MULTI-OBJECTIVE TRAJECTORY PLANNING FOR REDUNDANT MANIPULATORS USING AUGMENTED LAGRANGIAN

Amar KHOUKHI, Luc BARON and Marek BALAZINSKI
Mechanical Engineering Department,
École Polytechnique of Montreal,
C. P. 6079, Succ. CV, Montreal, QC, Canada H3C 3A7
(amar.khoukhi, luc.baron, marek.balazinski)@polymtl.ca

ABSTRACT

In this paper, a multi-objective trajectory planning system is developed for redundant manipulators. This system involves kinematic redundancy resolution, as well as robot dynamics, including actuators model. The kinematic redundancy is taken into account through a secondary criterion of joint limits avoidance. The optimization procedure is performed subject to limitations on actuator torques and workspace, while passing through imposed poses. The Augmented Lagrangian with decoupling (ALD) technique is used to solve the resulting constrained non convex and nonlinear optimal control problem. Furthermore, the final state constraint is solved using a gradient projection. Simulations on a three degrees of freedom planar redundant serial manipulator show the effectiveness of the proposed system.

Key words: Redundant Manipulators, Trajectory Planning, Augmented Lagrangian, Decoupling, Projection

PLANIFICATION MULTI-OBJECTIVE DE TRAJECTOIRE DES MANIPULATEURS REDONDANTS PAR LAGRANGIEN AUGMENTÉ

RÉSUMÉ

Dans cet article, nous considérons le problème de planification multi-objective de trajectoire des manipulateurs sériels redondants. À partir des modèles cinématique et dynamique du robot, et ceux de l'espace de travail et de la tâche, le problème est formulé dans le cadre du calcul variationnel, comme un programme non linéaire sous contraintes. La redondance cinématique est résolue en appliquant une version modifiée de l'algorithme de la Jacobienne augmentée. La technique du Lagrangien augmenté est ensuite appliquée à une représentation découplée de la dynamique du robot afin de résoudre le problème résultant de commande optimale. Des études de simulations montrent l'efficacité de cette approche comparée à celles développées jusqu'à date, notamment les approches utilisant les méthodes de pénalité ou celles basées uniquement sur la cinématique du robot.

Mots Clés : Manipulateurs Redondants, Planification de trajectoire, Lagrangien augmenté, Découplage, Projection

1. INTRODUCTION

A great advantage of robots is their ability and flexibility to rearrange themselves for new tasks. Utilization of robot's flexibility presupposes effective motion planning. This is aimed at generating trajectories for a specific task, according to a set of desired performance criteria [1, 2]. The task is usually specified in terms of a motion of the end-effector (EE), which results in a geometric path to be followed with a given time law. Moreover, the robot arm is actuated at the joints, thus requiring control actions to be performed by the joint servos. A feasible planning could be satisfactory if the required target performance is not too tight. Otherwise, one must use an optimal (or near optimal) approach, and then test how it is robust to changes of the dynamic parameters [3, 4]. One of the motion control algorithm problems is that the desired trajectory may cause saturation of the speed and/or torques delivered by the joint actuators in the vicinity of singularities. This might occur in many regions of the workspace, due to nonlinear kinematic transformations between task and joint spaces. Furthermore, when the assigned trajectory results are unfeasible due to actuation limits or passing through singular poses, the motion planning system must still generates torques that allow achieving the performance criteria, while avoiding singularities and satisfying other task-related constraints. Several studies had considered this problem [6-9]. Some of them dealt with kinematic redundancy resolution, while others included robot dynamics and force optimization [3-5].

In this paper, the multi-objective trajectory planning problem is formulated based on kinematic and dynamic, including actuators models. The cost functional involves time optimization through sampling period variations and electric energy as well as a measure of manipulability. The resulting constrained non convex and nonlinear optimal control problem is solved using AL technique on a decoupled form of the robot dynamics. The advantage of using such a technique – as compared to other optimization methods like penalty methods – is its ability to deal with non convexities (due mainly to the strong non linear character of the system's constraints) and ill conditioning that may occur during the iterative resolution process. Furthermore, in many applications, such as pick and place or assembly tasks, the final state attainability constraint is a primary issue. This constraint is achieved through a gradient projection algorithm [10-11]. After problem modelling and formulation, simulation results on a three degrees of freedom planar redundant serial manipulator show very encouraging results as compared to other approaches, namely, minimum-time control and kinematic-based methods. In section 2, the kinematic and dynamic models are considered as well as other associated constraints. In section 3, the augmented Lagrangian with decoupling and projection gradient technique is developed. Section 4 provides simulation results, and section 5 concludes this paper.

2. MODELING

2. 1. Kinematic modeling

The forward kinematics problem deals with the determination of the EE motion from a given motion of the joints [12]. At the velocity level, it is expressed in a vector form as:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{q}} \tag{1}$$

where $\boldsymbol{q}(t) = [q_1(t), q_2(t), ..., q_n(t)]^T$ is the n-dimensional joint angles vector, $\mathbf{x}(t) = [x_1(t), x_2(t), ..., x_m(t)]^T$ is the m-dimensional position vector of the EE and $\mathbf{J}$ is the mxn robot's Jacobian [12]. Although this study is applicable to a general n-degrees-of-freedom (DOF) serial manipulator with m-DOF at the Cartesian EE level, it is implemented on a two dimensional positioning robot.

The inverse kinematics problem is the determination of the joints motion from a given EE motion. Because $\mathbf{J}$ is not a square matrix, a kinematic redundancy holds for this inverse kinematics problem. A first approach to utilize redundancy to solve the inverse kinematics was proposed by Liégeois [7]. In his approach, a gradient projection is used to devise a general solution expressed as:

$$\dot{\boldsymbol{q}} = \mathbf{J}^{\perp}\dot{\mathbf{x}} \; + (\mathbf{I} - \mathbf{J}^{\perp}\mathbf{J})\mathbf{z} \tag{2}$$

where $\mathbf{J}^{\perp}$ is a generalized inverse of $\mathbf{J}$. The pseudo-inverse solution in the first term of eq. (2) is known to minimize the two-norm of the joint velocity vector, whereas the second term is called the homogeneous or null-space solution. The later does not contribute to the EE motion yet determines the minimum-norm solution. However, it has been shown that rather than driving the robot away from singularities at very high demands in joint velocities, this solution sometimes leads the robot to singularities [8]. A weighted pseudo-inverse by the inertia matrix is used here. This allows dynamic consistency compared to traditional pseudo-inverse. It is given by:

$$\mathbf{J}_{\mathbf{D}}^{\perp}=\mathbf{D}^{-1}\mathbf{J}^{T}(\mathbf{J}\mathbf{D}^{-1}\mathbf{J}^{T})^{-1} \tag{3}$$

In order to include a secondary task criterion within a performance index $r(\boldsymbol{q})$, $\mathbf{z}$ is chosen to be

$$\mathbf{z} = \pm b\nabla r(\boldsymbol{q}) \tag{4}$$

where b is a positive real number and $\nabla r(\boldsymbol{q})$ the gradient of $r(\boldsymbol{q})$. A positive sign in eq. (4) indicates that the criterion is to be maximized. A negative sign indicates minimization. Including joint limits avoidance constraint through redundancy resolution might be performed by choosing $r(\boldsymbol{q}) = \frac{1}{2}(\boldsymbol{q} - \bar{\boldsymbol{q}})^{T}\mathbf{W}(\boldsymbol{q} - \bar{\boldsymbol{q}})$ with $\bar{\boldsymbol{q}}$ being chosen such that it ensures joint limits avoidance. For example,

$$\bar{\boldsymbol{q}} = \frac{1}{2}(\boldsymbol{q}_{\max} + \boldsymbol{q}_{\min}) \tag{5}$$

yields $$\mathbf{z} = -\mathbf{W}(\boldsymbol{q} - \bar{\boldsymbol{q}}), \tag{6}$$

where $\mathbf{W}$ is a positive weighting matrix to scale the magnitude of the manipulator response to joint displacement. A typical choice for this matrix is $\mathbf{W} = \mathrm{diag}\,(\boldsymbol{q}_{\max} - \boldsymbol{q}_{\min})$

2. 2. Dynamic modelling

The robot dynamic model is developed using a Lagrangian formalism*,* which includes actuators model. This model allows closed-form expression of joint rates and accelerations characterizing the motion resulting from joint torques. It can be expressed in continuous-time as:

$$\mathbf{D}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \mathbf{V}(\boldsymbol{q}, \dot{\boldsymbol{q}}\,)\dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q}) \; = \boldsymbol{\tau} \tag{7}$$

where $\boldsymbol{\tau}$ is the $n \times 1$ joint torques vector produced by the joint actuators, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ are vectors describing joint positions, rates and accelerations, $\mathbf{D}(\boldsymbol{q})$ is the $n \times n$ manipulator inertia matrix, $\mathbf{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the $n \times n$ matrix representing Coriolis and centrifugal wrenches, and $\mathbf{G}(\boldsymbol{q})$ is the $n \times 1$ vector representing gravity forces [12]. Now, following references [2, 3], the discrete-time dynamic model can be approximated as:

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & h_k \boldsymbol{I}_{n\times n} \\ \boldsymbol{O}_{n\times n} & \boldsymbol{I}_{n\times n} \end{bmatrix} \boldsymbol{x}_k - \begin{bmatrix} \frac{h_k^2}{2} \boldsymbol{I}_{n\times n} \\ h_k \boldsymbol{I}_{n\times n} \end{bmatrix} \left[\mathbf{D}^{-1}(\boldsymbol{x}_{1k}) \left[\mathbf{V}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) \boldsymbol{x}_{2k} + \mathbf{G}(\boldsymbol{x}_{1k}) \right] \right] + \begin{bmatrix} \frac{h_k^2}{2} \boldsymbol{I}_{n\times n} \\ h_k \boldsymbol{I}_{n\times n} \end{bmatrix} \mathbf{D}^{-1}(\boldsymbol{x}_{1k}) \boldsymbol{\tau}_k \tag{8}$$

where $\boldsymbol{x}_k=(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k)^T=(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})^T$ is the $2n$-dimensional robot state and $\boldsymbol{I}$ is the nxn identity matrix. Equation (8) is written for simplicity as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_{dk}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \tag{9}$$

2. 3. Constraints modeling

In addition to eqs. (2)-(9), the following constraints are considered:

- Boundary conditions: These concern the starting and final states $\boldsymbol{x}_s$ and $\boldsymbol{x}_T$

$$\boldsymbol{x}_0 = \boldsymbol{x}_S, \; \boldsymbol{x}_N = \boldsymbol{x}_T \tag{10}$$

- Redundancy resolution ensuring joint limits avoidance: This is achieved using eqs. (2) and (3)

$$\boldsymbol{x}_{2k} = \mathbf{J}^{\perp}_{\mathrm{D}k}(\boldsymbol{x}_{1k}) \dot{\mathbf{x}}_k + (\mathbf{I} - \mathbf{J}^{\perp}_{\mathrm{D}k}(\boldsymbol{x}_{1k}) \mathbf{J}_k(\boldsymbol{x}_{1k})) \mathbf{z}_k(\boldsymbol{x}_{1k})$$

(11)

with $$\mathbf{z}_k = -\mathbf{W}\,(\boldsymbol{x}_{1k} - \bar{\boldsymbol{x}}_{1k}), \; \mathbf{W} = \mathrm{diag}(\bar{\boldsymbol{x}}_1) = \mathrm{diag}(\frac{\boldsymbol{x}_{1\max} + \boldsymbol{x}_{1\min}}{2}) \tag{12}$$

where $\bar{\boldsymbol{x}}_{1k}$, $\boldsymbol{x}_{1\max}$ and $\boldsymbol{x}_{1\min}$ refer, respectively, to discrete values of joint angles $\bar{\boldsymbol{q}}$, $\boldsymbol{q}_{\max}$ and $\boldsymbol{q}_{\min}$

- Admissible domain of the sampling periods

$$h_{\min} \le h_k \le h_{\max}\;,\; k = 0,\ldots,N-1 \tag{13}$$

- Admissible domain of the actuator torques

$$\boldsymbol{\tau}_{\min} \le \boldsymbol{\tau}_k \le \boldsymbol{\tau}_{\max}\;,\; k = 0,\ldots,N-1 \tag{14}$$

- Imposed passage through intermediate EE poses: This consists of a set of two-dimensional point positions defined as:

$$\left\| \mathbf{p} - \mathbf{p}_l \right\| - \mathrm{T}_{\mathrm{PassTh}lp} = 0,\; l = 1,\ldots,L \tag{15}$$

where $\mathbf{p}$ is the current position vector of the EE, $\mathbf{p}_l$ is the l^{th} passage point and L is the number of imposed points and $\mathrm{T}_{\mathrm{PassTh}lp}$ is the passage tolerance. For the sake of simplicity, all equality and inequality constraints are written as $s_i(\boldsymbol{x}) = 0 \;\; i = 1,\ldots,\mathrm{I}$ and $\mathbf{g}_j(\boldsymbol{x}, \boldsymbol{\tau}) \le 0 \;\; j = 1,\ldots,\mathrm{J}$, regardless if they depend only on state, control inputs or both, where I and J denote the numbers of equality and inequality constraints.

3. NON LINEAR PROGRAMMING FORMULATION

Several criteria had been proposed, such as traveling time, consumed energy, obstacle avoidance and manipulability measure [2-6]. The most popular is traveling time for the obvious reason of production targets. However, the major drawback of this control criterion is its bang-bang character, producing non smooth trajectories exceeding joint speed and/or acceleration limits achievable by the actuators [4]. To deal with this problem, some authors introduced a virtual time augmenting the actual time in an effort to track the desired trajectory. In this paper, a multi-objective planning strategy is implemented.

The discrete-time constrained optimal control problem consists of finding the optimal ~~sequences $(\boldsymbol{\tau}_0, \boldsymbol{\tau}_2, ..., \boldsymbol{\tau}_{N-1})$ and $(h_0, h_2, ..., h_{N-1})$ 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 multi-objective performance index E_d including energy, traveling time, and a singularity avoidance function. It is expressed as:

$$\underset{\substack{\boldsymbol{\tau}_k \in \mathsf{C} \\ h_k \in \mathrm{H}}}{\mathrm{Min}} \left\{ E_d = \left[\sum_{k=0}^{N-1} [\boldsymbol{\tau}_k \boldsymbol{U} \boldsymbol{\tau}_k^T + \iota + \delta\, \varpi\, (\boldsymbol{x}_{1k})] h_k \right] \right\} \tag{16}$$

where $\mathsf{C}, \mathrm{H}, \boldsymbol{U}$, ι and δ are, respectively, the set of admissible torques, the set of admissible sampling periods, electric energy matrix weight, a positive scalar time weight and a weight factor for singularity avoidance. h_k is the EE traveling time between two successive discrete poses k and $k+1$, $k = 1, ..., N$, with the overall traveling time being $T = \sum_{k=1}^{N} h_k$, ϖ is a singularity avoidance function defined as:

$$\varpi\, (\boldsymbol{x}_{1k}) = \frac{1}{\sqrt{\det(\mathbf{J}(\boldsymbol{x}_{1k}) \mathbf{J}^T(\boldsymbol{x}_{1k}))}} \tag{17}$$

This optimization problem is performed subject to constraints (9)-(15).

3. 1. Augmented Lagrangian (AL)

The problem (20) is a multi-objective non-linear and non-convex optimal control problem. It is solved using an Augmented Lagrangian technique, which transforms the constrained problem into a non-constrained one. The degree of penalty for violating the constraints is regulated by penalty parameters. This method 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 [10, 11]. The AL function is written as:

$$L_{\boldsymbol{\mu}}(\boldsymbol{x}, \boldsymbol{\tau}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \boldsymbol{\sigma}) = E_d + \sum_{k=0}^{N-1} \boldsymbol{\lambda}_{k+1}^T \left[\boldsymbol{x}_{k+1} - \boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \nu_k, h_k) \right] + \sum_{k=0}^{N-1} h_k \left\{ \sum_{j=1}^{2} \left[\Phi_{\boldsymbol{\mu}_g}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j(\boldsymbol{x}_k, \boldsymbol{\tau}_k)) + \Psi_{\boldsymbol{\mu}_s}(\boldsymbol{\sigma}_k^j, \boldsymbol{s}_j(\boldsymbol{x}_k)) \right] \right\} \tag{18}$$

In eq. (18), the function $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$ is defined by the discrete state eq. (9) at the sampling point k, N is the total sampling number $\boldsymbol{\lambda} \in R^{6\mathrm{N}}$ designates the co-states obtained from the adjunct equations, $\boldsymbol{\sigma}$ and $\boldsymbol{\rho}$ are Lagrange multipliers with appropriate dimensions, associated to equality and inequality constraints and $\boldsymbol{\mu}_S$ and $\boldsymbol{\mu}_g$ are the corresponding penalty coefficients. The adopted penalty functions combine penalty and dual methods allowing relaxation of inequality constraints as soon as they are satisfied. These are given by:

$$\Psi_{\boldsymbol{\mu}_s}(\boldsymbol{a}, \boldsymbol{b}) = (\boldsymbol{a} + \frac{\boldsymbol{\mu}_s}{2} \boldsymbol{b})^T \boldsymbol{b}, \quad \Phi_{\boldsymbol{\mu}_g}(\boldsymbol{a}, \boldsymbol{b}) = \frac{1}{2\boldsymbol{\mu}_g} \left\{ \left\| \mathrm{Max}(0, \boldsymbol{a} + \boldsymbol{\mu}_g \boldsymbol{b}) \right\|^2 - \left\| \boldsymbol{a} \right\|^2 \right\} \tag{19}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ refer to Lagrange multipliers and left hand side of equality and inequality constraints. The Karush-Kuhn-Tucker first order optimality conditions [11] state that for a trajectory $(\boldsymbol{x}_0, \boldsymbol{\tau}_0, h_0, ..., \boldsymbol{x}_N, \boldsymbol{\tau}_{N-1}, h_{N\text{-}1})$ to be optimal solution to the problem, there must exists some positive Lagrange multipliers $(\boldsymbol{\lambda}_k, \boldsymbol{\rho}_k)$, unrestricted sign multipliers $\boldsymbol{\sigma}_k$ and positive penalty coefficients $\boldsymbol{\mu}_S$ and

$\boldsymbol{\mu}_g$ such that:

$$\frac{\partial L_\mu}{\partial \boldsymbol{x}_k}=0,\ \frac{\partial L_\mu}{\partial \boldsymbol{\tau}_k}=0,\ \frac{\partial L_\mu}{\partial h_k}=0,\ \frac{\partial L_\mu}{\partial \boldsymbol{\lambda}_k}=0,\ \frac{\partial L_\mu}{\partial \boldsymbol{\rho}_k}=0,\ \frac{\partial L_\mu}{\partial \boldsymbol{\sigma}_k}=0,$$

$$\boldsymbol{\sigma}_k^T s(\boldsymbol{x})=0,\ \boldsymbol{\rho}_k^T g(\boldsymbol{x},\boldsymbol{\tau},h)=0 \text{ and } g(\boldsymbol{x},\boldsymbol{\tau},h)\le 0 \tag{20}$$

The development of the above conditions enables one to derive the iterative formulas to solve the optimal control problem by updating control variables, Lagrange multipliers and penalty coefficients. However, in eq. (9), $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$ contains the inverse of the inertia matrix $\mathbf{D}^{-1}(\boldsymbol{x}_1)$ and Coriolis and centrifugal wrenches $\mathbf{V}(\boldsymbol{x}_1, \boldsymbol{x}_2)$. These might be very cumbersome to express. In developing the first order optimality conditions and computing the co-states $\boldsymbol{\lambda}_k$, an inverse of the mentioned inertia matrix and its derivatives with respect to state variables must be computed, resulting in huge calculations.

3. 2. Augmented Lagrangian with decoupling (ALD)

The computational difficulty mentioned beforehand is solved using a linear-decoupled formulation [13].
Theorem: Under the invertibility condition of the inertia matrix, the control law defined as

$$\boldsymbol{u}=\mathbf{D}(\boldsymbol{x}_1)\boldsymbol{v}+\mathbf{V}(\boldsymbol{x}_1,\boldsymbol{x}_2)\boldsymbol{x}_2+\mathbf{G}(\boldsymbol{x}_1) \tag{21}$$

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

$$\dot{\boldsymbol{x}}_2=\boldsymbol{v} \tag{22}$$

where $\boldsymbol{v}$ is an auxiliary input.
This can be demonstrated by substituting the proposed control law given by eq. (21) into the dynamic model eq. (7). One gets $\mathbf{D}(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2=\mathbf{D}(\boldsymbol{x}_1)\boldsymbol{v}$, since $\mathbf{D}(\boldsymbol{x}_1)$ is invertible, it follows that $\dot{\boldsymbol{x}}_2=\boldsymbol{v}$
This gives the robot a decoupled linear behavior approximated by the following discrete linear equation:

$$\boldsymbol{x}_{k+1}=\mathbf{F}_{dk}\boldsymbol{x}_k+\mathbf{B}_{dk}\boldsymbol{v}_k=\boldsymbol{f}_{d_k}^D(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k) \tag{23}$$

with
$$\mathbf{F}_{dk}=\begin{bmatrix}\boldsymbol{I}_{n\times n} & h_k\boldsymbol{I}_{n\times n}\\ \boldsymbol{O}_{n\times n} & \boldsymbol{I}_{n\times n}\end{bmatrix},\ \mathbf{B}_{dk}=\begin{bmatrix}\frac{h_k^2}{2}\boldsymbol{I}_{n\times n}\\ h_k\boldsymbol{I}_{n\times n}\end{bmatrix} \tag{24}$$

Notice that while this dramatically eases the calculation of the co-states. The non-linearity is transferred to the objective function. The decoupled problem consists then of finding the optimal sequences of sampling periods and accelerations $(h_0, h_2,..., h_{N-1})$, $(\boldsymbol{v}_0, \boldsymbol{v}_2,..., \boldsymbol{v}_{N-1})$, allowing the robot to move from an initial state $\boldsymbol{x}_0 = \boldsymbol{x}_S$ to a final state $\boldsymbol{x}_N = \boldsymbol{x}_T$, while minimizing the cost function E_d^D, expressed as:

$$\underset{\substack{v\in\mathbf{V}\\ h_k\in\mathbf{H}}}{\mathrm{Min}}\left\{E_d^D=\sum_{k=0}^{N-1}\left[\ \left[\mathbf{D}(\boldsymbol{x}_{1k})\boldsymbol{v}_k+\mathbf{V}(\boldsymbol{x}_{1k},\boldsymbol{x}_{2k})\boldsymbol{x}_{2k}+\mathbf{G}(\boldsymbol{x}_{1k})\right]^T\boldsymbol{U}\left[\mathbf{D}(\boldsymbol{x}_{1k})\boldsymbol{v}_k+\mathbf{V}(\boldsymbol{x}_{1k},\boldsymbol{x}_{2k})\boldsymbol{x}_{2k}+\mathbf{G}(\boldsymbol{x}_{1k})\right] + \iota+\delta\varpi(\boldsymbol{x}_{1k})\ \right]h_k\right\} \tag{25}$$

Subject to the decoupled dynamic state eq. (22), and the above-mentioned constraints.
The augmented Lagrangian function (ALD) associated to the decoupled problem (25) is

$$L_{\mu}^{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{v}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \sigma) = E_{d}^{D} + \sum_{k=0}^{N-1} \boldsymbol{\lambda}_{k+1}^{T} \left[\ \boldsymbol{x}_{k+1} - \boldsymbol{f}_{d_k}^{\boldsymbol{D}}(\boldsymbol{x}_k, v_k, h_k) \ \right]$$

$$+ \sum_{k=0}^{N-1} h_k \left\{ \sum_{j=1}^{2} \left[\Phi_{\boldsymbol{\mu}_g}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j(\boldsymbol{x}_k, \boldsymbol{\tau}_k)) + \Psi_{\boldsymbol{\mu}_s}(\boldsymbol{\sigma}_k^j, \boldsymbol{s}_j(\boldsymbol{x}_k)) \right] \right\} \tag{26}$$

with the function $\boldsymbol{f}_{d_k}^{D}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k)$ being defined by the state of eq (23), and other parameters appearing in eq. (26) are defined above. Again, the development of the first order optimality conditions allows deriving iterative formulas for control and state variables, Lagrange multipliers as well as penalty coefficients.

These expressions are quite long and are not detailed here. The final state constraint $\boldsymbol{x}_N = \boldsymbol{x}_T$ is not included within the penalty procedure, as we have to satisfy it at each iteration. A re-adjustment is performed with an orthogonal projection on the tangent space of this constraint, through the application of the descent direction

$$\boldsymbol{d} \ = - \mathbf{P}_v \nabla_v L_{\mu}^{D} \tag{27}$$

with $\mathbf{P}_v$ being defined as

$$\mathbf{P}_v = \mathbf{I}_d - \mathbf{Q}_v^T (\mathbf{Q}_v \mathbf{Q}_v^T)^{-1} \mathbf{Q}_v \tag{28}$$

where $\mathbf{I}_d$ is an identity matrix with appropriate dimension and $\mathbf{Q}_v$ is the projection matrix on the tangent space of the final state constraint. The re-adjustment process allows satisfying target attainability with any given ε- precision [11]. If other imposed states are to be satisfied at each iteration, then the re-adjustment procedure must be extended to these constraints. Figure 1 depicts a flowchart diagram of ALD function and architecture for the multi-objective trajectory planning problem. In this procedure, one has to select robot parameters, task definition, (such as starting, intermediate and target positions), workspace limitations and simulation parameters (block 1). Then a kinematic unit (block 2) defines a feasible initial solution satisfying boundary constraints in joint angles, velocities, accelerations and jerks. This solution is defined through a cycloidal profile. This profile has been chosen as it allows a near-minimum time smooth continuous trajectory as compared to a trapezoidal profile [14]. Then an inner optimization loop (block 3) solves for the AL minimization with respect to sampling periods and acceleration control variables. One first computes the gradients of the Lagrangian, the co-states backwardly and the projection matrix and operator. Then a steepest descent is calculated and tested against a suitable tolerance (block 4). If non-satisfied, one computes new search directions and updates sampling time and acceleration inputs. Then go back to inner optimization loop to update gradients and direction descent. When satisfied, one goes further to test other equality and inequality constraints against feasibility tolerances (block 5). If non feasible, go back to the inner optimization unit. Else, if feasible, do a convergence test (block 6) for cost minimization and constraints satisfaction against optimal tolerances. If convergence holds, display optimal trajectory and end the program. Otherwise, go further to the dual part of AL (block7) to test for constraints satisfaction and update penalty and tolerance parameters. If the constraints are not violated with respect to first order optimal tolerances then the multipliers are updated without decreasing penalty. If they are violated, decrease penalty while keeping unchanged Lagrange multipliers, 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 for the primal problem.

4. SIMULATION RESULTS

A three revolute (n=3-DOF) serial manipulator moving on the vertical plan with a 2-DOF task is considered (Fig. 2). The robot kinematic and dynamic parameters are given in Tables 1 and 2. The simulation objectives are to: 1) minimize travelling time and instantaneous energy during the motion; 2) resolve the redundancy and avoid singularities; 3) satisfy several constraints related to limits of joint angles, rates, accelerations and torques. The first example is to move the tip of the manipulator along a straight line from the starting position (1.95, 0.82) (m), corresponding to joint angles (0^o, 30^o, 30^o), to the final position (1.35, 1.3) (m), without considering the orientation. Hence, this task is performed with a serial 3-DOF planar manipulator that is redundant with respect to the given task. The joint velocities are zero at the starting and ending positions and the EE travels through a distance of 0.75 m. The multi-objective trajectory is performed by taking each weight equals to unity in the performance index. The initial pose values are chosen to satisfy a secondary goal which consists of avoiding joint limits. Figure 3 shows the (x, y) position variations for the cycloidal minimum-time trajectory and those generated based on ALD approach including robot dynamics, kinematics and constraints. Although the cycloidal minimum-time trajectory is a straight line, it is slightly disturbed in the case of its ALD counterpart. The bias between the two paths is due to the non-linearity of the dynamic model, considering for example the gravity effects. Figure 4 displays the corresponding joint angle variations. Figure 5 shows the associated instantaneous variations of the consumed electrical and kinetic energy and sampling periods. One notices the significant and monotonous diminishing of the consumed energy with the ALD as compared to the cycloidal one. As for traveling time, after 4 outer and 4 inner ALD iterations, one notices a dramatic reduction compared to the initial cycloidal trajectory, with $t_{\text{Cycloidal}} = 2\,\text{sec}$ and $t_{\text{Multi-objective}} = 1.237\,\text{sec}$, an increase of 45% for the initial solution. On the other hand, although one gets a feasible path with the cycloidal profile, the associated torques and necessary energy, computed from the robot inverse dynamics are fairly high and exceed the nominal values. The same trajectory has been performed with three imposed passage points for the robot EE. Figure 6 shows the (x, y) coordinate variations while satisfying passage through imposed points (1.7, 0.9), (1.5, 1.0), (1.4, 1.2) (m), (in this Figure, the cycloidal trajectory does not consider this constraint). Figure 7 shows consumed energy and traveling time for the trajectory with passages. The second trajectory is a second straight line, starting from the joint position (0^0, 15^0, 20^0) corresponding to the Cartesian coordinates (0.2, 0.9) (m) to the ending position (0.64, 0.45) (m). The consumed energy and sampling period variations are shown in Fig. 8. The third trajectory is a circle with a center (-0.045, 0.75) (m) and radius 0.14 (m). The joint angle variations are shown in Fig 9, and the consumed energy and traveling time are shown in Fig. 10. In order to assess the sensitivity of the proposed multi-objective trajectory planning to dynamic parameter changes, the mass of the third link was increased by 1.5 Kg. Figs. 11 (a) and (b) show respectively, the variations of sampling time and consumed energy for the original and modified third link mass trajectories. Although the necessary energy and traveling time to perform the modified link mass task grow greater, the achieved performance is good, as it reaches the position with a precision of order 10^{-3} (m). This highlights the ALD good robustness to parameters changes as illustrated in Fig 12.

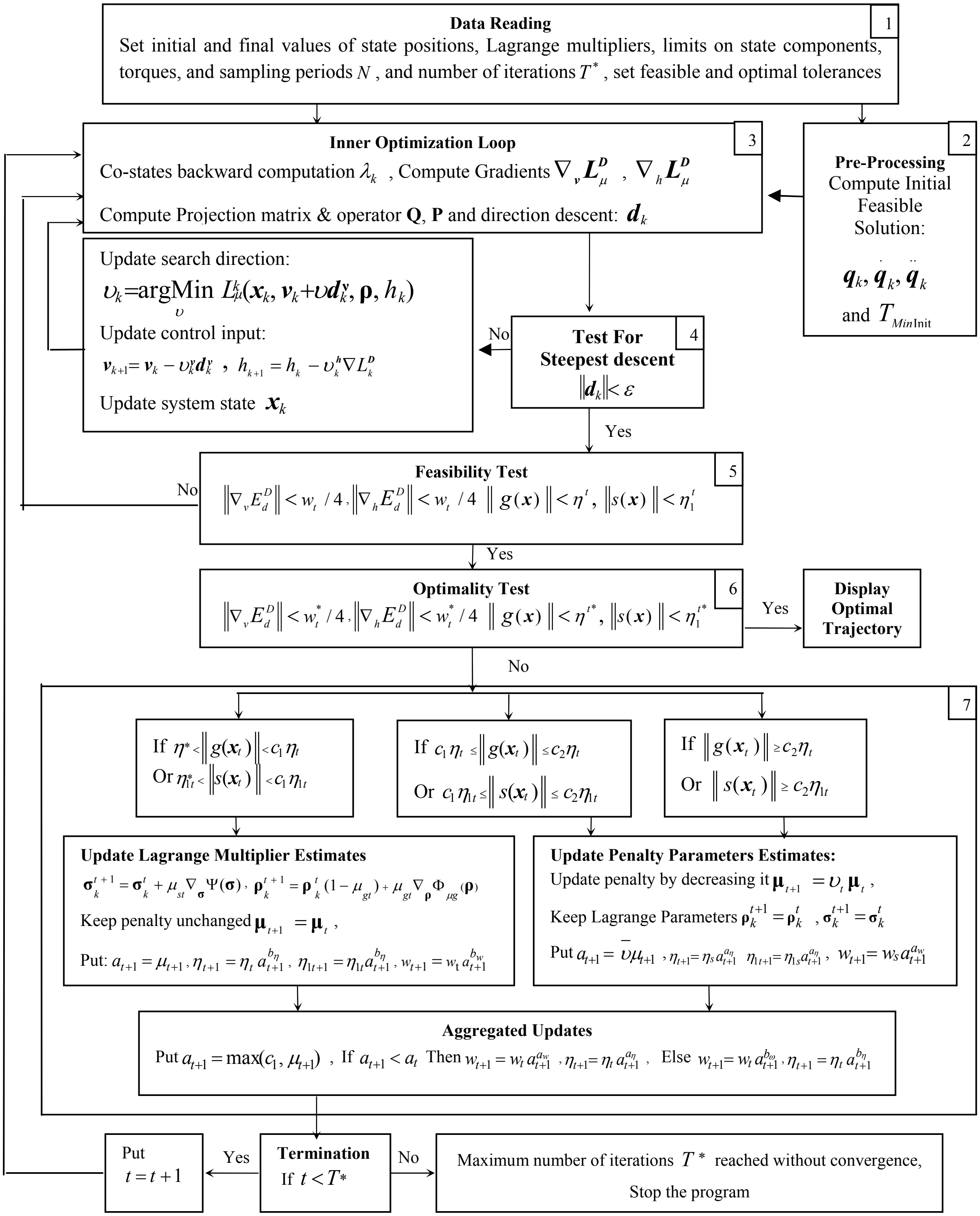
On the whole, the computation time is quite long. It took about 9 minutes on a Pentium III, 996 MHz, to simulate and get the performances of the first straight line trajectory. This is due mainly to the high nonlinear robot dynamics and projection matrix and operator calculations to satisfy the final state constraint at each iteration.

Table 1. Kinematic and dynamic parameters of the manipulator

Link I	Mass (kg)	Inertia I_i(kg m^2)	Length L_i(m)
1	8	9.23	1.0
2	4	7.5	0.8
3	2	5.21	0.45

Table 2. Limits of workspace, actuator torques, joint rates and accelerations and sampling periods

Parameter	x (m)	y (m)	τ_1 (Nm)	τ_2 (Nm)	τ_3 (Nm)	$\dot{q}_{i\text{Motor}}$ (rad/s)	$\ddot{q}_{i\text{Motor}}$ (rad/s^2)	h (sec)
Max	2.5	2.5	40	25	20	2	3	0.1
Min	0.4	0.20	-40	-25	-20	-2	-3	0.001



1
Data Reading
Set initial and final values of state positions, Lagrange multipliers, limits on state components, torques, and sampling periods N, and number of iterations T^*, set feasible and optimal tolerances
2
Pre-Processing
Compute Initial Feasible Solution:
$q_k, \dot{q}_k, \ddot{q}_k$
and $T_{Min\text{Init}}$
3
Inner Optimization Loop
Co-states backward computation λ_k, Compute Gradients $\nabla_v L^D_\mu$, $\nabla_h L^D_\mu$
Compute Projection matrix & operator **Q**, **P** and direction descent: d_k
Update search direction:
$\upsilon_k = \underset{\upsilon}{\operatorname{argMin}}\, L^k_\mu(x_k, v_k + \upsilon d^v_k, \boldsymbol{\rho}, h_k)$
Update control input:
$v_{k+1} = v_k - \upsilon^v_k d^v_k$, $h_{k+1} = h_k - \upsilon^h_k \nabla L^D_k$
Update system state x_k
No
4
Test For Steepest descent
$\|d_k\| < \varepsilon$
Yes
5
Feasibility Test
$\|\nabla_v E^D_d\| < w_t/4$, $\|\nabla_h E^D_d\| < w_t/4$ $\|g(x)\| < \eta^t$, $\|s(x)\| < \eta^t_1$
No
Yes
6
Optimality Test
$\|\nabla_v E^D_d\| < w^*_t/4$, $\|\nabla_h E^D_d\| < w^*_t/4$ $\|g(x)\| < \eta^{t^*}$, $\|s(x)\| < \eta^{t^*}_1$
Yes
Display Optimal Trajectory
No
7
If $\eta^* < \|g(x_t)\| < c_1\eta_t$
Or $\eta^*_{1t} < \|s(x_t)\| < c_1\eta_{1t}$
If $c_1\eta_t \le \|g(x_t)\| \le c_2\eta_t$
Or $c_1\eta_{1t} \le \|s(x_t)\| \le c_2\eta_{1t}$
If $\|g(x_t)\| \ge c_2\eta_t$
Or $\|s(x_t)\| \ge c_2\eta_{1t}$
Update Lagrange Multiplier Estimates
$\boldsymbol{\sigma}^{t+1}_k = \boldsymbol{\sigma}^t_k + \mu_{st}\nabla_{\boldsymbol{\sigma}}\Psi(\boldsymbol{\sigma})$, $\boldsymbol{\rho}^{t+1}_k = \boldsymbol{\rho}^t_k(1-\mu_{gt}) + \mu_{gt}\nabla_{\boldsymbol{\rho}}\Phi_{\mu g}(\boldsymbol{\rho})$
Keep penalty unchanged $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t$,
Put: $a_{t+1} = \mu_{t+1}$, $\eta_{t+1} = \eta_t a^{b_\eta}_{t+1}$, $\eta_{1t+1} = \eta_{1t} a^{b_\eta}_{t+1}$, $w_{t+1} = w_t a^{b_w}_{t+1}$
Update Penalty Parameters Estimates:
Update penalty by decreasing it $\boldsymbol{\mu}_{t+1} = \upsilon_t \boldsymbol{\mu}_t$,
Keep Lagrange Parameters $\boldsymbol{\rho}^{t+1}_k = \boldsymbol{\rho}^t_k$, $\boldsymbol{\sigma}^{t+1}_k = \boldsymbol{\sigma}^t_k$
Put $a_{t+1} = \bar{\upsilon}\mu_{t+1}$, $\eta_{t+1} = \eta_s a^{a_\eta}_{t+1}$ $\eta_{1t+1} = \eta_{1s} a^{a_\eta}_{t+1}$, $w_{t+1} = w_s a^{a_w}_{t+1}$
Aggregated Updates
Put $a_{t+1} = \max(c_1, \mu_{t+1})$, If $a_{t+1} < a_t$ Then $w_{t+1} = w_t a^{a_w}_{t+1}$, $\eta_{t+1} = \eta_t a^{a_\eta}_{t+1}$, Else $w_{t+1} = w_t a^{b_\omega}_{t+1}$, $\eta_{t+1} = \eta_t a^{b_\eta}_{t+1}$
Put $t = t+1$
Yes
Termination If $t < T^*$
No
Maximum number of iterations T^* reached without convergence, Stop the program

Fig. 1 Function and architecture of the ALD system for the multi-objective trajectory-planning problem

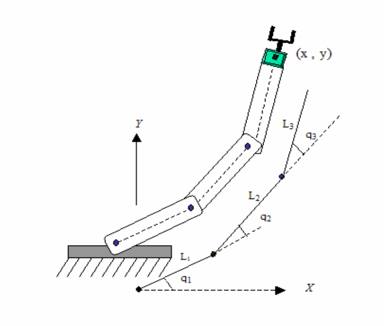


Fig. 2. Geometry of a 3-DOF planar manipulator of joint angles $q_j,\ j=1,2,3$

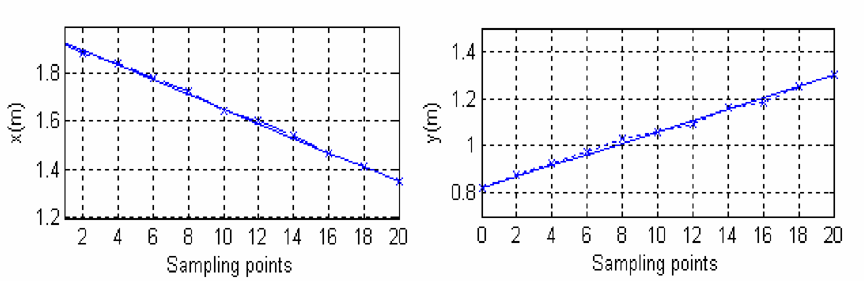


Fig. 3. EE trajectory along axis x and y displacement in meters versus sampling points,

(___) Cycloidal profile, (*-*-*-*) Minimum-time including robot dynamics and constraints

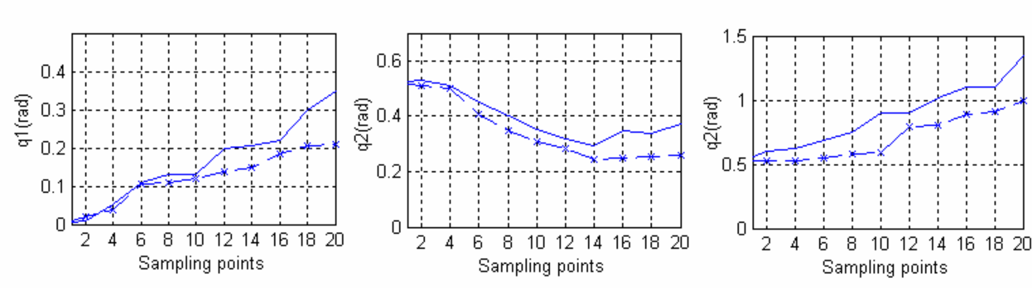


Fig. 4. Joint q_i variations

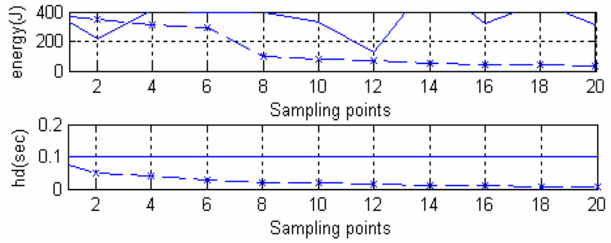


Fig. 5. Corresponding energy and sampling period h variations

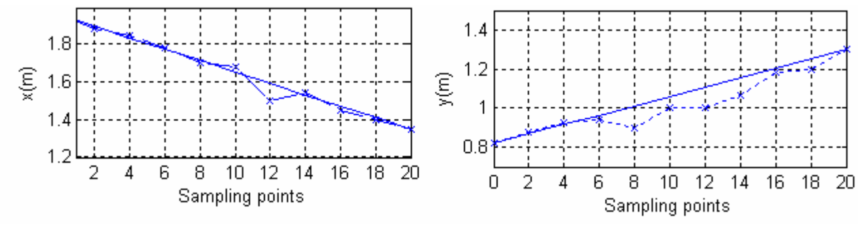


Figure 6 (x, y) coordinate variations while satisfying the passage through imposed points (1.7, 0.9), (1.5, 1.0), (1.4, 1.2) (m)

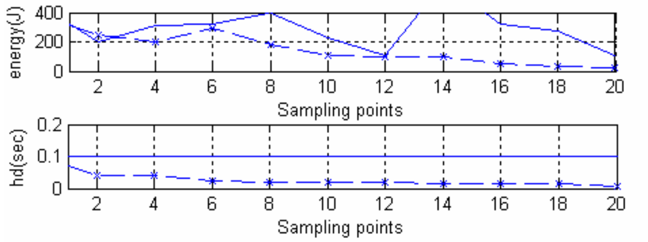


Figure 7 Associated consumed energy and traveling time to the trajectory with imposed passages

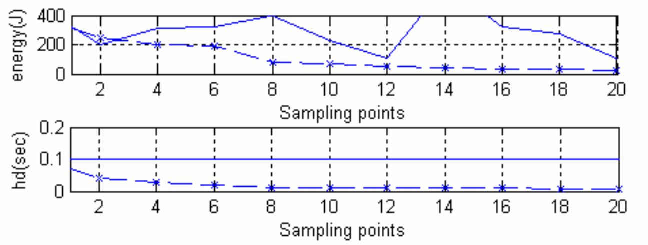
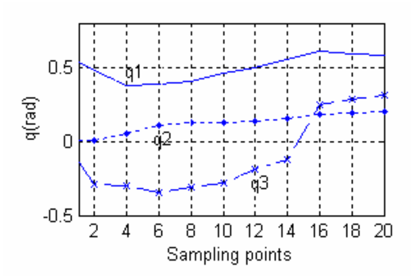


Fig. 8. Energy and sampling period *h* instantaneous variations for the second trajectory



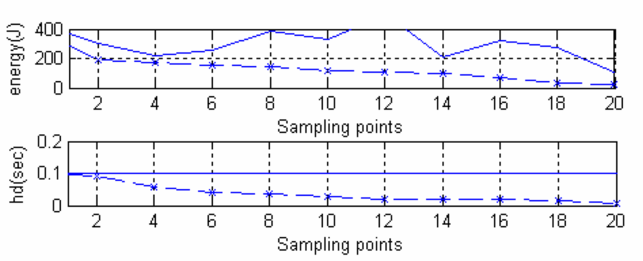
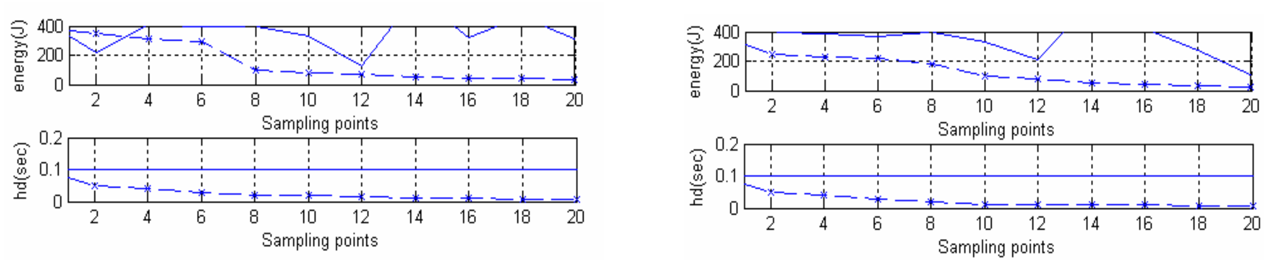


Fig.9. Instantaneous variations of joint angles q_j, $j = 1,2,3$ for the circle trajectory

Fig. 10. Instantaneous variations of consumed energy and sampling time for the circle trajectory



(a)

(b)

Fig.11. Instantaneous variations of consumed energy and sampling time for the first example
(a) Trajectory with third link mass 2 Kg (b) Trajectory with increased third link mass by 1.5 Kg

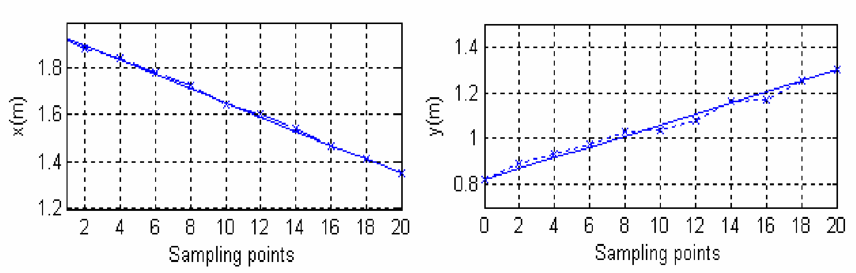


Fig.12. EE trajectory along axis *x* and *y* displacement for example 1 with modified third link mass

5. CONCLUSION

In this paper, a method for computing a multi-objective trajectory planning is developed. This method includes joint angles, rates, accelerations, jerks, workspace, and actuator torques limitations. It is solved through an Augmented Lagrangian on a decoupled form of the robot dynamics. According to preliminary simulation results, this approach is effective and robust in solving the non-convex and non-linear constrained motion planning problem. The trajectories are smooth and singularity free, a capability which makes them very suitable for use as reference inputs to a feedback like PID position controller or as training datasets on which to build an objective data-driven neuro-fuzzy system for on-line planning and control. An ongoing work is on reducing the computational time by accelerating the convergence rate of the algorithm. Another short trend is to use the outcomes of such a multi-objective trajectory planning to build a data-driven neuro-fuzzy control system for on-line planning with reasonable computational time.

ACKNOWLEDGMENTS

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

[1] C. Ahrikencheikh and A. Seireg: "*Optimized-Motion Planning*", John Wiley & Sons, 1994.
[2] A. Khoukhi: "An Optimal Time-Energy Control Design for a Prototype Educational Robot ", *Robotica* v. 20, p 661-671, 2002.
[3] A. Khoukhi, L. Baron, M. Balazinski: "A Decoupled Approach to Time-Energy Trajectory Planning of Parallel Kinematic Machines", *Proc. of 16th CISM-IFFToMM Symposium, Robot Design, Dynamics and Control, ROMANSY'2006,* p 179-186, Warsaw, Poland, June 20-24, 2006.
[4] H. Jurgen, Mc. John: "Determination of Minimum-Time Maneuvers for a Robotic Manipulator Using Numerical Optimization Methods", *Mech. of Struct. and Mach.*, 27, (2), p 185-201, 1999.
[5] O. Dahl: "Path-Constrained Robot Control with Limited Torques-Experimental Evaluation", *IEEE Trans. Robot. Automat.*, v. 10, p 658-669, Oct.1994.
[6] Y. Nakamura: "*Advanced Robotics: Redundancy and Optimization*", Addison-Wesley, 1991.
[7] A. Liegeois: "Automatic Supervisory Control of Configurations and Behavior of Multibody Mechanisms", *IEEE Trans. Syst., Man, Cybern.*, v SMC-7, p 868–871, Dec. 1977.
[8] C. R. Carignan: "Trajectory Optimization for Kinematically Redundant Arms", *Journal of Roboric Systems*, 8(2), p 221 - 248, 1991.
[9] S. Chiaverini: "Singularity-Robust Task-Priority Redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators", *IEEE Trans. On Robotics & Automation,* 13, (3), p 398 – 410, 1997.
[10] T. Rockafellar: "Lagrange Multipliers and optimality", *SIAM Review* 35, p 183-238, 1993.
[11] E. Polak: "*Optimization, Algorithms and Consistent Approximation*", Springer, N.Y, 1997.
[12] J. J. Craig: "*Introduction to Robotics Mechanics and Control*", Pearson Prentice Hall, Pearson Education, Inc., Upper Saddle River, NJ 07458, 2005.
[13] A. Isidori: "*Non linear control systems*", Springer; 3rd Edition, London, UK, 1995.
[14] A. Khoukhi, K. Demirli, L. Baron and M. Balazinski: "Hierarchical Neuro-Fuzzy Near-Minimum Time Trajectory Planning of a Serial Planar Robot", in revision, *Engineering Applications of Artificial Intelligence,* 2007.