PARAMETRIC OPTIMIZATION FOR OPTIMAL SYNTHESIS
of robotic systems' motions

Taha Chettibi, Moussa Haddad
Laboratory of Structure Mechanics

Samir Hanchi
Laboratory of Fluid Mechanics,
E.M.P., B.E.B., BP 17, 16111, Algiers, Algeria.

Keywords: Robotic Systems, Parametric Optimization, Motion Synthesis.

Abstract: This paper presents how a problem of optimal trajectory planning, that is an optimal control problem, can be transformed into a parametric optimization problem and in consequence be tackled using efficient deterministic or stochastic parametric optimization techniques. The transformation is done thanks to discretizing some or all continuous system's variables and forming their time-histories by interpolation. We will discuss three different methods where, in addition to transfer time *T,* considered optimization parameters are: 1) independent state and control parameters, 2) control parameters and 3) independent position parameters. The simplicity and the efficiency of the third mode allow us to use it to solve the problem of optimal trajectory planning in complex situations, in particular for holonomic and non-holonomic systems.

1 INTRODUCTION

The problem of optimal motion synthesis for robotic systems is a fundamental issue in robotics. It is generally stated as an optimal control problem OCP. Due to its strategic importance, the treatment of this problem received a great attention from researchers and was the subject of many papers. In fact, we find a large diversity of proposed techniques that can be classified into two main families, namely direct and indirect methods (Stryk, 1993)(Steinbach, 1995)(Betts, 1999). The indirect methods are based on the calculus of variation and lead generally to a multipoint boundary value problem BVP. However, such techniques suffer from many drawbacks:

- User must have knowledge of OCP theory in order to be able to compute all elements of involved solution program (particularly the Hamiltonien and its gradients). Further more, even if the user has the requisite theoretical background, constructing these expressions for complicated applications might be very difficult (Betts, 1999).
- The approach is not flexible because each new problem requires a new derivation of relevant elements.
- If the problem description includes path inequalities, the user must estimate a priori the constrained arc sequence. This tends to be quite difficult and makes the definition of arc boundaries extremely difficult (Bryson, 1999).
- One main difficulty of implementation is that the user must guess values of the adjoint variables (co-states) that is not an intuitive task because they are not significant physical quantities (Chettibi et al, 2004 *a*, *b*).
- Singular arcs, where switching functions are nulls (Geering et al, 1986), involve a particular treatment for example: by introducing a perturbed energy term in the performance index (Chen and desrochers, 1988, 1990, Chen et al 1993).
- This approach suffer also from proper deficiencies of applied numerical methods (shooting and finite difference methods) used for the treatment of resulting BVP (instability, need of accurate initial guess, …) (Ascher and Petzold, 1998).

Chettibi T., Haddad M. and Hanchi S. (2005).
PARAMETRIC OPTIMIZATION FOR OPTIMAL SYNTHESIS - of robotic systems' motions.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics - Robotics and Automation*, pages 3-10
DOI: 10.5220/0001160700030010

To overcome the indirect method's lacks, direct methods have been introduced. They are based on the conversion of the original OCP into a Parametric Optimization Problem POP by the parameterization of the class of some system variables time functions. So, this kind of methods restricts the attention to some parameterized family of possible trajectories, thus reducing the original infinite-dimensional problem to a more tractable finitely parameterized optimization problem. This is, of course, done at the expense that the solution will be optimal in the selected class, i.e., suboptimal with respect to the original problem. Nevertheless, this drawback must be weighted against the following practical advantages (Bryson, 1999)(Chettibi et al, 2004 *a*, *b*):

- Do not require any additional analytical derivations.
- Systematic treatment of path constraints and inequality constraints.
- The number and sequence of constrained arcs do not have to be guessed
- Singular arcs are handled without any special coding; their number and location do not have to be estimated.
- Small space of search.

In what follows, three classes of methods will be discussed. The unknowns in each class, in addition to transfer time T, are:

Class 1: independent state parameters and control parameters,
Class 2: Control parameters,
Class 3: Independent position parameters.

Of course, according to the adopted conversion method, different numerical integration techniques will be employed and the amount of calculus effort will differ. In fact, classes 1 and 2 are commonly used to propose an approximated solution of the original OCP (Stryk and Bulirsch, 1993) (Stryk, 1993) (Steinbach, 1995) (Betts, 1999). In contrast, the third class is rarely employed (Chettibi et al, 2004(a), (b)) (Bobrow et al, 2001). One of the objectives of the present paper is to illustrate the efficiency and simplicity of this class and its ability to handle complex problems arising for example in holonomic and non-holonomic robotic systems.

Once the trajectory parameterization is performed, the original problem becomes a nonlinear parametric optimization problem that can be treated using efficient deterministic or stochastic parametric optimization techniques.

2 DYNAMIC MODEL OF ROBOTIC SYSTEMS

In order to synthesis optimal motions for robotic systems, a complete dynamic model is needed. Consider a robot or in general a constrained mechanical system composed of p unconstrained systems, each described by n_i coordinates q_i with a Lagrangian $L_i(q_i, \dot{q}_i)=T_i-U_i$, where

$$\boldsymbol{T}_i = \frac{1}{2}\dot{\boldsymbol{q}}_i^T \boldsymbol{M}_i(\boldsymbol{q}_i)\dot{\boldsymbol{q}}_i$$

is the kinetic energy and U_i is the potential energy for the i^{th} system (Angels, 1997). Let the p systems be connected through m_h holonomic constraints described by $C(q)=0$ (for e.g. closure condition for parallel robots), and m_n nonholonomic constraints described by $N(q,\dot{q})=0$ (for e.g. condition of pure rolling and non-slipping for mobile robots). We assume that all $m = m_h + m_n$ constraints are time-invariant (scleronomic) and that they can be jointly written in a Pfaffian form as follows (Bicchi et al, 2001):

$$\begin{bmatrix} \boldsymbol{A}_h(\boldsymbol{q}) \\ \boldsymbol{A}_n(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} = \boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = 0$$

With $\boldsymbol{A}_h \in \Re^{m_h} \times \Re^{n}$, $\boldsymbol{A}_n \in \Re^{m_n} \times \Re^{n}$, $\boldsymbol{q} \in \Re^{n}$.

A is supposed of full rank unless constraints are redundant and the system is said to be hyperstatic.

Equations describing the system dynamics are thus obtained as:

$$\boldsymbol{M}\,\ddot{\boldsymbol{q}} + \boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{f}_{ext}, \tau_{ext}) + \boldsymbol{A}^T \lambda = \tau \qquad (1a)$$

$$\boldsymbol{C}(\boldsymbol{q}) = 0 \qquad (1b)$$

$$\boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = 0 \qquad (1c)$$

Where $\boldsymbol{M} = \boldsymbol{diag}\lfloor \boldsymbol{M}_1, \boldsymbol{M}_2, ..., \boldsymbol{M}_p \rfloor$ is the mass matrix of the mechanical system, $\boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{f}_{ext}, \tau_{ext})$ is the non-linear dynamical vector which contains the gyroscopic, centrifugal and gravity terms as well as any others non conservative forces as external forces f_{ext} and torques τ_{ext}. τ stands for the generalized joint forces. The unknown Lagrangian multipliers vector $\lambda \in \Re^m$ can be interpreted as a reaction force capable of enforcing the constraints. Note that H can include discontinuous terms like those due to dry friction efforts. In that case, the system (1) is no longer differentiable and must be treated with precautions from numerical point of view.

The system (1) is a mixed set of differential and algebraic equations where q are differential variables describing the system's state and λ are algebraic variables. Relation (1) is defined as a set of Differential Algebraic Equations (DAE) of index 3 (Ascher and Petzold, 1998). If the mechanical system is not under any kind of nonholonomic or holonomic constraints, (1) becomes a simple system of ordinary differential equations ODE. It is the case for open chain robots with tree-like topology (Dombre and Khalil, 1999).

Solving (1) for τ is known as the Inverse Dynamic Model (IDM), while for q is the Forward Dynamic Model (FDM). The two problems are of quite different complexity. In fact, the former problem is a relative straightforward algebraic operation based on the substitution of $q(t)$ and its time derivatives in (1). In contrast, the later problem involves the integration of a DAE system using an implicit numerical method such as backward differentiation formulae (BDF) or implicit Runge-Kutta (IRK) methods. However, this approach leads generally to ill-conditioned problems. To handle this problem, reducing the index of the DAE system by differentiating constraints has been proposed (Ascher and Petzold, 1998).

Relation (1) can be transformed as follows:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{A}^t \\ \boldsymbol{A} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \tau - \boldsymbol{H} \\ -\dot{\boldsymbol{A}}\dot{\boldsymbol{q}} \end{bmatrix} \tag{2}$$

Although (2) is mathematically equivalent to (1) but its numerical behaviour is better (Bicchi et al, 2001). Under the condition of non-singularity of the matrix on the left hand side of (2) we can write:

$$\begin{bmatrix} \ddot{\boldsymbol{q}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \boldsymbol{M} & \boldsymbol{A}^t \\ \boldsymbol{A} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \tau - \boldsymbol{H} \\ -\dot{\boldsymbol{A}}\dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_a(\boldsymbol{q}, \dot{\boldsymbol{q}}, \tau) \\ \boldsymbol{f}_\lambda(\boldsymbol{q}, \dot{\boldsymbol{q}}, \tau) \end{bmatrix} \tag{3}$$

By introducing the state vector $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix} \in \Re^{2n}$ and the control vector $\boldsymbol{u} = \tau$, relation (3) becomes:

$$\begin{bmatrix} \dot{\boldsymbol{X}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{f}_a(\boldsymbol{q}, \dot{\boldsymbol{q}}, \tau) \\ \boldsymbol{f}_\lambda(\boldsymbol{q}, \dot{\boldsymbol{q}}, \tau) \end{bmatrix} \tag{4}$$

Note that any explicit dependence on t is omitted for notational simplicity, but of course all the quantities above are function of times. In addition, non-autonomous problems can be transformed into the above form (4) by defining an additional differential variable X_{2n+1} satisfying the initial value problem (IVP) $\dot{\boldsymbol{X}}_{2n+1}(t) = 1$ with $\boldsymbol{X}_{2n+1}(0) = 0$.

3 CONSTRAINTS

Any feasible motion of the robot must satisfy at any moment, in addition to relation (4), others constraints reflecting the limitations of the robot's capacities and the nature of both assigned task and the environment. In fact, if obstacles are present in the robot workspace, collisions must be avoided. Therefore, the following constraint holds during any transfer:

$$C(q(t)) = False \tag{5a}$$

Here, C denotes a Boolean function that indicates whether the robot at configuration q is in collision either with an obstacle or with itself.

Furthermore, when the robot is asked to move along a prescribed geometric path, this can be represented by the six-dimensional vector $\boldsymbol{R} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \psi, \theta, \varphi)$ ($(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ for the position and (ψ, θ, φ) for the orientation of the end effector relative to an inertial frame). The vector R is a known function of the distance along the path, $s(t)$, and may be expressed in terms of coordinates $q(t)$, using the forward kinematic model (Angels 1997) :

$$\boldsymbol{R}(s(t)) = \boldsymbol{P}(\boldsymbol{q}(t)) \tag{5b}$$

This is an equality constraint that must be hold during the whole transfer.

In addition, we have generally box constraints on the following physical quantities:

- *position:* $\boldsymbol{q}^{\min} \le \boldsymbol{q}(t) \le \boldsymbol{q}^{\max}$ (5*c*);
- *velocity:* $\dot{\boldsymbol{q}}^{\min} \le \dot{\boldsymbol{q}}(t) \le \dot{\boldsymbol{q}}^{\max}$ (5*d*);
- *acceleration:* $\ddot{\boldsymbol{q}}^{\min} \le \ddot{\boldsymbol{q}}(t) \le \ddot{\boldsymbol{q}}^{\max}$ (5*e*);
- *jerk:* $\dddot{\boldsymbol{q}}^{\min} \le \dddot{\boldsymbol{q}}(t) \le \dddot{\boldsymbol{q}}^{\max}$ (5*f*);
- *control:* $\tau^{\min} \le \tau(t) \le \tau^{\max}$ (5*g*);

The above mentioned constraints (5*a*,…,5*g*) constitute a set of path constraints and can be written in the following abstract form:

$$\boldsymbol{g}(t, \boldsymbol{X}(t), \dot{\boldsymbol{X}}(t), \tau(t)) \le 0 \tag{6}$$

Therefore, the set of feasible motions for any robotic system is limited by a large number of geometric, kinematic and dynamic constraints. The search for optimal trajectories in such a set is a quite hard task and involves adequate strategies able to tackle simultaneously all these constraints.

4 BASIC FORMULATION OF A TRAJECTORY OPTIMIZATION PROBLEM

It can be stated as follows:

find a state function $X(t)$ and a control $\tau(t)$ on time interval $[0, T]$ such that a scalar performance criterion

$$\boldsymbol{J} = \phi\left(\boldsymbol{T}, \boldsymbol{X}(\boldsymbol{T})\right) + \int_0^T \boldsymbol{L}\left(\boldsymbol{t}, \boldsymbol{X}(\boldsymbol{t}), \tau(\boldsymbol{t})\right)\boldsymbol{dt} \qquad (7)$$

is minimized, subject to path constraints (6), differential algebraic constraints (4), and the prescribed limit conditions

$$\boldsymbol{t} = 0, \quad \boldsymbol{X}(0) = \boldsymbol{X}_0 \qquad \boldsymbol{t} = \boldsymbol{T}, \quad \boldsymbol{X}(\boldsymbol{T}) = \boldsymbol{X}_f \qquad (8)$$

Relations (4), (6), (7) and (8) constitute a generic OCP. In this formulation, the final time is fixed. A problem with free final time can be transformed into this format by normalising the time and introducing a new variable X_{2n+1} satisfying the initial value problem (IVP) $\dot{X}_{2n+1}(t) = 1$ with $X_{2n+1}(0) = 0$ (the same action we proposed for non-autonomous problems).

J denotes the real valued objective criterion to be minimized. In general, *J* contains significant physical parameters related to the robot behaviour and also to the productivity of the robotic system. We propose here the following general expression that is a balance between transfer time *T* and quadratic average of actuator efforts *:*

$$\boldsymbol{J} = \mu \boldsymbol{T} + \frac{(1-\mu)}{2}\int_0^{\boldsymbol{T}} \sum_{i=1}^{n} \left(\frac{\tau_i}{\tau_i^{\max}}\right)^2 \boldsymbol{dt} \qquad (9)$$

μ is a weighting coefficient ($0 \le \mu \le 1$). The case: $\mu = 1$, corresponds to the minimum time problem.
The numerical treatment of above formulated OCP with conventional indirect methods seems to be abandoned in the favour of direct methods based on the conversion of an OCP into a POP.

5 CONVERSION OF THE TRAJECTORY OPTIMIZATION PROBLEM INTO A PARAMETRIC OPTIMIZATION PROBLEM

The conversion of the problem of trajectory optimization into a POP starts with the definition of *N* nodes (Knots), at fixed times, $0 = \boldsymbol{t}_1 < \boldsymbol{t}_2 < ... < \boldsymbol{t}_N = \boldsymbol{T}$, uniformly or not uniformly distributed along the time scale. Then, the system's kinematic or dynamic continuous variables are replaced by their values at the nodes ($\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \ddot{\boldsymbol{q}}_k$ or τ_k) and some form of interpolation (fig. 1). In addition to final time parameter *T*, parameters of the new parametric optimization problem can be chosen in various ways as a combination of $\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k, \ddot{\boldsymbol{q}}_k$ or τ_k. Along the optimization process, these parameters are varied inside their admissible range until an optimum minimizing the cost function and satisfying all constraints has been reached. Examples of methods for interpolating these knots (nodes) are high-order polynomial and piecewise polynomials (cubic- splines or B-splines).

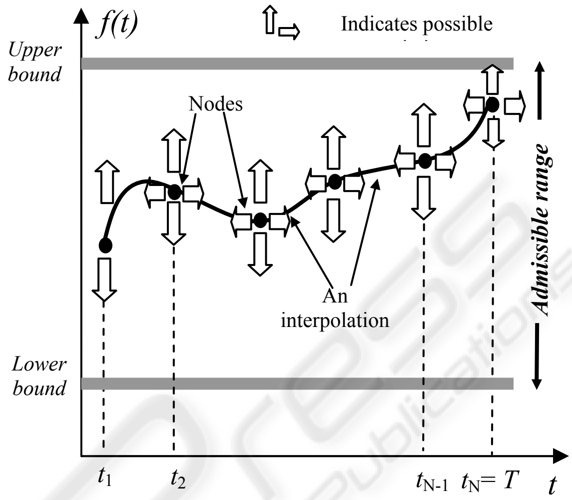

Figure 1: Discretization of continuous system's variables.

Note that also the location of these nodes along the time scale can be considered as additional unknowns of our problem. The interest of this fact arises particularly when we would like to capture critical evolution areas like switching time for bang-bang controls and, in consequence, to avoid excessive refinement of the time grid.

If *X* denotes the set of chosen parameters, the corresponding POP is to find the value of *X* that minimizes the cost function (7) written here:

$$\min_{\boldsymbol{X}} \quad \boldsymbol{J} = \boldsymbol{F}(\boldsymbol{X}) \qquad (10.a)$$

Subject to the equality constraints:

$$\boldsymbol{C}_{eq}(\boldsymbol{X}) = 0 \qquad (10.b)$$

And the inequality constraints:

$$\boldsymbol{C}_{ineq}(\boldsymbol{X}) \le 0 \qquad (10.c)$$

F, C_{eq} and C_{ineq} are just a transcription of relations (4), (6) and (7) in terms of the new variable *X*.

5.1 Conversion with independent states and controls as unknowns

In this first mode of conversion, both the control vector τ and state vector $\boldsymbol{X}(\boldsymbol{t}) = [\boldsymbol{q}(\boldsymbol{t})\ \dot{\boldsymbol{q}}(\boldsymbol{t})]^t$ are discretized. This involves an implicit integration of (4). This is performed by calculating the residuals on each subinterval and driving them to zero as a part of the optimization process. Hence we get additional equality constraints. This discretization can be performed according different schemes: midpoint rule, trapezoidal rule or in general using a Runge-Kutta scheme (Hull,1997) (Betts,1999).

5.2 Conversion with Controls as unknowns

In this conversion method, in addition to *T*, the unknowns are the values of τ at the nodes. The control history $\tau(t)$ is formed by interpolation. In this case, the dynamic equation (1), written under the state form (4), must be integrated on $[0,\boldsymbol{T}]$ to get the time evolution of the system state $\boldsymbol{X}(t)=[\boldsymbol{q}(t)\,\dot{\boldsymbol{q}}(t)]^t$ (i.e. to compute the FDM). Such a method can be seen as a shooting method because once a guess of $\tau(t)$ is made the sate equations are generally integrated in one pass. So, the time history $\tau(t)$ is varied until the final state (8*b*) is matched while all imposed constraints are respected and cost function is optimized.

5.3 Conversion with independent generalized coordinates as unknowns

In this conversion method, *T* and the values of independent generalized coordinates (defined from $q(t)$ at selected knots) are considered as the unknowns of the problem. The time history of $q(t)$ is then formed by interpolation. Time derivatives of $q(t)$, i.e. $\dot{q}$ and $\ddot{q}$, are systematically deduced. Then, torques τ are computed using IDM (relation (1)). So, all elements of the optimization problem, objective function and constraints, can be easily evaluated and then checked.

In this conversion, dynamic equations (1), exploited through the IDM, are employed just to verify any constraints imposed on torques τ. So, the motion generator can be seen here as a conventional kinematic planner.

6 SOLUTION TECHNIQUES

Once the original OCP has been transformed into a POP using the above mentioned methods, the problem can be treated by parametric optimization techniques. These techniques can be regrouped into two main families: deterministic and stochastic techniques. Deterministic methods use first and second order information (gradient and Hessian) to build a descent direction and to define a good progress step. In contrast, stochastic techniques need neither gradient nor Hessian values to process the optimization problem. They are based on randomized process able to select good candidates. We are not here going to establish a comparison between these two families but, just we mention that theses techniques can be used to solve the resulting POP.

7 NUMERICAL EXAMPLES

7.1 A 2 d.o.f. robot

We are concerned here by the IBM 7535 B04 robot modelled as a 2 d.o.f planar robot (Geering et al., 1986). It was the bench mark for many simulation works dealing with the minimum time trajectory planning problem (Geering et al., 1986, Chen and Desrochers, 1988, 1990, Chen et al, 1993, Bessonnet, 1992, Lazrak, 1996). We try here to solve this problem using parametric optimization instead of Pontryagin Maximum Principle (PMP). The three discretization schemes proposed in § 5 are used to transform the original problem. Then, we propose the SQP technique to solve the resulting NL problem. In all cases, the task to be achieved is characterized by null limit joint velocities while the motion starts at (0, 0) and ends at (0, $\pi/4$). Only constraints on torques are imposed ($|\tau_1|\le 25Nm, |\tau_2|\le 9Nm$). Table 1 summarizes the main simulation results we get using the three conversion modes.

In the first conversion mode, the robot's dynamic model has been discretized using a mid-point rule and using a uniform grid of 20 nodes. It has the biggest number of variables and constraints. However, the resolution of the corresponding NL program does not involve the highest CPU time. In fact, in the second mode of conversion, that involves only 29 variables, we need more time (26 times vs the first mode and 130 times vs the third mode) to solve the NL program. This is basically due to the fact that, at iteration of the optimization process, the robot dynamic model (4) is integrated using standard ODE solvers in order to get the corresponding kinematics that is time consuming. In reality, in this second mode, the optimization process behaves like a shooting method. Once $\tau(t)$ is estimated inside the admissible range, relation (1) is integrated from the initial state to a final state that must meet the state specified by the assigned task. This is traduced in the program by four equality constraints. In contrast to the two precedent modes, the last one seems to be more efficient, we have less variables and a good estimation of the optimal solution in a very short time (versus optimal solution for the same task found using a PMP based method and given in (Chen and Desrochers, 1990) that is 0.99s).

From this analysis, we think that the third mode is much more efficient and robust to handle the problem of trajectory optimization in more complex situations. The two following examples attest and confirm this point of view.

Table 1: Numerical results using direct and indirect methods

Discretization mode	Number of parameters	Objective function (s)	CPU time (s)
1) τ, x	127	1.02	20
2) τ	29	1.14	525
3) q	5	1.05	4
PMP (Chen & Desrochers,1990)		0.99	-

7.2 Optimal motion planning for a closed chain robot

We consider here a 2-DOF planar parallel robot (fig. 2). It consists of three identical two-link legs intersecting in a central point *C*. The robot configuration is defined by $[q_1, q_2, q_3, q_4, q_5, q_6]$. Let $q_a = [q_1, q_2, q_3]$ corresponds to active joints and $q_p = [q_4, q_5, q_6]$ to passive joints. The coordinates of *C* are $[x(t), y(t)]$ and can be expressed all the time as a function of q_a and q_p as follows:

$$\begin{cases} x(t) = l_1 \cos(q_i) + l_2 \cos(q_i + q_{i+3}) \\ y(t) = l_1 \sin(q_i) + l_2 \sin(q_i + q_{i+3}) \end{cases} \quad i = 1,\ldots,3$$

Where $\left\|\overrightarrow{A_i B_i}\right\| = l_1$, $\left\|\overrightarrow{B_i C}\right\| = l_2$.

So, the robot configuration can be parameterized using only the two independents coordinates $x(t)$ and $y(t)$. Knowing $[x(t), y(t)]$ and their time derivatives, active joints' torques can be computed using adequate techniques proper to closed chain robots (Cheng *et al.*, 2001). So, we decide to take $[x(t), y(t)]$ as unknowns for the motion planning problem.

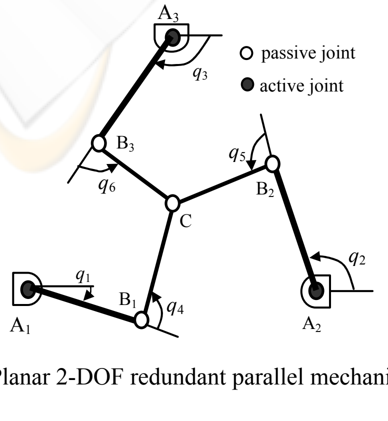

Figure 2: Planar 2-DOF redundant parallel mechanism

Table 2: Characteristics of one leg of the redundant planar parallel manipulator

j	L (m)	M (kg)	d (m)	I (kg.m²)	τ^{min} (N.m)	τ^{max} (N.m)	q^{min} (m)	q^{max} (m)
1	0.6	10	0.3	0.1	-25	25	–0.5	0.5
2	0.6	10	0.3	0.1	–	–	–0.5	0.5

Table 3: Numerical results for the Planar 2-DOF redundant parallel robot.

μ	Cost function(s)	Transfer time(s)
0.25	0.52	1.58
0.50	0.80	1.21
0.75	0.96	1.07
1.00	1.04	1.04

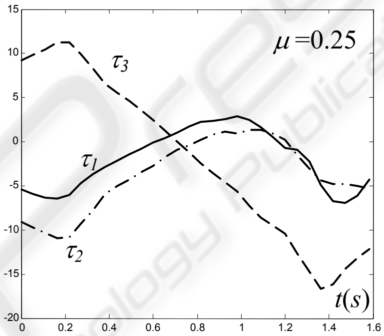

Figure 3*a*

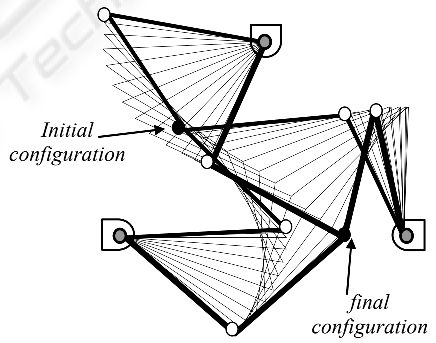

Figure 3*b*

Figure 3: Simulation results for the Planar 2-DOF redundant parallel robot: (*a*) active joint torques, (*b*) optimised motion, for μ = *0,25*

The assigned task here consists to achieve a transfer between an initial posture (x=-0.3, y=0.2) and a final one (x=0.3, y=-0.3) while minimizing the objective function (9) and respecting bounds (5*c*) and (5*g*) on the values of active joints' positions and torques (Table 2).

The time evolution of [$x(t)$, $y(t)$] is parameterized using seven free nodes uniformly distributed along the time scale. The simulation is done for various values of μ and corresponding results are reported in Table 3 and depicted on figure 5.

We observe that the proposed method succeeds to find a solution in both situations. Increasing the coefficient μ means that we attempt to minimize more T, this is guaranteed by higher torques amplitudes (increasing cost function). In fact, on fig. 5.a (μ=1), we note clearly the presence of several saturation areas of active joints' torques in order to ensure high speed transfer (fig.5.*c*). While, profiles of fig.5.*b* (μ=0.25) are quite smooth but the transfer is executed slower (fig.5.*d*).

Example 3: A mobile robot (Nonholonomic system)

This section gives numerical results concerning minimum–time trajectories ($\mu=1$) for a *Wheeled Mobile Robot* WMR constituted of a platform and two independently driven wheels (Yamamoto *et al.*, 1999). Constraints on driving torques are: $-1.0 \le \tau_1, \tau_2 \le 1.0 \quad (\boldsymbol{N.m})$. The workspace consists of a $24m \times 24m$ flat floor with three obstacles (Fig. 5*a*). The WMR is required to move freely, without following a specified path, from initial to final states $\boldsymbol{X}_0$ and $\boldsymbol{X}_f$ given by:

$$\mathbf{X}_0 = [x_0 \quad y_0 \quad \theta_0]^T == [\mathbf{3} \quad 3 \quad 0] \, ,$$
$$\mathbf{X}_f = [x_f \quad y_f \quad \theta_f]^T = [23 \quad 23 \quad \pi/6] \, .$$

In addition to the vector $\tau_a(t)$ of actuator efforts and the final time T, we must find the motion defined by $\boldsymbol{X}(t) = [x(t) \quad y(t) \quad \theta(t)]^T$ such as the initial and final states are matched, constraints are respected and the traveling time is minimized.

The robot independent position parameters are $x(t)$ and $y(t)$. The orientation $\theta(t)$ can be deduced from the nonholonomic constraint:

$$-\dot{x}(t) \cdot \boldsymbol{Sin}(\theta(t)) + \dot{y}(t) \cdot \boldsymbol{Cos}(\theta(t)) = 0 \, .$$

At each iteration of the optimization process the WRM motion $\boldsymbol{X}(t)$ is defined in two main steps.

***Step* 1** : specify the robot path $\boldsymbol{X}(\lambda)$.
***Step* 2** : specify the motion profile $\lambda(\xi)$ on this path.

$\boldsymbol{X}(\lambda)$, $\lambda \in [0, 1]$, describes the geometry of the robot path in the (O, $\boldsymbol{x}$, $\boldsymbol{y}$) plane while $\lambda(\xi)$, $\xi \in [0, 1]$, determines the time evolution along this path (ξ represents a normalized time scale: $\xi = t / T$).
Hence, the problem is transformed to a parametric optimization problem. One of the parameters is the unknown traveling time T. The other parameters are two sets, S_P and S_C, of free discretisation nodes. The set S_P is composed of N_P control points in the robot workspace (Fig. 4*a*) while S_C consists of N_C collocation points in the (ξ, λ) plane (Fig. 4*b*). With S_P, we can define a path $\boldsymbol{X}(\lambda)$ using parametric functions, such as B-spline, that takes into account limit states. With S_C, we can define a motion profile $\lambda(\xi)$ using, for example, a clamped cubic spline interpolation that takes into account the other constraints (Chettibi et al 2004*b*, Haddad et al, 2005).

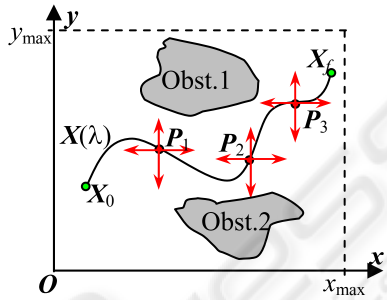

Figure 4a: A path $\boldsymbol{X}(\lambda)$ through N_P free control points

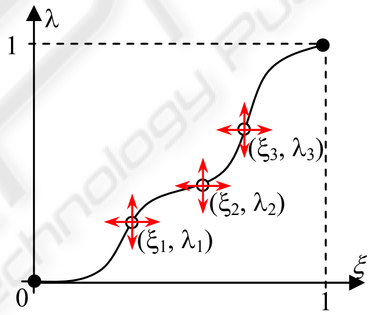

Figure 4b: A motion profile $\lambda(\xi)$ with N_C free collocation points

The optimization method adopted here uses a simulated annealing process that scans simultaneously the available solution space of both sets S_P and S_C to propose candidate trajectory profiles $\boldsymbol{X}(\xi) \equiv \boldsymbol{X}(\lambda(\xi))$ for a global minimization of the traveling time.

For this problem we have adopted for $\boldsymbol{X}(\lambda)$ a fourth-order B–spline model with $N_p = 6$ control points and for $\lambda(\xi)$ a clamped cubic spline model with $N_C = 4$ interpolation points. The required runtime was about 4 minutes on a 2.4 GHz P4.

Simulation results are shown in Figure 5. These results are quite similar to those given in (Yamamoto *et al.* 1999). The calculated traveling times are of the same order (17.82 vs. 18.94 *sec*).

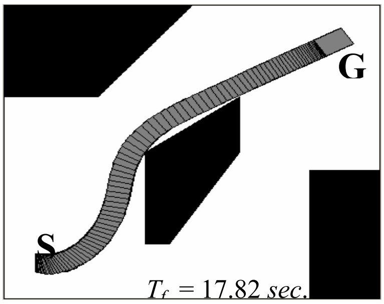

Figure 5a: Simulation result

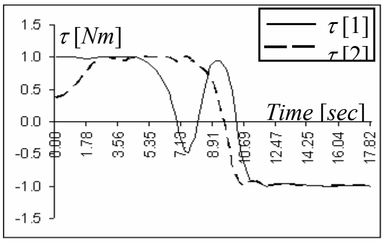

Figure 5b: Time evolution of joint torques

8 CONCLUSION

We have demonstrated that a trajectory optimization problem, that is an optimal control problem, can be converted into a parametric optimization problem using three different conversion modes. We shown that using independent position parameters as principle variables of the optimization problem offers many facilities and leads to comparable results to those obtained heavy and classical indirect methods.

Furthermore, the simplicity and the efficiency of this conversion mode allow us to use it to solve the problem of optimal trajectory planning in complex situations, in particular for holonomic and non-holonomic systems.

ACKNOWLEDGMENTS

Authors would like to thank Prof. H. E. Lehtihet for his suggestions and helpful discussions.

REFERENCES

Angeles J., 1997, *Fundamentals of robotic mechanical systems. Theory, methods, and algorithms*, Springer Edition.

Ascher U. M. & Petzold L. R., 1998, *Computer Methods for ordinary differential equations and differential-algebraic equations*, SIAM edition.

Bessonnet, G. 1992. *Optimisation dynamique des mouvements point à point de robots manipulateurs*. Thèse d'état, Université de Poitiers, France.

Betts, J. T. 1998. Survey of numerical methods for trajectory optimization. *Jour. Of Guidance, Cont. and Dyn.*, 21(2), 193-207.

Bryson A. E., 1999, *Dynamic Optimization*, Addison Wesley Longman, Inc.

Bicchi A., Pallottino L., Bray M., Perdomi P. , 2001, Randomized parallel Simulation of constrained multibody systems for VR/Haptic applications, *Proc. IEEE int. conf. on rob. & Aut*., Korea.

Bobrow, J. E., Martin, B. J., Sohl, G., Wang, E. C., Park, F. C., and Kim, J. 2001. Optimal robot motions for physical criteria. *Jour. Of Rob. Syst*. 18 (12), 785-795.

Chen, Y., and Desrochers, A. 1990. A proof of the structure of the minimum time control of robotic manipulators using Hamiltonian formulation. *IEEE Trans. On Rob. and Aut*. 6(3), pp388-393.

Chettibi T., H. E. Lehtihet, M. Haddad, S. Hanchi, 2004*a*, Minimum cost trajectory planning for industrial robots, *European Journal of Mechanics/A*, pp703-715.

Chettibi T., Haddad M., Rebai S., Hentout A., 2004*b*, A Stochastic off line planner of optimal dynamic motions for robotic manipulators, *1st Inter. Conf. on Informatics, in Control, Automation and Robotics*, Portugal.

Dombre E. & Khalil W., 1999, *Modélisation, identification et commande des robots*, second edition, Hermes.

Geering, H. P., Guzzella, L., Hepner, S. A. R., and Onder, C. H. 1986. Time-optimal motions of robots in Assembly tasks. *IEEE Trans. On Automatic control* , Vol. AC-31, N°6.

Haddad M., Chettibi T., Lehtihet H. E. and Hanchi S. 2005. "A new approach for minimum time motion planning problem of wheeled mobile robots", accepted at the 2005IFAC congress, Prague.

Hull, D. G. 1997. Conversion of optimal control problems into parameter optimization problems, *Jour. Of Guidance, Cont. and Dyn.*, 20(1), 57-62.

Lazrak, M. 1996. *Nouvelle approche de commande optimale en temps final libre et construction d'algorithmes de commande de systèmes articulés.* Thèse d'état, Université de Poitiers.

Steinbach M.C., 1995, Fast recursive SQP methods for large scale optimal control problem. PhD thesis, Universität Heidelberg.

Stryk, O. V. and Bulirsch R. 1993. Direct and indirect methods for trajectory optimization. *Annals of Operations research*, Vol 37, pp. 357-373.

Stryk, O. V. 1993, Numerical solution of optimal control problems by direct collocation. *Optimal control theory and numerical methods, Int. series of Numerical Mathematics,* Vol 111, pp 129-143.

Yamamoto, M., Iwamura M. and Mohri A. (1999). *Quasi-time-Optimal Motion Planning of Mobile Platforms in the presence of obstacles*, Proc. Of 1999 IEEE ICRA, 2958-2962.