

A Method for Trajectory Optimization of Robots having Contacts or Motion Constraints

Sylvain Miossec, Kazuhito Yokoi

JRL AIST

Tsukuba, Japan

sylvain.miossec@aist.go.jp, kazuhito.yokoi@aist.go.jp

Abderrahmane Kheddar

JRL CNRS

Tsukuba, Japan

kheddar@ieee.org

Abstract—The problem of motion trajectory optimization with constraints is considered for robotic system with actuated joints. Constraints concern contacts and path-following in the Cartesian space. These constraints arise in walking patterns. In this study we are dealing with cases where the same constraints hold during a given motion. The trajectory optimization is performed by defining some state variables as polynomials, and the best parameters of the polynomial functions are obtained from suitable optimization programs. By taking into account motion constraints, it appears that the number of the problem's optimization parameters can be reduced. Subsequently, the structure of the optimization problem is different. A procedure to solve this optimization problem is proposed. A simple case is used to compare the proposed methods with the one that is used for solving the similar problem without parameters reduction.

I. INTRODUCTION

Considerable work have been devoted to optimal control of systems. There are mainly two numerical approaches to solve such problems: (i) indirect methods are based on the Pontryagin's Maximum Principle which results in solving a two-points boundary value problem, and (ii) direct methods that solve the discretized problem by using parametric optimization techniques. Indirect methods give more precise results relatively to direct ones, but the convergence domain is smaller. In [1], authors used a variational approach to solve the trajectory optimization problem. In [2], [3], [4] the authors used a collocation method. The authors of [5], [6], [7], [8], [9], [10] defined the evolution of joints or cartesian space variables with polynomials or spline functions. Polynomials and splines are characterized with boundary values or control points, which are the optimization variables.

This paper deals with optimization of either contact trajectories or geometrically constrained ones. Contact trajectories require solving for the distribution of internal forces due to over-actuation when a contact occurs. Afterwhat, the optimization problem to be solved is actually the same in both formulations. To generate a motion with geometric constraint, we suggest using a minimal number of parameters since the system has less degrees of freedom (dof). The problem of optimal trajectory generation with contacts have been considered in [1], [7], [8], [9]. In [10] closed chain mechanisms

have been considered. In [1], [8], [9] geometric constraints are simply formulated as equality constraints at the control points. Reduction of parameters' number is proposed in [5], [6], [7], [10] (in [5], [6] geometric constraints were only considered at discrete instants of the motion). Then the missing parameters are determined from the reduced ones. One need to check the existence of the remaining parameters: this corresponds to a situations where solution to an inverse geometric model doesn't exist. However, in [5], [6], [7], [10] the existence issue was not considered. In this paper we emphasize rigorously on the obtained optimization problem when the number of parameters is reduced in the geometric constraint case. We consider the conditions of existence for the inverse geometric problem; we show that the problem to be solved cannot be handled by classic SQP or Interior point methods. An optimization procedure to solve such problems is presented.

A comparative study is made between our parameters reduction method and a method without reduction. There have been some work devoted to comparative studies on trajectory optimization problems, see [11], [12], [13], but not in the presence of contacts. For robotic manipulators, [11], [12] have shown that the trajectory optimization is faster when torques computation is obtained from the inverse dynamic model (relatively to having a collocation equation between joint angles and torques). The inverse model is computed faster than the direct one. Moreover, the number of optimization's iterations is more important even with a reduced number of parameters. The gain of the overall computation time is subsequent to the dynamics model computation rather than to simplifications of the optimization problem. However, [13] claimed that using flat outputs or, at least, outputs with higher relative degree, decreases the size of the searched space for the trajectory optimization; hence decreasing the computation time. In this case, the gain of computation time is directly linked to the reduction of the search space for the trajectory optimization. Therefore, we want to investigate, in the case of trajectory optimization with contact, if there are some benefits in reducing the dimension of the searched space.

This paper is organized as follow: in section II we illustrate the general optimization problem in the case of geometric constraint in both cases: without and with reduction of the optimization parameters. Then, in section III we compare, for a two-link robot with a geometric constraint, the two

This research has been conducted at the JRL, Intelligent Systems Research Institute, AIST, Central 2, 1-1-1, Umezono, Tsukuba, 305-8568 JAPAN and was supported by a Post-doctoral Fellowship of the Japan Society for the Promotion of Science (JSPS)

formulations of the trajectory optimization.

II. GENERAL FORMULATION OF THE OPTIMIZATION PROBLEM TO SOLVE

A. System studied

We consider robotic systems in which all joints are actuated; the dynamic model obtained from Lagrange formulation is:

$$u = A(q)\ddot{q} + H(q, \dot{q}) - J(q)^T F_c \quad (1)$$

$u \in \mathbb{R}^n$ is the control input, $q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$, $\ddot{q} \in \mathbb{R}^n$ are respectively the joints' positions, velocities and accelerations, F_c is the vector of contact force exerted by the environment, $A(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, and $H(q, \dot{q}) \in \mathbb{R}^n$ is the vector of Coriolis, centrifugal and gravity effects, $J(q)$ is the jacobian matrix. Joint limits are expressed in the form:

$$q_{\min} \leq q \leq q_{\max} \quad (2)$$

$q_{\min}$, $q_{\max}$ are respectively the minimal and maximal limits. Actuators' limitations are additional constraints that can be approximated by linear inequalities on u and $\dot{q}$, that is:

$$au + b\dot{q} + c \leq 0 \quad (3)$$

B. Characteristics of the desired motion

Let a motion be for time $t \in [t_0, t_f]$, where $t_0 = 0$. The motion takes place in an environment having obstacles. Constraints describing static obstacles avoidance can be written:

$$c(q(t)) \leq 0 \quad (4)$$

Two type of motions are considered: motions under a contact linking part of the robot to the environment, and constraint motion along a planned path. Both constraints can be rewritten as geometric equalities of the form:

$$g(q(t)) = 0 \quad (5)$$

When contacts with the environment occur, additional force constraints (6) must be satisfied; they stand for the unilaterality of the contact and for Coulomb friction law.

$$s(F_c, \dot{q}(t)) \stackrel{=}{\leq} 0 \quad (6)$$

Finally additional constraints on the initial and the final states that can be written as (7). For example, constraints that are defined for an optimal motion between a given initial and final state (point-to-point motion).

$$\begin{cases} g_{q_0}(q(t_0)) = 0 \\ g_{\dot{q}_0}(q(t_0), \dot{q}(t_0)) = 0 \\ g_{\ddot{q}_0}(q(t_0), \dot{q}(t_0), \ddot{q}(t_0)) = 0 \\ g_{q_f}(q(t_f)) = 0 \\ g_{\dot{q}_f}(q(t_f), \dot{q}(t_f)) = 0 \\ g_{\ddot{q}_f}(q(t_f), \dot{q}(t_f), \ddot{q}(t_f)) = 0 \end{cases} \quad (7)$$

We aim at motions that minimize the following criteria:

$$\begin{aligned} \mathcal{C}(q(t), \dot{q}(t), u(t)) \\ = \phi(q(t_f), \dot{q}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(q(t), \dot{q}(t), u(t)) dt \end{aligned} \quad (8)$$

When $\mathcal{L}(q(t), \dot{q}(t), u(t)) = 0$ the criteria depends only on end-motion states, e.g. the desired finale velocity or time. The integral term can represent energy consumption during all the motion. With all previous characteristics of the desired motion, the trajectory optimization problem is given by:

$$\begin{aligned} \min_{q(t), \dot{q}(t), u(t), t_f} \mathcal{C}(q(t), \dot{q}(t), u(t)) \\ \text{subject to (1)} \cdots (7) \end{aligned} \quad (9)$$

The goal is to find the optimal functions $q(t)$, $u(t)$ and the final time t_f . For simple problems, the Pontryagin's Maximum Principle allows finding analytical solutions. In the general case, no method exist to find $q(t)$, $u(t)$ and t_f exactly and numerical methods are used instead. In sections II-C and II-D we present two ways of approximating this problem so that it can be solved with existing numerical methods.

C. Optimization problem in the case without reduction

An idea to solve the problem (9), called the direct method, see [2], [3], [4], [5], [6], [7], [8], [9], [11], [12], is to discretize $q(t)$ and sometimes $u(t)$. Then the new discrete problem is solved with an optimization program, such as the SQP method, see [14], [15] or an Interior point method (IPM), see [16]. Different discretization methods were proposed. We define $q(t)$ as a sequence of polynomials connected at intermediate points. Those polynomials satisfy continuity conditions at connection points. $u(t)$ will be determined thanks to the inverse model (1), as done in [5], [6], [7], [8], [9], [11], [12].

The points of connection of the polynomials are given by the times $t_0 < t_1 < \cdots < t_{N-1} < t_N = t_f$. At connection instants, the variables $q(t)$ are written as:

$$\begin{cases} q_k = q(t_k) & k = 0 \cdots N \\ \dot{q}_k = \dot{q}(t_k) & k = 0 \cdots N \\ \ddot{q}_k = \ddot{q}(t_k) & k = 0 \cdots N \end{cases} \quad (10)$$

The expression of the function $q(t)$ written with polynomials over $t \in [t_k, t_{k+1}]$, $k \in [0 \cdots N-1]$ is given by:

$$q^j(t) = \sum_{i=0}^{d_k} a_{ki}^j (t - t_k)^i \quad j = 1 \cdots n \quad (11)$$

where $q^j(t)$ is the j^{th} component of the vector of functions $q(t)$, d_k is the degree of the polynomials of the time interval $[t_k, t_{k+1}]$, $k \in [0 \cdots N-1]$, a_{ki}^j are the polynomials coefficients. The coefficients a_{ki}^j and the degree of polynomials are determined from the values of the variables at connection time, i.e. q_k , $\dot{q}_k$ and $\ddot{q}_k$, and from the continuity conditions at connection points. Hence, q_k , $\dot{q}_k$ and $\ddot{q}_k$ are considered as parameters. In fact we can just choose q_k or $\dot{q}_k$ with $\ddot{q}_k$, and impose the continuity up to the chosen derivative level. Considering the three parameters q_k , $\dot{q}_k$ and $\ddot{q}_k$, we have $d_k = 5$ for all intervals, and the 6 coefficients of each polynomial are determined from its six limit conditions. We consider equidistant connection points, starting from $t_0 = 0$. Then all t_i are determined from t_N and the vector of all the parameters is given by $p = [q_0, \dot{q}_0, \ddot{q}_0, \cdots, q_N, \dot{q}_N, \ddot{q}_N, t_N]$; size of is $3(N+1)n+1$. With this discretization, the equality constraints

must be written only at connection points. Otherwise the problem can be over-constrained. Hence, equality constraints (5) become:

$$\begin{cases} g(q_k) = 0 \\ G(q_k)\dot{q}_k = 0 \\ G(q_k)\ddot{q}_k + H(q_k, \dot{q}_k) = 0 \end{cases} \quad k = 0 \dots N \quad (12)$$

where $G(q_k) = \frac{\partial g}{\partial q} \Big|_{q=q_k}$ and $H(q_k, \dot{q}_k) = \frac{dG}{dt} \Big|_{t=t_k} \dot{q}_k$. (12) includes the derivations of the constraints (5) up to the last chosen derivative order.

The optimization problem (9) is rewritten in the form:

$$\begin{aligned} \min_p \mathcal{C}(q(p, t), \dot{q}(p, t), \ddot{q}(p, t)) \\ \text{subject to (2), (3), (4), (6), (7), (12)} \end{aligned} \quad (13)$$

In (13), the criteria is not written with respect to $u(t)$, since $u(t)$ is determined from (1). In the case of contact constraints, there might be an infinite F_c satisfying (6), hence an infinite $u(t)$ representing different repartition of internal forces in the closed-chain. We compute $u(t)$ by solving the problem (14) which gives $u(t)$ that minimizes the criteria (8) while respecting the dynamic constraints (1), (3) and (6).

$$\begin{aligned} \min_{u(t)} \mathcal{L}(q(t), \dot{q}(t), u(t)) \\ \text{subject to (1), (3), (6)} \end{aligned} \quad (14)$$

$\mathcal{L}(q(t), \dot{q}(t), u(t))$ is usually quadratic in $u(t)$. In 2D case (1), (3) and (6) are linear constraints in $u(t)$; hence, this problem is a classical QP. In the 3D case, the contact constraints (6) are not linear in $u(t)$; but, it is possible to make a linearization. Moreover, in (13) some inequality constraints are written $\forall t \in [t_0, t_f]$, which makes (13) become a semi-infinite optimization problem. A solution to solve such problems is to discretize the semi-infinite constraints. It is possible to use different discretization points than the times of connection points. The obtained problem can be solved by any SQP-based optimization algorithms, e.g. NPSOL [14], or FSQP [15].

D. Optimization problem in the reduction case

Our proposed problem formulation is similar to the one presented in the previous section. It is based on the discretization of the problem by writing $q(t)$ as a sequence of polynomials. The difference is in reducing the number of parameters as much as possible by using the constraints (5) and (7).

A first approach reduces the problem to the definition of the evolution of as much variables as available dof. For this, we solve (5) with respect to some components of $q(t)$:

$$q_b(t) = h(q_a(t)) \quad (15)$$

$q(t)$ is partitioned as $q(t) = [q_a(t) \ q_b(t)]$; $q_a(t)$ is defined as polynomials (11), and $q_b(t)$ is deduced from (15). Note that the inversion giving (15) is subject to restrictions. For this inversion to be possible, some constraints must hold, that is:

$$f(q_a(t)) \leq 0 \quad (16)$$

If $q_a(t)$ does not satisfy (16), there is no solution for the inversion of (5), and $q_b(t)$ is not defined.

In fact we have observed from the example of a 2-link robot (section III), that it is better to use a different approach. When there is a constraint on the position of a given point of the system, it seems better to write as much as possible the motion in this space. Rather than mixing the spaces in which motion is written. Therefore, we will define partially a motion in the cartesian space, in where most of the constraints are written. We note by $z(t) \in \mathbb{R}^{n_z}$ the absolute position and/or orientation of one or several points of the system. $z(t)$ only includes the points that are involved in the constraints (5). The geometric constraint (5) can be written in function of $z(t)$, that is:

$$g_z(z(t)) = 0 \quad (17)$$

where $g_z : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_c}$ ($n_z \geq n_c$). We consider the partitioning of $z(t) = [z_a(t) \ z_b(t)]$, $z_a(t) \in \mathbb{R}^{n_z - n_c}$ and $z_b(t) \in \mathbb{R}^{n_c}$, for which it is possible to determine $z_b(t)$ by inverting (17). $z_a(t)$ is defined as a polynomial like (11). The expression giving $z_b(t)$ from $z_a(t)$ is:

$$z_b(t) = h_z(z_a(t)) \quad (18)$$

Usually, there is no constraints for the existence of this inversion, and this expression is simple when $z_b(t)$ is constant.

By defining $z_a(t)$, we have defined only part of the motion. The remaining dof are defined in the joint space $q(t)$. But we have to partition $q(t) = [q_a(t) \ q_b(t)]$. The joint variables $q_a(t) \in \mathbb{R}^{n - n_z}$ are defined as polynomials. The joint variables $q_b(t) \in \mathbb{R}^{n_z}$ are obtained by inverse geometric model from $z(t)$. The expression of the inverse geometric model is:

$$q_b(t) = h_{qz}(q_a(t), z(t)) \quad (19)$$

Here the existence of $q_b(t)$ is subject to:

$$f_{qz}(q_a(t), z(t)) \leq 0 \quad (20)$$

Furthermore, by derivation of relations (18) and (19), we obtain the relations (21) and (22). Those relations allow to determine $\dot{q}(t)$ and $\ddot{q}(t)$ from $\dot{z}_a(t)$, $\ddot{z}_a(t)$, $\dot{q}_a(t)$ and $\ddot{q}_a(t)$:

$$\begin{cases} \dot{z}_b(t) = H_z(z_a(t))\dot{z}_a(t) \\ \ddot{z}_b(t) = H_z(z_a(t))\ddot{z}_a(t) + H'_z(z_a(t), \dot{z}_a(t)) \end{cases} \quad (21)$$

where $H_z(z_a(t)) = \frac{\partial h_z}{\partial z_a}$ and $H'_z(z_a(t), \dot{z}_a(t)) = \frac{dH_z}{dt} \dot{z}_a$. There is generally no restrictions for the existence of $\dot{z}_b(t)$ and $\ddot{z}_b(t)$.

$$\begin{cases} \dot{q}_b(t) = H_{qz}(q_a(t), z(t))[\dot{q}_a(t)^T \ \dot{z}(t)^T]^T \\ \ddot{q}_b(t) = H_{qz}(q_a(t), z(t))[\ddot{q}_a(t)^T \ \ddot{z}(t)^T]^T \\ \quad + H'_{qz}(q_a(t), z(t), \dot{q}_a(t), \dot{z}(t)) \end{cases} \quad (22)$$

where $H_{qz}(q_a(t), z(t)) = \frac{\partial h_{qz}}{\partial q_a}$ and $H'_{qz}(q_a(t), z(t), \dot{q}_a(t), \dot{z}(t)) = \frac{dH_{qz}}{dt} [\dot{q}_a(t)^T \ \dot{z}(t)^T]^T$.

For the existence of solutions to (22), the constraint (20) must hold. But this constraint must be strictly satisfied, since the case $f_{qz}(q_a(t), z(t)) = 0$ is generally be a singularity. Therefore, in the optimization process, we will consider the constraint (23) rather than (20).

$$f_{qz}(q_a(t), z(t)) + \epsilon \leq 0 \quad (23)$$

where $\epsilon > 0$ will be chosen very small.

From (18) and (19), $q(t)$ is obtained from $z_a(t)$ and $q_a(t)$; thus only $z_a(t)$ and $q_a(t)$ are defined as polynomials. At instants $t_k \in [1 \dots N-1]$, (18), (19), (21) and (22) hold. The optimization parameters of instant t_k are $[z_{a_k}, \dot{z}_{a_k}, \ddot{z}_{a_k}, q_{a_k}, \dot{q}_{a_k}, \ddot{q}_{a_k}]$ (keeping the subscribe k to note the variables at instant t_k). At each t_k , there is $3(n - n_c)$ parameters instead of $3n$ for the method of the previous section.

For the instants t_0 and t_N the additional equality constraints (7) hold. Then, the number of parameters is reduced further. We assume that the optimization problem (9) is well defined; that is, constraints (5) and (7) are independent. We consider two cases for constraints (7): (i) if equality constraints are written on cartesian coordinates; it is possible to reduce the parameters as proposed, i.e. just by adding constraints (7) to constraints in (17), and (ii) if the constraints (7) are explicitly written in the joint space; the components of q_a , $\dot{q}_a$ or $\ddot{q}_a$ are fixed. The constrained components of q_b , $\dot{q}_b$ or $\ddot{q}_b$ are rewritten as constraints on z , using the geometric model. These new constraints are added to those in (17). Therefore, if there are n_i initial constraints and n_f final constraints in (7), there will be $3(n - n_c) - n_i$ initial parameters and $3(n - n_c) - n_f$ final parameters. And, the vector of all the parameters is given by $p_r = [z_{a0}, \dot{z}_{a0}, \ddot{z}_{a0}, q_{a0}, \dot{q}_{a0}, \ddot{q}_{a0}, \dots, z_{aN}, \dot{z}_{aN}, \ddot{z}_{aN}, q_{aN}, \dot{q}_{aN}, \ddot{q}_{aN}, t_N]$ and is of size $3(N + 1)(n - n_c) - n_i - n_f + 1$.

Finally the optimization problem we obtain does not include equality constraints anymore, that is:

$$\begin{aligned} \min_{p_r} \mathcal{C}(q(p_r, t), \dot{q}(p_r, t), \ddot{q}(p_r, t)) \\ \text{subject to (2), (3), (4), (23)} \end{aligned} \quad (24)$$

For some values of p_r , the vectors $q(p_r, t)$, $\dot{q}(p_r, t)$ and $\ddot{q}(p_r, t)$ are not defined, and hence the optimization problem is not defined, because the constraint (23) doesn't hold. So during the optimization processing, the constraint (23) must hold. Yet, usual SQP-based and Interior point methods allow violation of the constraints during the optimization processing: they cannot be used in this case. Therefore, one can use feasible optimization methods such as FSQP or KNITRO programs, which guarantee constraint holding during the optimization. In our case study, KNITRO proved to be much faster, because it has been designed to handle sparse problems. On the contrary, FSQP can only handle dense problems. In motion optimization problems, sparsity increases with the number of phases N of the problem. KNITRO optimization algorithm is built in two steps: the first step solves the problem without feasibility guarantee until a feasible point is found. The second step continues solving the problem while always satisfying the constraints. We will adapt this procedure for our problem using three steps:

- 1) minimize the maximum of the constraints (23) until they hold;
- 2) minimize the maximum of the constraints (2), (3), (4) until they hold while always satisfying the constraints (23);

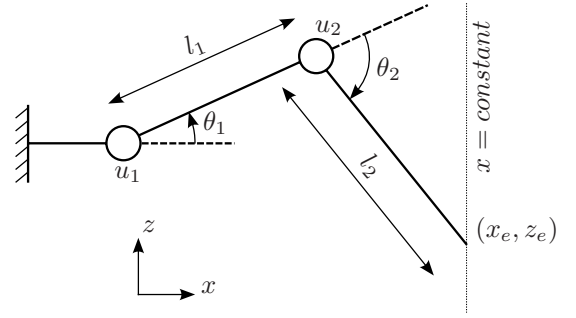


Fig. 1. Illustration of the 2dof robot considered

- 3) minimize the criteria while always satisfying all the constraints;

With this procedure, the constraints (2), (3), (4) and the criteria will never be evaluated in a case where they are not defined.

III. COMPARISON OF THE TWO APPROACHES

A. Application-case study with the two methods

1) *A 2dof robot and its considered motions:* We consider a two-link planar robot, see Fig. 1, actuated at both joints $q = [\theta_1 \ \theta_2]^T$ through $u = [u_1 \ u_2]^T$ respectively. The joint limits of the constraints (2) are given by $\theta_{1\min} = -\frac{\pi}{2}$, $\theta_{1\max} = \frac{\pi}{2}$, $\theta_{2\min} = -\frac{\pi}{2}$ and $\theta_{2\max} = \frac{\pi}{2}$. The actuators limits are:

$$\begin{cases} \pm u_1 \dot{\theta}_{1\max} \pm \dot{\theta}_1 u_{1\max} - \dot{\theta}_{1\max} u_{1\max} \leq 0 \\ \pm u_2 \dot{\theta}_{2\max} \pm \dot{\theta}_2 u_{2\max} - \dot{\theta}_{2\max} u_{2\max} \leq 0 \end{cases} \quad (25)$$

$\dot{\theta}_{1\max}$ and $\dot{\theta}_{2\max}$ are the maximal velocities when $u_1 = 0$ and $u_2 = 0$ respectively. $u_{1\max}$ and $u_{2\max}$ are the maximal torques when $\dot{\theta}_1 = 0$ and $\dot{\theta}_2 = 0$ respectively.

We consider a constrained vertical motion of the end-effector, that is:

$$x_e(t) = 0.363\text{m} \quad (26)$$

We also considered a point-to-point motion starting from $z_{e0} = z_{e\min} + 0.1$ and going to $z_{ef} = z_{e\max} + 0.1$. $z_{e\min}$ and $z_{e\max}$ are respectively the minimum and maximum reachable positions. We have the following initial and final constraints:

$$\begin{cases} z_{e0} = z_{e\min} + 0.1 \text{ m} \\ z_{ef} = z_{e\max} - 0.1 \text{ m} \\ \dot{\theta}_{10} = \dot{\theta}_{20} = \dot{\theta}_{1f} = \dot{\theta}_{2f} = 0 \\ \ddot{\theta}_{10} = \ddot{\theta}_{20} = \ddot{\theta}_{1f} = \ddot{\theta}_{2f} = 0 \end{cases} \quad (27)$$

Finally, we consider the energy criteria to be:

$$\mathcal{C}(q(t), \dot{q}(t), u(t)) = \int_{t_0}^{t_f} \sum_{j=1,2} \max \left(\frac{R u_j^2}{K_{em}^2} + u_j \dot{\theta}_j, 0 \right) dt \quad (28)$$

where R is the resistance and K_{em} the electro-mechanical constant of both actuators. This criteria represents the energy lost in the actuators' resistance and the mechanical energy transferred to the robot. But the function $\max(\cdot, 0)$ means that the energy returned by the robot is not reused nor consumed, just dissipated in a resistance.

2) *Optimization problem without reduction:* We consider a motion with 4 to 10 phases. The formulation of the problem without reduction, see section II-C, the equality constraints are kept in the optimization problem. Here, we reduce the number of parameters with the initial and final constraints (27). Therefore the vector of parameters, composed of $6N - 5$ parameters, is given by $p = [\theta_{11}, \theta_{21}, \dot{\theta}_{11}, \dot{\theta}_{21}, \ddot{\theta}_{11}, \ddot{\theta}_{21}, \dots, \theta_{1N-1}, \theta_{2N-1}, \dot{\theta}_{1N-1}, \dot{\theta}_{2N-1}, \ddot{\theta}_{1N-1}, \ddot{\theta}_{2N-1}, t_f]$. The motion constraint (26) is rewritten at connection points as:

$$x_{e_j} = 0.353m \quad \dot{x}_{e_j} = 0 \quad \ddot{x}_{e_j} = 0 \quad j = 0, 1, 2 \quad (29)$$

The optimization problem to solve reduces to minimize (28) with respect to p , with the constraints (2), (25), (27) and (29).

3) *Optimization problem with reduction:* We consider the same number of connection points as in the case of no parameters' reduction. The vertical motion constraint applies to the end-effector. We define part of the motion in the cartesian space. The part of the motion defined is $z_e(t)$, as a polynomial function similarly to (11). $x_e(t)$ is obtained from (26). In fact, for this case study, it is appropriate to define the motion in the Cartesian space; there is no need to define additional joint variables as polynomials. $\theta_1(t)$ and $\theta_2(t)$ are obtained from the inverse geometric model. With notations of II-D, $z_e(t)$ stands for $z_a(t)$, $x_e(t)$ for $z_b(t)$, $q_a(t)$ is empty, $q_b(t)$ is $\theta_1(t)$ and $\theta_2(t)$. The constraint concerning the existence of the inverse model solution:

$$\begin{cases} -z_{e_{\max}} < z_e(t) < z_{e_{\max}} \\ z_{e_{\max}} = \sqrt{(l_1 + l_2)^2 - 0.363^2} \end{cases} \quad (30)$$

The constraint concerning the existence of the inverse model solution (30) is equivalent to (20). Then the proposed reduction in section II-D, leads to the vector of parameters $p_r = [z_{e1}, \dot{z}_{e1}, \ddot{z}_{e1}, \dots, z_{eN-1}, \dot{z}_{eN-1}, \ddot{z}_{eN-1}, t_f]$ with $3N - 2$ parameters. The optimization problem reduces to minimize (28) with respect to p_r , with the constraints (2), (25) and (30). This problem is solved with the three steps proposed in section II-D.

B. Results

Using both formulations, we conducted several optimizations with different number of phases using MATLAB together with the optimization program KNITRO [16]. The optimization is initialized with a motion parameterized by a 5 order polynomial going from initial to final states. Also, an algebraically computed gradient is provided to the optimization program to increase the computation speed and improve the convergence near the solution. The solution is presented Fig. 2. Another solution is found by using the other inverse geometric model solution. The Fig. 3 summarizes the obtained results. The optimization times are obtained on a Pentium Xeon 2.3GHz. They can be improved using programming languages, like Fortran or C/C++. The Fig. 4 presents the maximum violation of the path constraint (in position).

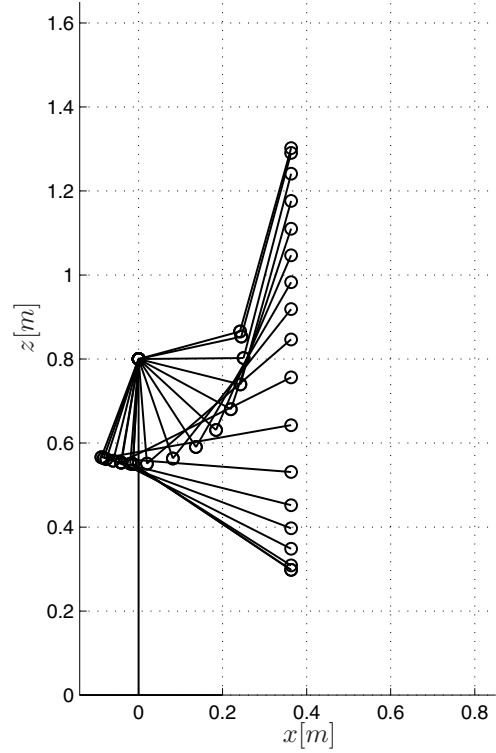


Fig. 2. Optimal motion obtained

C. Comments

The Fig. 3 show that the reduction based method requires more time and iterations for small number of phases (4 and 5 phases), still the precision of path constraint satisfaction decreases, see Fig. 4. The reduction method becomes however faster for higher number of phases. From curves' shapes extrapolation, the reduction method seems to be suitable for higher number of phases, and allows to satisfy the path constraint even for very few number of phases. To conclude, the reduction of parameters seems better both for any number of phases.

These results have obviously some limitations. First, the parameter reduction is difficult to implement because it needs additional modules (inverse models). Also, the differences between the two methods will depend on the number of contacts. Indeed, the reduction method for motions with many contacts is beneficial whereas it is not with small number of contacts. Hence, this approach is appropriate to humanoids' optimal walking motions, and even better suited to more legged robots. Another limitation of the reduction method is that expressing the motion in the Cartesian space will not allow motions close to singularities; whereas working in the joint space do not restrict such motions. Finally, the proposed reduction can only be systematically applied to systems with independent contacts that are explicitly written in the Cartesian space. There is still work to be done in order to extend the method to systems having contact state changing during a motion, as in [1].

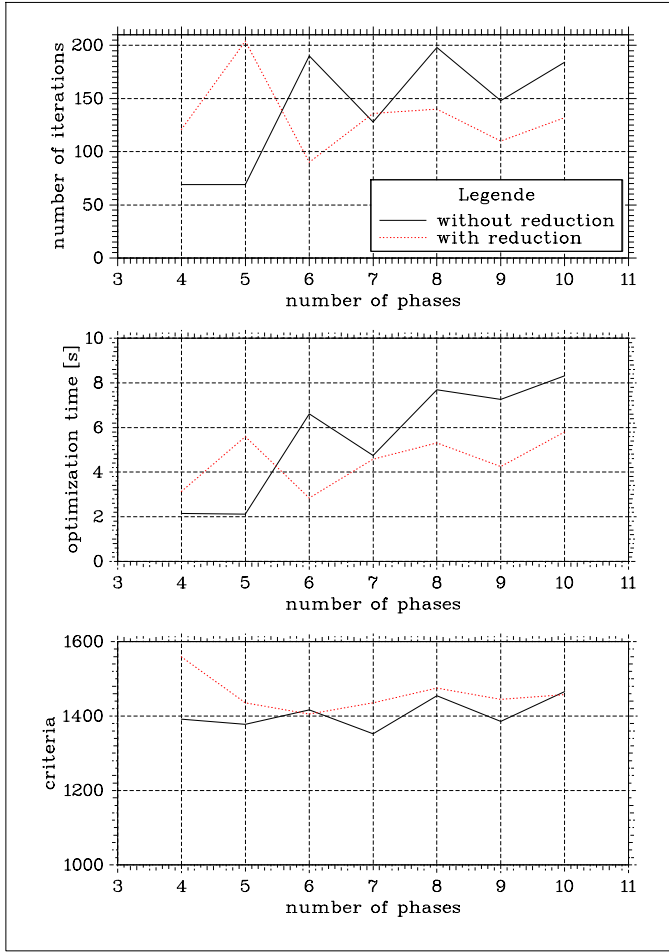


Fig. 3. Comparison of the results of optimization. The top and middle curves show that the reduction of parameters is more efficient for more phases. The bottom curves show that the values of energy consumption are close for both situations, except for few number of phases; in this situation the no reduction case allows smaller criteria due to some path constraint violation (Fig. 4).

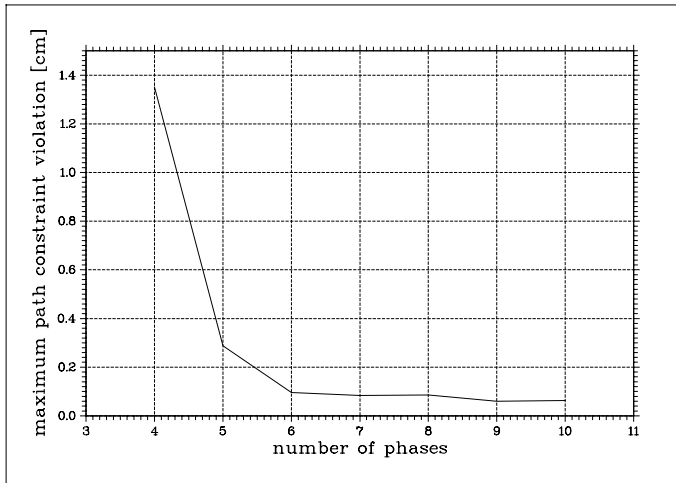


Fig. 4. Errors in verification of path constraint for the no reduction case: the fewer the number of phases is, the larger is the violation of the path constraint.

IV. CONCLUSION

In this paper we discuss how to perform parameters reductions for the trajectory optimization of robotic systems with contacts or motion constraints. We show that the optimization problem to be solved is of particular form and cannot be solved with classical SQP methods and Interior point methods, but only with feasible algorithms, such as the program KNITRO or FSQP. We illustrate on an example that the trajectory optimization is better. But, the proposed approach can only be applied for motions with independent contacts, explicitly written in the cartesian space, and for completely actuated systems. In future work, we aim at applying the proposed method to obtain optimal walking motions, and other motions with contacts for the humanoid robot HRP-2 [17].

REFERENCES

- [1] M. Zefran and V. Kumar, "Optimal control of systems with unilateral constraints," in *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, 1995.
- [2] M. Hardt, K. Kreutz-Delgado, and J. W. Helton, "Optimal biped walking with a complete dynamical model."
- [3] J. Denk and G. Schmidt, "Synthesis of a walking primitive database for a humanoid robot using optimal control techniques," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2001, pp. 319–326.
- [4] M. C. Steinbach, H. G. Bock, G. V. Kostin, and R. W. Longman, "Mathematical optimization in robotics: Towards automated high speed motion planning," *Surveys on Mathematics for Industry*, 1997.
- [5] P. H. Channon, S. H. Hopkins, and D. T. Pham, "Derivation of optimal walking motions for a biped walking motion," *Robotica*, vol. 10, no. 2, pp. 165–172, 1992.
- [6] C. Chevallereau and Y. Aoustin, "Optimal reference trajectories for walking and running of a biped robot," *Robotica*, vol. 19, pp. 557–569, 2001.
- [7] T. Saidouni and G. Bessonnet, "Generating globally optimised sagittal gait cycles of a biped robot," *Robotica*, vol. 21, pp. 199–210, 2003.
- [8] J. Lo, G. Huang, and D. Metaxas, "Human motion planning based on recursive dynamics and optimal control techniques," *Multibody System Dynamics*, vol. 8, pp. 433–458, 2002.
- [9] P. Seguin and G. Bessonnet, "Generating optimal walking cycles using spline-based state-parameterization," *International Journal of Humanoid Robotics*, vol. 2, no. 1, pp. 47–80, 2005.
- [10] S.-H. Lee, J. Kim, F. Park, M. Kim, and J. Bobrow, "Newton-type algorithms for dynamics-based robot movement optimization," *IEEE Transactions on Robotics*, vol. 21, no. 4, August 2005.
- [11] O. V. Stryk, "Optimal control of multibody systems in minimal coordinates," in *Proceedings of the Annual GAMM Conference*, 1997.
- [12] M. C. Steinbach, "Optimal motion design using inverse dynamics," Tech. Rep., 1997.
- [13] N. Petit, M. B. Milam, and R. M. Murray, "Inversion based constrained trajectory optimization," in *2001 IFAC Symposium on Nonlinear Control Systems Design*, 2001.
- [14] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's guide for NPSOL 5.0*.
- [15] C. Lawrence, J. L. Zhou, and A. L. Tits, *User's guide for cfsqp 2.5: a C code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints*, Electrical Engineering Department and Institute for Systems Research, University of Maryland.
- [16] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps," Optimization Technology Center, Northwestern University, Evanston, IL, USA, Tech. Rep., 2003.
- [17] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, K. Hirata, M. Akachi, and T. Isozumi, "Humanoid robot hrp-2," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, 2004.