^i, g_j^P(x_i, v_k, h_i)) \right]
\]

(60)

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

\[
\nabla_{h_k} L^P = \left[ M(x_{ik}) v_k + N(x_{ik}, x_{ik2}) x_{ik2k} + G(x_{ik}) \right]^T U \left[ M(x_{ik}) v_k + N(x_{ik}, x_{ik2}) x_{ik2k} + G(x_{ik}) \right] \\
+ x_{ik2k} Q x_{ik} + 1_k + \delta \sigma (x_{ik}) + \sum_{j=1}^{2} \sum_{i=1}^{N} \Psi_{r_j} (\sigma_j^i, s_j^P(x_j)) + \sum_{j=1}^{2} \Phi_{p_j} (p_j^i, g_j^P(x_i, v_k))
\]

(61)

The gradient of the Lagrangian with respect to acceleration variables is

\[
\nabla_{v_k} L^P = 2 M(x_{ik}) U^T \left[ M(x_{ik}) v_k + N(x_{ik}, x_{ik2k}) x_{ik2k} + G(x_{ik}) \right] h_k + f_{d_j}^P \lambda_k + \\
h_k \left[ \sum_{j=1}^{2} \nabla x_i \Phi_{p_j} (p_j^i, g_j^P(x_i, v_k, h_i)) \right]
\]

(62)

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

4.4.1. Initial Solution

To speed up 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 trapezoidal velocity profile. This solution is divided into three zones, starting with acceleration zone of 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 applied. 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.

$$h_k = t_{k+1} - t_k = \frac{t_f - t_0}{N}, \quad k = 0, 1, \ldots, N - 1$$ (63)

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 method is used at each iteration of the optimization process. This method allows the computing of an approximate Hessian matrix using only the first derivative information without the need to store this 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 [18]. At the $(t+1)^{th}$ iteration, set $\alpha_i = v_{r,i} - v_i$ as the update of the control variable $v$, $\beta_i = \nabla_v L_{r,i} - \nabla_v L_i$ as the update of the gradient and $H_i^{-1}$ the approximation of the inverse of the Hessian. The inverse of the approximate Hessian $H_i^{-1}$ can be obtained using the BFGS update formula:

$$H_{i+1}^{-1} = H_i^{-1} + \frac{a_i \alpha_i^T}{\beta_i \beta_i} + \frac{\alpha_i \alpha_i^T}{\beta_i \beta_i}, \quad \text{with} \quad V_i = I - (\beta_i \alpha_i^T) / (\beta_i \beta_i)$$ (64)

The following pseudo-code describes the BFGS two-loop iterative procedure used to compute the search direction $H_i^{-1} \nabla_v L_i$ efficiently by using the last $m$ pairs of $(\alpha_i, \beta_i)$:

$$s \leftarrow \nabla_v L_i$$

For $i = t-1, t-2, \ldots, t-m$
\( \gamma_i \leftarrow \alpha_i^i s/\beta_i^i a_i; \)
\[ s \leftarrow s - \gamma_i \beta_i; \]
End (for) \hspace{1cm} (65)
\[ r \leftarrow (H_i^0)^{-1} s; \]
For \( i = t - m, t - m, \ldots, t - 1 \)
\[ \delta_i \leftarrow \beta_i r/\beta_i^i a_i; \]
\[ r \leftarrow r + (\gamma_i - \delta_i) a_i; \]
End (for)
Stop with result \( H_t^0 \nabla_j L_i = r \)

where \( (H_i^0)^{-1} \) is the initial approximation of the inverse of the Hessian matrix. One can set it as:
\( (H_i^0)^{-1} = \varsigma_i I \), with \( I \) is the identity matrix of appropriate dimension, and \( \varsigma_i = (\alpha_{i_{-1}}^r \beta_{i_{-1}}) / (\beta_{i_{-1}}^r \beta_{i_{-1}}) \).

4. 4. 3. Overall Solution Procedure

A systematic procedure flowchart for implementation of the augmented Lagrangian appears in the appendix. In this procedure, after selecting robot parameters, task definition, (such as starting, intermediate and final poses), workspace limitations and simulation parameters (block 1), the kinematic unit (block 2) defines a feasible solution satisfying initial and final poses. Then the inner optimization loop (block 3) solves for ALD minimization with respect to sampling periods and acceleration control variables to give the robot dynamic state. This state is then tested to verify whether it is within feasibility tolerances. The feasibility decision is made by testing the norms of all equality and inequality constraints against given tolerances. If the feasibility test fails, the inner optimization unit is restarted. If the feasibility test succeeds, i.e., the current penalty values maintain near-feasibility of iterates, a convergence test is made against optimal tolerances. If convergence is achieved the optimal results are displayed and the program ends. If a condition of non-convergence occurs, further steps are made into the dual part of the ALD (block 4), 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 the multipliers are kept unchanged and penalty values are decreased to ensure that the next sub-problem will place more emphasis on
reducing constraints violations. In both cases the tolerances are decreased to force the subsequent primal iterates to generate 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 [39]. Preliminary results are encouraging. This PKM consists of a two degrees of freedom planar parallel manipulator. The robot kinematic and dynamic models considered have been developed in [37]. A schematic of the manipulator is depicted in Figure 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, 3, 4, and 5 for the first leg and links 2, 6, 7, and 8 for the second leg. Prismatic actuators actuate the link 3 and 8, 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 2D space with a constant orientation.

5.2. Kinematic and Dynamic Analysis

As illustrated in Figure 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 base, where \( O' \) is the origin of frame \( A \) and \( O \) the origin of frame \( B \). To characterize the planar four-bar parallelogram, the chains \( A_iB_1 \) and \( A_2B_2 \) are considered as shown in Figure 4. Vectors \( A_iA \) and \( A_iB \) \((i=1,2)\) define the positions of points \( A_i \) in frames \( A \) and \( B \) respectively. Vectors \( h_{iB} \) \((i=1,2)\) define the position of \( B_i \) points in frame \( B \). The geometric parameters of the manipulator are \( A_iB_i = L \ (i=1,2), \ A_1A_2 = 2r \ and \ B_1B_2 = 2R \).

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:

\[
J_i \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = J_x \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}
\]

(66)

where \( J_i \) and \( J_x \) are respectively, the 2×2 inverse and forward Jacobian matrices of the manipulator, which can be expressed as
\[ J_i = \begin{bmatrix} y - y_1 & 0 \\ 0 & y - y_2 \end{bmatrix}, \ J_s = \begin{bmatrix} r + x + R & y - y_1 \\ x - r + R & y - y_2 \end{bmatrix} \] (67)

If \( J_i \) is non-singular, the Jacobian matrix of the manipulator can be obtained as

\[ J = J_i^{-1} J_s = \begin{bmatrix} (r + x - R)/(y - y_1) & 1 \\ (x - r + R)/(y - y_2) & 1 \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \] (68)

Accordingly, it is clear that singularity occurs when one of the following cases holds [12, 13]:

1\textsuperscript{st} case: \( |J_i| = 0 \) and \( |J_s| \neq 0 \). This case is known as the type one 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.

2\textsuperscript{nd} case: \( |J_i| \neq 0 \) and \( |J_s| = 0 \). This case is known as the type two 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.

3\textsuperscript{rd} case: \( |J_i| = 0 \) and \( |J_s| = 0 \). This corresponds to the type three 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 parameter condition given by:

\[ L + r = R \] (69)

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

\[
\begin{bmatrix}
\dot{\tau}_1 \\
\dot{\tau}_2
\end{bmatrix} = \begin{bmatrix}
(m_p + \frac{4}{3} m_i) + 2(m_s + \frac{m_i}{3})(J_{11} + J_{21}) & (m_s + \frac{2}{3} m_i)(J_{11} + J_{21}) \\
(m_s + \frac{2}{3} m_i)(J_{11} + J_{21}) & m_p + 2m_s + \frac{8}{3} m_i
\end{bmatrix} \begin{bmatrix}
\dot{x} \\
\dot{y}
\end{bmatrix} + \begin{bmatrix}
L^2 \frac{\dot{x}^2}{\sqrt{x^2 + y^2}} (m_s + \frac{2}{3} m_i) - \frac{\dot{\chi}^2}{\sqrt{x^2 + y^2}} (J_{11} + J_{21}) \\
L^2 (m_s + \frac{2}{3} m_i) + \frac{\dot{\chi} \dot{y}}{\sqrt{x^2 + y^2}} (J_{11} + J_{21})
\end{bmatrix} \begin{bmatrix}
\ddot{J}_{11} \sqrt{J_{11}} (x + r - R)^{3/2} + J_{21} \sqrt{J_{21}} (x - r + R)^{3/2} \\
\ddot{J}_{11} \sqrt{J_{11}} (x + r - R)^{3/2} + J_{21} \sqrt{J_{21}} (x - r + R)^{3/2}
\end{bmatrix} + \begin{bmatrix}
\dot{y} (\dot{x} - \dot{\chi}) (\chi^2 + y^2)^{3/2} + g (m_s + m_i) (J_{11} + J_{21}) \\
\dot{\chi}^2 (m_s + \frac{2}{3} m_i) + \frac{\dot{\chi} \dot{y}}{\sqrt{x^2 + y^2}} (J_{11} + J_{21}) + g (m_p + m_i)
\end{bmatrix} \] (70)
More details on the derivation of the dynamic model may be found in [37].

Following the streamline developed in previous sections, a discrete-time state-space model associated with state Equation (70) is:

\[
x_{k+1} = \begin{bmatrix} I_{2 \times 2} & h_k I_{2 \times 2} \\ O_{2 \times 2} & I_{2 \times 2} \end{bmatrix} x_k - \overline{M}_c(x_k) \begin{bmatrix} \frac{h_k^2}{2} I_{2 \times 2} \\ h_k I_{2 \times 2} \end{bmatrix} \left[ \overline{N}_c(x_{1k}, x_{2k}) x_{2k} + \overline{G}_c(x_{1k}) - \tau_k \right]
\]

(71)

The optimal control problem consists of minimizing criterion (47) subject to dynamic Equation (71), equality and inequality constraints (37-44), and the following specific constraints.

- **Workspace limitations:**

  \[
  x_{k_{\min}} \leq x_k \leq x_{k_{\max}}, \quad y_{k_{\min}} \leq y_k \leq y_{k_{\max}} \quad \text{for} \quad k = 0, 2, \ldots, N
  \]
  (72)

- **Singularity avoidance:**

  In our considered case study, the first type singularity constraint may be expressed by:

  \[
  (y_k - y_{1k}) \left( y_k - y_{2k} \right) \geq \epsilon_1,
  \]
  (73)

  whereas the second is given as:

  \[
  (x_k + \text{sgn}(x_k)(r-R)) \geq \epsilon_2
  \]
  \[
  \epsilon_1 \quad \text{and} \quad \epsilon_2 \quad \text{represent small positive tolerances.}
  \]

  The third singularity type concerns 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:

  \[
  s(x) = \left\| \mathbf{p} - \mathbf{p}_f \right\| - T_{\text{PassThL}} = 0
  \]
  (75)

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

5.3. Simulation Data and Scenario

The following numeric values are used: The EE mass is \(m_{EE} = 200.0 \, \text{kg}\), 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 \(r = 0.075 \, \text{m}\), \(R = 1.2030 \, \text{m}\) and the strut length \(L = 1.9725 \, \text{m}\). Table 1 shows the limits of the workspace, actuator torques and
sampling periods. For ALD, the following parameter values had been taken, for $\omega = 0.5$, $\eta = 0.5$, $\alpha_w = \alpha_d = 0.4$, $\beta_r = \beta_t = 0.4$, $w_0 = \eta_0 = \eta_{0t} = 10^{-2}$, $w^* = \eta^* = 10^{-5}$, $\gamma_1 = 0.25$, $\gamma_2 = 1.2$, $v = 0.01$, $\tilde{v} = 0.3$. The initial Lagrange multipliers $\sigma_0$, $\rho_0$ components are set to zero. The singularity weight is $\delta = 1$. The maximum value for $\delta_{\max} = 10^{42}$, and the minimum value for $\delta_{\min} = 10^{-42}$.

The simulated 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 $x_f = 0.7$, $y_f = 1.6$ (in meters). The initial and final linear and angular velocities are equal to zero. The maximum velocity is $0.2m/sec$ and maximum acceleration is $2 m/sec^2$. The maximum allocated time for this trajectory is $10$ sec. In the presented simulations, the focus is on time-energy constrained trajectory planning by the augmented Lagrangian. A more kinematic related performance evaluation and design for a similar case study may be found in [37, 39]. Typically, four simulation objectives are considered:

- Compare robot trajectories for different values of the weights $U$, $Q$, $I$, and $\delta$.
- Assess the effects of the dynamic parameters changes on the augmented Lagrangian sensitivity and on the behavior of the PKM.
- At which precision vs. time consumption, the augmented Lagrangian achieves passage satisfaction through imposed poses?
- To what extent the number of inner and outer iterations of AL impacts PKM performance vs. CPU Time?

To start, Figure 5 shows the velocity profile used to initialize the augmented Lagrangian. Figures 6 and 7 show the simulation results for both initial kinematic and augmented Lagrangian solutions. In part (a) of these figures, the first plot from the top shows the displacement along $x$-axis of the end-effector point of operation. The second plot shows the displacement along $y$-axis of the end-effector point of operation. The third shows the instantaneous values of 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 quickly gets torque values outside the admissible domain resulting in high values for energy cost. With the augmented Lagrangian however, with four inner and 7 outer iterations, the variations of
the energy consumption increase smoothly and monotonously. Figure 8 displays the simulation outcomes for only the 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) compared to a trajectory computed with only the 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), (0.5, -1.0), all in meters. Figure 9 shows the trajectory corresponding to passage through imposed poses. With 10 dual iterations and 7 primal iterations one gets a precision of \(7 \times 10^{-4}\), which confirms the well-known constraints satisfaction performance of ALD for constrained optimization problems as compared to its counterparts such as penalty methods. Furthermore, we observe that the proposed variational approach has not only been successful in finding a precision trajectory, but also the obtained trajectories are singularity-free and minimum time-energy. 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, \(t_r = \sum_{k=0}^{N-1} h_k\) is the total travel time, Energy = \(\sum_{k=0}^{N-1} [(\tau_k U_{\tau_k}^T + x_{\tau_k} Qx_{\tau_k}^T)]\) is the consumed electric and kinetic energy, and APEq and APIneq for Achieved Precision for Equality and Inequality constraints satisfaction, respectively. The values shown for the total travel time \(t_r\), Energy, and AP correspond to those computed for the last outer iteration. Although the purpose of the paper is offline programming, the whole processing time necessary to get a solution is quite long. It takes about 11 minutes to get the trajectory of the first example (without imposed passages). This is mainly due to the calculations of several gradients at each iteration. Several ways are undertaken to reduce the computational burden. One of these would be eliminating the square root in the measure of manipulability.

Table 1. Limits of workspace, actuator torques and sampling periods

<table>
<thead>
<tr>
<th>Parameter</th>
<th>(x)-coordinate (m)</th>
<th>(y)-coordinate (m)</th>
<th>(\tau_1) (N)</th>
<th>(\tau_2) (N)</th>
<th>(h) (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>0.8</td>
<td>-0.720</td>
<td>550</td>
<td>700</td>
<td>0.7</td>
</tr>
<tr>
<td>min</td>
<td>-0.8</td>
<td>-1.720</td>
<td>-550</td>
<td>-700</td>
<td>0.005</td>
</tr>
</tbody>
</table>
Table 2. Convergence history of Minimum Time-Energy Planning with ALD

<table>
<thead>
<tr>
<th>NDisc</th>
<th>NPrimal</th>
<th>NDual</th>
<th>CPU (sec)</th>
<th>( t_f ) (sec)</th>
<th>Energy (J)</th>
<th>APEq</th>
<th>APIneq</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4</td>
<td>7</td>
<td>12.62</td>
<td>9.50</td>
<td>6874.37</td>
<td>3.10^{-3}</td>
<td>3.10^{-3}</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>7</td>
<td>29.25</td>
<td>8.01</td>
<td>5156.28</td>
<td>10^{-3}</td>
<td>10^{-3}</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>10</td>
<td>35.54</td>
<td>7.31</td>
<td>4910.39</td>
<td>5.10^{4}</td>
<td>4.10^{4}</td>
</tr>
<tr>
<td>30</td>
<td>5</td>
<td>10</td>
<td>69.83</td>
<td>6.07</td>
<td>4010.23</td>
<td>2.10^{-5}</td>
<td>3.10^{-5}</td>
</tr>
</tbody>
</table>

Fig. 5. Initial solution, a trapezoidal velocity profile

(a) - Variations of \( x, y \) coordinates of the EE, and sampling periods
(b) - Variations of torques \( \tau_1, \tau_2 \), and energy

28
Fig. 7. Simulation results with the ALD, (Minimum Time-Energy)  
(a)- Variations of $x, y$, coordinates of the EE, and sampling periods  
(b)- Variations of torques $\tau_1, \tau_2$, and energy

Fig. 8. ALD (Minimum Energy)  
(a)- Variations of $x, y$, coordinates of the EE, and fixed sampling periods  
(b)- Variations of torques $\tau_1, \tau_2$, and energy

Fig. 9. ALD Minimum Time-Energy with imposed passage through  
Cartesian positions (0.0, -1.4), (0.4, -1.1), and (0.5, -1.0)  
(a) Variations of $x, y$, coordinates of the EE, and sampling periods  
(b) - Variations of torques $\tau_1, \tau_2$ and energy
5.4. Sensitivity Analysis

The multi-objective control considered so far is dependent on the values of the dynamic parameters of the PKM. As PKMs are strongly non-linear and coupled mechanical systems, several of these parameters such as inertial parameters are known only approximately or may change. For this reason a sensitivity analysis [40] is necessary to assess how robust the proposed approach is to the parameter changes. This is performed by varying the value of the EE mass. Figure 10 shows the ALD simulation with modified EE mass as $m_{EE} = 300.0 \text{ kg}$. One notices that the needed actuator torques and necessary energy and time to achieve the same task are higher, especially at the beginning.

![Fig. 10. Disturbed ALD (Minimum Time-Energy) (Modified mass of EE, $m_{EE} = 30.0 \text{ kg}$)](image)

6. Conclusions and Discussions

The basic contribution of this paper is the formulation and resolution of the trajectory-planning problem of parallel kinematic machines 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 augmented Lagrangian algorithm is used to solve the resulting non-linear and non-convex 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 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 travel time and actuator torques.
than schemes based solely on kinematics. Moreover, it produces smoother trajectories compared to minimum time-based control. Furthermore, the proposed trajectory planning is robust to dynamic parameters changes. This in fact is due to the ability of the Augmented Lagrangian to cope with numerical ill-conditioning problems, as compared to other optimization techniques like penalty methods. Another major advantage of this approach is that one can introduce any type of constraints related to the robot, task or environment, for example 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 anonymous reviewers for the valuable comments and 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.

References


[22] A. Khoukhi, L. Baron, and M. Balazinski, A decoupled approach to time-energy trajectory planning of parallel kinematic machines, Proceedings of 16th CISM-IFToMM Robotics and


[38] Available:  www.mathworks.com/products/matlab/

Appendix: Augmented Lagrangian Algorithm

Fig. 11. Flowchart for Augmented Lagrangian algorithm function and operation
Figure Captions:

**Figure 1.** Overall Off-line Programming Framework of PKMs

**Figure 2.** Geometry of a PKM

**Figure 3.** Illustration of EE passage through imposed poses (positions and orientations)

**Figure 4.** A schematic representation of the planar parallel manipulator

**Figure 5.** Initial solution, a trapezoidal velocity profile

**Figure 6.** Kinematic simulation results

**Figure 7.** Simulation results with the ALD, (Minimum Time-Energy)

**Figure 8.** ALD (Minimum-Energy)

**Figure 9.** ALD Minimum Time-Energy with imposed passage through Cartesian positions

**Figure 10.** Disturbed ALD (Minimum Time-Energy) (modified mass of EE $m_{EE} = 300.0$ kg)

**Figure 11.** Flowchart for Augmented Lagrangian algorithm function and operation

List of Tables:

**Table 1.** Limits of the workspace, actuator torques and sampling periods

**Table 2.** Convergence history of the Minimum Time-Energy Planning with ALD

Vitae

**DR. AMAR KHOUKHI**

Dr. Amar Khoukhi received his Advanced Diploma of University Studies in operation research from University of Paris-IX, France in 1985, an Engineering Doctorate from Software Engineering Dept. Superior School of Telecommunications Engineers, Paris, France in 1991, and a PhD degree in 2007 in robotics, from École Polytechnique, the engineering faculty of university of Montreal.

Over the last several years, Dr. Khoukhi was an associate research and lecturer at Control Systems and Automation Dept., Superior School of Electronics Engineers, Paris, France. From 1998 to 2001 he was a faculty member at the Computer Science Dept., University of Algiers, Algeria. From 2004 to 2007 he held an NSERC Post-Graduate Scholarship and worked as a lecturer and associate research in the Mechanical Engineering Department of École Polytechnique of Montreal, Canada. His research interests include robotics, applied optimization, and intelligent systems.

**DR. LUC BARON**

Dr. Luc Baron received his B.Eng. and M.A. Sc. degrees from École Polytechnique, Montreal, Canada, in 1983 and 1985, respectively, and his 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 École Polytechnique, the engineering faculty of University of Montreal, and as Research Engineer at Walsh Automation Inc. Since 1997, Dr. Baron is Professor in the Department of Mechanical Engineering at École Polytechnique. His current research interests are in the field of robotics and industrial automation, kinematics, synthesis and control of manipulators. Dr. Baron is a member of IEEE, an Associate Member of ASME and a registered engineer in Quebec.
DR. MAREK BALAZINSKI

Dr 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 Czestochowa 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.

Dr Balazinski is currently professor and head of the Manufacturing Section in the Department of Mechanical Engineering at École Polytechnique of Montréal. Over the last ten years, he has published over 100 research papers and he is an active member of CIRP (The International Academy for Production Engineering).