Optimal Trajectory Generation for Manipulator Robots under Thermal Constraints

Matthieu Guilbert, Pierre-Brice Wieber, Luc Joly

Optimal Trajectory Generation for Manipulator Robots under Thermal Constraints

Matthieu Guilbert
Stäubli Robotics
Faverges, France 74210
Email: matthieu.guilbert@inrialpes.fr

Pierre-Brice Wieber
INRIA Rhône Alpes
Montbonnot, 38330
Email: pierre-brice.wieber@inrialpes.fr

Luc Joly
Stäubli Robotics
Faverges, France 74210
Email:l.joly@staubli.com

Abstract

We propose here to deal with the optimization of velocity profiles of manipulator robots with a minimum time criterion subject to thermal constraints. This paper deals with the real impact of thermal limitations on optimal velocity profiles and the methods to calculate the corresponding optimal trajectories. We first calculate analytically the optimal solution in a simple case in order to verify the validity of the numerical algorithm and also to present a general methodology to calculate optimal trajectories in robotics using results from the theory of calculus of variations and not from the theory of optimal control. We derive then a numerical algorithm based on the discretization of the time law through an interpolation with non uniform cubic splines. This algorithm shows robust and efficient convergence properties and the trajectories thus generated were executed successfully on a Stäubli Rx90.

Index Terms

Thermal modelization, Trajectory generation, Calculus of variations, Optimal Control, non-uniform cubic splines.

Optimal Trajectory Generation for Manipulator Robots under Thermal Constraints

I. Introduction

The programming of industrial robots is generally based on the operator's experience, regardless of the system's exact dynamics or relevant optimization criteria. Due to the complexity of robots and manufacturing systems, even highly qualified operators can only reach a limited level of efficiency. A better exploitation of the performances of robots integrated in manufacturing systems can only be achieved then by using computer aided optimization methods. Works on trajectory optimization usually focus on generating trajectories with minimum time or minimum energy criteria subject to the actuators' limitations. These limitations are often maximum authorized velocity, acceleration or torque as in the references [1] [2] [3] [4] [5]. These limitations don't really reflect all the real limitations of a robot which are also overheating, wearing and breaking. We propose here to deal with the optimization of velocity profiles with a minimum time criterion subject to thermal constraints. After deriving a thermal model of a Stäubli robot in section II, we calculate analytically the corresponding optimal profile in a simple case in section III, then we derive a numerical algorithm to deal with the general case in section IV. Numerical results of this algorithm are compared then in section V to the analytical solution, and tested on a real industrial robot.

II. Temperature prediction for robotic systems

Minimizing the duration of robotic applications usually induces stonger demands on the mechanical and electrical parts. Wearing and overheating are some of the classical consequences of these demands, and we will focus here on the increase of temperature. Since a high temperature can cause damages, this increase of temperature must be controlled and since the rise of temperature is a slow phenomenon (it can take more than 5 hours to stabilize), sensors can't be used to measure the stabilized temperature, reason why we need a thermal model to predict it. Since most robotic applications are cyclic, it is possible to derive a model which predicts the stabilized temperature corresponding to a given cycle once this cycle is known. To predict this temperature, the reference [6] proposes to take into account the loss by Joule effect in the motors. Only the reference [7] takes into account both the loss of the motors and the loss in the mechanical parts of actuators. Heat tranfers between gears, motors, and other parts of the robot can be described by conduction, convection and radiation phenomena [8], but in practice, these three transfer modes are simultaneous and not easy to separate: their study is therefore often empirical.

A thermal model will be derived in section II-A which only takes into account the conduction phenomenon (this is the major heat transfer in our system), then the validity of the model will be tested in section II-B.

A. Thermal model of the system

In order to predict the temperature of the robot, we will predict in fact the temperature at different points considered to be representative of the system, from a thermal point of view. Moreover, in an industrial robot, the articulations are often enclosed in casings, there exist therefore strong thermal coupling between actuators. Specifically, we will identify our model on a Stäubli Rx90 in which the actuators are enclosed by pairs. Six heat sources can be distinguished then, 3 by actuator:

$$\begin{pmatrix} \Delta T_1 \\ \Delta T_2 \end{pmatrix} = A \begin{pmatrix} I_1 \\ I_2 \end{pmatrix} + B \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} + \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} \quad (1)$$

with ΔT_j the elevation of temperature of the representative point j, A and B two constant matrices which represent the thermal resistances in the different materials, γ_1 and γ_2 two constant vectors representing the constant loss of the coils in the brakes, I_1 and I_2 representing the loss by Joule effect of the coils depending on the consumed current, V_1 and V_2 representing the loss due to friction in the gears depending on the velocity, with:

$$I_j = \frac{1}{t_f}\int_0^{t_f} i_j(t)^2 dt, \text{ and } V_j = \frac{1}{t_f}\int_0^{t_f} v_j(t)^2 dt$$

where $i_j(t)$ is the current load and $v_j(t)$ is the rotation velocity of the motor.

B. Identification and validation of the model

To identify the constants in this model for a given robot, 100 different trajectories have been executed on this robot with different current and velocity mean values (I_j and V_j). For each trajectory, the stabilized temperature is measured after 6 hours of execution. The parameters can be identified then with a least squares procedure. The reliability of the model can be evaluated by studying the error prediction of this model and by calculating the confidence intervals of the identified parameters.

The general tendancy of the prediction error can be seen in figure 1 to show non-linear behaviours for low currents and velocities. This obviously means that not all the physical phenomena have been modeled in equation (1). However despite the empirical design of this model, it gives good predictions with a mean error of 5%.

Confidence intervals of the thermal resistances appearing in the matrices A and B have also been calculated. Statistically speaking, we are 95% sure that the parameters are in the

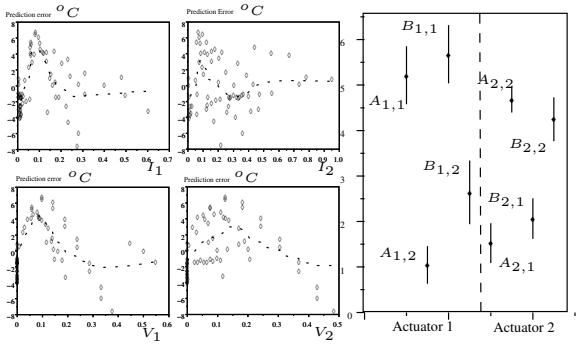

Fig. 1. Left: Prediction error according to I_1, I_2, V_1, V_2. Right: Confidence intervals of the elements of the matrix A and B

intervals represented in figure 1. Since the intervals don't cross the zero axis, all the identified parameters appear to have an influence on the predicted temperature, so all of them need to be present in equation (1).

III. Trajectory optimization: analytical solution in a simple case

A. Definition of the problem

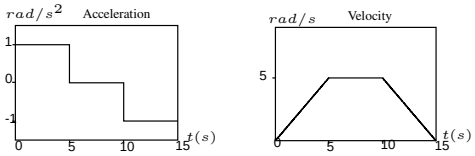

Fig. 2. Acceleration and velocity for a BANG-zero-BANG optimal trajectory

Minimum time control problems in robotics are classicaly solved with the help of BANG-BANG or BANG-zero-BANG solutions [9]. Such solutions appear when bounds are expressed on the control variables. Figure 2 shows such an optimal solution when bounds are expressed on the acceleration and the velocity. They can be derived with the help of the Maximum Principle of Pontryagin [9], and they present jumps of the control variables from one bound to another, what explains their name. In our case, the bounds are not directly expressed on the control variables but on the temperature, therefore such profiles are not correct answers to our problem. To find an analytical solution when the bounds are expressed on the temperature, we consider a simple movement of a horizontal axis of our robot. By consequence, the system's dynamics presents no gravity effects, a constant inertia, no centrifugal and coriolis forces, what leads to the simple dynamic model:

$$\Gamma = J\ddot{q} + F_v\dot{q} + F_s \tag{2}$$

where Γ is the articular torque, $\ddot{q}$ the acceleration, $\dot{q}$ the velocity, J the inertia of the whole system and F_v and F_s the viscous and Coulomb friction (a constant here since we will consider a trajectory where the sign of the velocity does not change). In terms of function to minimize and constraints, the problem we need to solve here is:

$$\min\ t_f = \int_0^{t_f} 1\,dt \tag{3}$$

subject to:

$$\frac{1}{t_f}\int_0^{t_f} a(J\ddot{q} + F_v\dot{q} + F_s)^2 + b\,\dot{q}^2\,dt \;=\; T_{max} - c \tag{4}$$

$$\int_0^{t_f} \dot{q}\,dt \;=\; q_f - q_0 \tag{5}$$

with q_0 and q_f the initial and final position of the axis, a, b, c the constants of the thermal model (always strictly positive) and T_{max} the maximal authorized temperature. Note that the constraint (4) is an equality instead of an inequality since we consider that it will be an active constraint for this trajectory.

This is a minimization problem subject to isoperimetric constraints in which the end point is not fixed [10]. Lagrange multipliers λ_1 and λ_2 need to be introduced and we want to find the saddle points of:

$$\int_0^{t_f} \left(1 + \frac{\lambda_1}{t_f}(a(J\ddot{q} + F_v\dot{q} + F_s)^2 + b\dot{q}^2) + \lambda_2\dot{q}\right) dt \tag{6}$$

With:

$$F(t,\dot{q},\ddot{q}) = 1 + \frac{\lambda_1}{t_f}(a(J\ddot{q} + F_v\dot{q} + F_s)^2 + b\dot{q}^2) + \lambda_2\dot{q},$$

the necessary condition in this case is the Euler-Lagrange differential equation:

$$\frac{\partial F(t,\dot{q},\ddot{q})}{\partial \dot{q}} - \frac{d}{dt}\left(\frac{\partial F(t,\dot{q},\ddot{q})}{\partial \ddot{q}}\right) = 0 \tag{7}$$

with boundary conditions:

$$\begin{cases} \dot{q}(0) = 0, \\ \dot{q}(t_f) = 0. \end{cases} \tag{8}$$

Since t_f is not fixed, the following transversality condition must hold:

$$F(0) - \ddot{q}(0)\frac{\partial F}{\partial \ddot{q}}(0) = 0. \tag{9}$$

B. Calculation of the optimal velocity profile

The first step to calculate the optimal velocity profile is to solve the Euler-Lagrange equation (7) which becomes after several calculations:

$$\frac{-2\lambda_1 aJ^2}{t_f}\dddot{q} + \frac{2\lambda_1(aF_v^2 + b)}{t_f}\dot{q} + \frac{2a\lambda_1 F_s F_v}{t_f} + \lambda_2 = 0 \tag{10}$$

or in another way

$$\dddot{q} - r\dot{q} = C(\lambda_1,\lambda_2) \tag{11}$$

with $r = \dfrac{aF_v^2 + b}{aJ^2}$ and $C(\lambda_1,\lambda_2) = \dfrac{F_s F_v}{J^2} + \dfrac{\lambda_2}{2a\lambda_1 J^2}$.

Note that if $\lambda_1 = 0$, the Euler-Lagrange equation (10) gives $\lambda_2 = 0$, and the problem (6) degenerates: we can fairly consider therefore that λ_1 is different from zero.

Integrating the equation (11) leads to solutions of the form:

$$\dot{q}(t) = \zeta\,\sinh(\sqrt{r}\,t) + \nu\,\cosh(\sqrt{r}\,t) - \frac{C(\lambda_1,\lambda_2)}{r}. \tag{12}$$

Now, the constants ζ, ν, λ_1, λ_2 and t_f can be determined with the boundary conditions (8), the constraints (4)-(5) and the transversality condition (9). In fact, we don't need to calculate the Lagrange multipliers λ_1 and λ_2 explicitely, and determining ζ, ν and C is enough to define the optimal solution:

$$\begin{aligned}\zeta(t_f) &= (q_f - q_i)\frac{\sqrt{r}(\cosh(\sqrt{r}\,t_f) - 1)}{-2\cosh(\sqrt{r}\,t_f) + 2 + t_f\sqrt{r}\sinh(\sqrt{r}\,t_f)},\\ \nu(t_f) &= -(q_f - q_i)\frac{\sqrt{r}\sinh(\sqrt{r}\,t_f)}{-2\cosh(\sqrt{r}\,t_f) + 2 + t_f\sqrt{r}\sinh(\sqrt{r}\,t_f)},\\ C(t_f) &= r\nu.\end{aligned} \tag{13}$$

Note that these constants are defined with respect to the final time t_f which is computed by solving numerically the constraint (4).

The figure 3 shows such an optimal velocity profile on a Stäubli Rx90 for a movement of its first axis from -2.26 rad to $+2.26$ rad. After solving the equation (4), we find $t_f = 1.38s$. Since the jerk appears in the necessary condition (11), the acceleration is continuous and differentiable everywhere, what helps to avoid vibrations. Note that the velocity on the boundaries of the trajectory can be fixed at will through the boundary conditions (8), but the acceleration on these boundaries is unfortunately imposed by the shape of the solution.

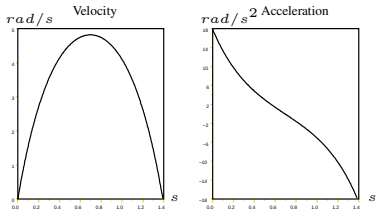

Fig. 3. Optimal velocity and acceleration profiles

IV. Trajectory optimization: numerical solution in the general case

In the general case the simple dynamics (2) of the previous section turns into [11]:

$$\Gamma = M(q)\ddot{q} + N(q,\dot{q})\dot{q} + G(q) + H(\dot{q}) = f(q,\dot{q},\ddot{q}), \tag{14}$$

with $M(q)$ the inertia matrix, $N(q,\dot{q})$ the matrix of centrifugal and coriolis effects, $G(q)$ the gravity effects and $H(\dot{q})$ the friction. This dynamics is much more complex than the previous one and an analytical solution to this general trajectory optimization problem might be out of reach. We are going therefore to look for a numerical solution.

The problem of trajectory generation is generally considered in robotics as a problem of optimal control where the state equation is in the form:

$$\ddot{q} \quad = \quad F(q,\dot{q},\Gamma). \tag{15}$$

Expressing the dynamics of the system in this form implicitely introduces then the idea to solve this dynamics in order to obtain the trajectory $(q(.),\dot{q}(.))$ for a given command $\Gamma(.)$. The gradient of the cost function needs therefore to be calculated with adjoint variables methods [12]. There exist then two general classes of methods to solve this kind of problem [13] [14]: indirect methods based on the application of the Maximum Principle of Pontryagin (using adjoint equations), generally considered to be very precise but very sensitive to initial conditions, and direct methods based on the discretization of both the trajectory $(q(.),\dot{q}(.))$ and the command $\Gamma(.)$ (without using adjoint equations), leading to a classical non linear optimization problem and considered to be less precise but less sensitive to initial conditions.

However, it may be an error to express the dynamics (14) as in equation (15): with dynamics expressed as in (14) the problem of trajectory generation appears directly as a problem of calculus of variations. The dynamics (14) can be directly integrated in the different constraints of our problem which becomes then:

$$\min_{(q,\dot{q})} t_f$$

$$\frac{1}{t_f}\int_0^{t_f} (Af(q,\dot{q},\ddot{q})^2 + B\dot{q}^2)\,dt + \gamma + T_{amb} - T_{max} \leq 0.$$

There are two ways of solving this problem: it is possible to calculate the optimality conditions similarly to what we've done in the previous section, but that may lead to set of complex non-linear ordinary differential equations with boundary conditions that may not be easy to solve. Another way is to consider a discrete approximation of the trajectory $q(t)$:

$$\tilde{q}(t) \quad = \quad S(p,t) \tag{16}$$
$$\dot{\tilde{q}}(t) \quad = \quad \dot{S}(p,t) \tag{17}$$
$$\ddot{\tilde{q}}(t) \quad = \quad \ddot{S}(p,t) \tag{18}$$

with S an interpolation function at least of class C^2, and p the parameters of the interpolation. The dynamics (14) takes then the following form:

$$\tilde{\Gamma}(t) = f(S(p,t),\dot{S}(p,t),\ddot{S}(p,t),t) = \tilde{f}(p,t) \tag{19}$$

Discretizing the trajectory $q(t)$ has several advantages: a small number of discretization parameters (the same ones for q, $\dot{q}$, $\ddot{q}$ and Γ), a finite dimensional problem implying a simpler calculation of the gradients, and the use of improved non-linear optimization algorithms such as Sequential Quadratic Programs (SQP) which are presently the most efficient algorithms for such problems. Note that similar choices have been tested in [13] with good results.

A. Definition of the optimization problem

We are not interested here in optimizing the geometric path of the trajectory and we focus rather on the optimization of the velocity profile of the trajectory along a specified geometric path. The first step consists therefore in defining the curvilinear abscissa $\lambda : [0,t_f] \rightarrow [0,1]$ (also called the time law) and the geometry function $Q : [0,1] \rightarrow \mathbb{R}^6$ which transforms this

curvilinear abscissa into an articular position, both functions being at least of class C^2 such that:

$$q(.) = Q(\lambda(.)) \quad (20)$$

$$\dot{q}(.) = \frac{dQ}{d\lambda}(\lambda(.))\dot{\lambda}(.) \quad (21)$$

$$\ddot{q}(.) = \frac{d^2Q}{d\lambda^2}(\lambda(.))\dot{\lambda}^2(.) + \frac{dQ}{d\lambda}(\lambda(.))\ddot{\lambda}(.) \quad (22)$$

The second step consists in discretizing this curvilinear abscissa: there exist various techniques of discretization but the most usual are polynomial splines [13] [5] [2] [15]. Since the time law must be at least of class C^2, we will use cubic splines and more precisely we will compute them here as in [2]. The spline is defined as shown in figure 4, with $\lambda(t_i) = \Lambda_i$ for $1 \leq i \leq n$, $t_1 = 0$ and $\Lambda_1 = 0$, $t_n = t_f$ and $\Lambda_n = 1$. Since $t_n = t_f$ is variable whereas $\Lambda_n = 1$ is fixed, we will consider that all the $\{t_i\}_{(1\leq i\leq n)}$ are variable whereas all the $\{\Lambda_i\}_{(1\leq i\leq n)}$ are fixed, leading to a non-uniform spline.

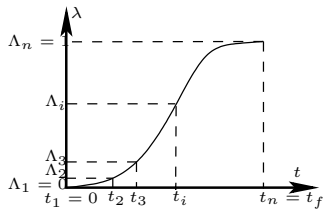

Fig. 4. Discrete time law

Following [2], our strategy to define this cubic spline is to impose the continuity of the velocity and the acceleration at the nodes t_i, and to fix the velocity on the boundaries. We use the intermediate variables $\{\ddot{\Lambda}_i\}_{(1\leq i\leq n)}$ to fix the acceleration at each knot. Each of the cubic polynomials $\lambda_j(t) = \lambda(t)$ for $t \in [t_j, t_{j+1}]$ constituting the spline can be written in terms of the $\ddot{\Lambda}_j$ and the h_j:

$$\lambda_j(t) = \frac{(t_{j+1}-t)^3}{6h_j}\ddot{\Lambda}_j + \frac{(t-t_j)^3}{6h_j}\ddot{\Lambda}_{j+1} + \left(\frac{\Lambda_{j+1}}{h_j} - \frac{h_j\ddot{\Lambda}_{j+1}}{6}\right)(t-t_i) + \left(\frac{\Lambda_j}{h_j} - \frac{h_j\ddot{\Lambda}_j}{6}\right)(t_{i+1}-t).$$

The continuity of the velocity is satisfied then by solving a linear system that leads to the computation of the $\{\ddot{\Lambda}_i\}_{(1\leq i\leq n)}$:

$$C(h)\ddot{\Lambda} = d(h) \quad (23)$$

with:

$$C(h) = \begin{pmatrix} 2h_1 & h_1 & & & 0 \\ h_1 & 2(h_1+h_2) & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & & & h_{n-1} & 2h_{n-1} \end{pmatrix}$$

$$d(h) = \begin{pmatrix} 6\left(\frac{\Lambda_2-\Lambda_1}{h_1} - v_1\right) \\ 6\left(\frac{\Lambda_3-\Lambda_2}{h_2} - \frac{\Lambda_2-\Lambda_1}{h_1}\right) \\ \vdots \\ 6\left(\frac{\Lambda_n-\Lambda_{n-1}}{h_{n-1}} - \frac{\Lambda_{n-1}-\Lambda_{n-2}}{h_{n-2}}\right) \\ 6\left(v_n - \frac{\Lambda_n-\Lambda_{n-1}}{h_n}\right) \end{pmatrix}$$

with $h_i = t_{i+1} - t_i$ the time intervals between knots, and $h = (h_1, h_2, ..., h_{n-1})^T$ the new set of parameters for the optimization procedure. After solving the linear system (23), the spline is totally determined by the $\{\Lambda_j\}_{1\leq j\leq n}$, the velocity on the boundaries v_1 and v_n and the time intervals $\{h_i\}_{1\leq i\leq n-1}$. Note that the matrix C is non singular here since it is diagonally dominant, and there are efficient numerical methods to invert such tridiagonal matrices.

Now, the cost function can be simply expressed by a linear function of the parameters, $t_f = \sum_{i=1}^{n-1} h_i$. The thermal model (1) leads to the following constraint:

$$\frac{A}{t_f}\int_0^{t_f}\Gamma^2(t)\,dt + \frac{B}{t_f}\int_0^{t_f}\dot{q}^2(t)\,dt + \gamma + T_{amb} - T_{max} \leq 0, \quad (24)$$

with T_{amb} the ambiant temperature and T_{max} the maximal authorized temperature. The integral temperature constraint (24) can be estimated then with a trapezoidal approximation, precise enough in our case as we will see in the next section:

$$\frac{1}{2t_f}\sum_{j=0}^{n-1}(A(\Gamma(t_j)^2 + \Gamma(t_{j+1})^2)h_j + B\left(\dot{q}(t_j)^2 + \dot{q}(t_{j+1})^2\right)h_j) + \gamma + T_{amb} - T_{max} \leq 0. \quad (25)$$

B. Optimization algorithm

The original trajectory optimization problem has been transformed into a minimization of a linear function subject to non linear constraints. Newton methods such as Sequential Quadratic Programming (SQP) are presently the most efficient ones to solve this kind of problem [16]. These methods need the gradients of both the cost function and the constraints. The gradients can be calculated numerically with finite differences methods, but that severely impedes the convergence of the SQP. Symbolic or automatic differentiation methods [17] can be used also, but in our problem the geometric path is fixed and the dynamics (14) can be formulated very simply as [4] [3] [18]:

$$\Gamma = m(\lambda)\ddot{\lambda} + c(\lambda)\dot{\lambda}^2 + f(\dot{\lambda}) + g(\lambda) \quad (26)$$

where m, c, g and f are vectors which respectively represents inertias, centrifugal and coriolis effects, gravity and friction. Since the constraints are evaluated at points $\{\Lambda_j\}_{1\leq j\leq n}$ which don't depend on the variables $\{h_i\}_{1\leq i\leq n-1}$, the gradient of the dynamic model is:

$$\frac{\partial\Gamma_j}{\partial h_k} = m(\Lambda_j)\frac{\partial\ddot{\Lambda}_j}{\partial h_k} + 2\,c(\Lambda_j)\frac{\partial\dot{\Lambda}_j}{\partial h_k}\dot{\Lambda}_j + \frac{\partial f}{\partial h_k}(\dot{\Lambda}_j) \quad (27)$$

Since we suppose that we optimize trajectories where the velocity has always the same sign, we can rely on a simple expression of the gradient of the Coulomb and viscous friction:

$$\frac{\partial f}{\partial h_k}(\dot{\Lambda}_j) = F_v\frac{dQ}{d\lambda}\frac{\partial\dot{\Lambda}_j}{\partial h_k} \quad (28)$$

The calculation of the gradient of the dynamic model amounts then to calculating $\frac{\partial \dot{\Lambda}_j}{\partial h_k}$ and $\frac{\partial \ddot{\Lambda}_j}{\partial h_k}$, what is trivial here through equation (23).

The last important point is the initialization of the optimization process: to help its convergence, we must choose a first iterate as close as possible to the optimal solution and satisfying the constraints. From a practical point of view, we generate a BANG-BANG profile, and we use a dichotomy technique to improve the first iterate by testing the constraints. The duration of the movement is stretched if any constraint is violated, compressed otherwise.

V. Numerical and Experimental Tests of the Optimal Trajectory Generator

To validate the numerical algorithm presented in the previous section, we have chosen to apply it to the same geometric path as in section III, a movement of the first axis of a Stäubli Rx90 from -2.26 rad to $+2.26$ rad. To solve the nonlinear optimization problem, we use the Feasible Sequential Quadratic Programming (FSQP) algorithm [19].

A. Numerical results

The numerical algorithm described in the previous section converges to the solution showed in figure 5, with an optimal time $t_f = 1.35s$. We can observe that it is very close to the analytical solution found in section III. The very small difference between these two solutions can be identified to be solely due to the discretization process (16)-(18). For the same reason, the approximate computation in (25) of the constraint (24) appears to slightly underestimate the limiting temperature constraint in this specific case, allowing a faster solution here than the truly optimal solution described in section III, with only $1.35s$ of cycle time rather than $1.38s$ in section III.

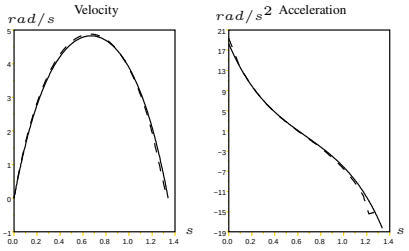

Fig. 5. Analytical (plain curve) and numerical (dashed curve) optimal profiles

More generally, this algorithm has been observed to converge properly as soon as the constraints are satisfied from the beginning of the optimization process, as soon as the first iterate is a feasible point. This condition appeared to be of great importance to obtain this convergence: the initialization described at the end of section IV-B appears therefore to be a key point for the robustness of the whole numerical algorithm.

B. Experimental results

The trajectory generated by the algorithm was executed on an industrial Stäubli Rx90 robot with a CS8 controller without filtering, in spite of the discontinuities of the acceleration at the boundaries. The stabilized temperature measured after 6 hours reaches the prescribed limit with only 3% of error. Both the dynamics and the models appear therefore to allow very precise predictions. Note however that oscillations appear in figure 6 which have not been predicted. These oscillations are due to the discontinuities of the acceleration on the boundaries that excite the vibration modes of the robot (fixing the acceleration at the boundaries might solve this problem).

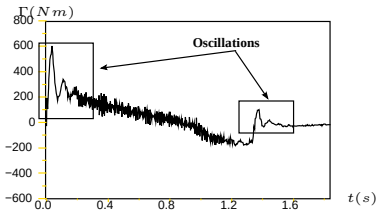

Fig. 6. Measured torque

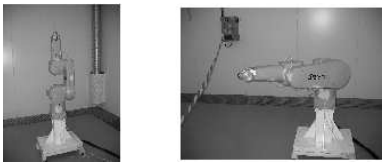

Fig. 7. Upright configuration on the left and flat configuration on the right

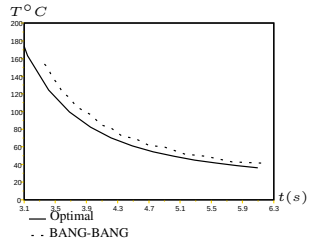

Fig. 8. Evolution of the optimal cycle time with respect to different temperature limits

A comparison between the BANG-BANG and the optimal profiles has been realized for a minimum and a maximum inertia configuration, respectively the upright and flat configurations illustrated in figure 7. This comparison led to the following result:

Configuration	Profile type	Temperature	Cycle Time
Flat	BANG-BANG	$106°C$	3.76s
Flat	Optimal	$97°C$	3.68s
Upright	BANG-BANG	$109°C$	3.05s
Upright	Optimal	$98°C$	2.98s

The optimal velocity profile appears to be 5% better than the BANG-BANG one from the point of view of cycle time, with a temperature increase lower by 8%. We can observe as well in figure 8 that in the case of different temperature limits, for a given cyclic movement, the optimal velocity profile

always gives results from 5% to 10% better than the BANG-BANG profile. It is obvious then that specifically optimizing the trajectories of manipulator robots with respect to their real physical limitations allows a superior usage of their real capacities, with faster execution times together with a milder management of these limitations.

VI. Conclusion

Trajectory optimization in robotics usually deals with constraints which don't really reflect the true limitations of a robot, maximal velocities and accelerations which have no direct connection to overheating, for example. This paper deals therefore with the real impact of thermal limitations on optimal velocity profiles and the methods to calculate the corresponding optimal trajectories. We first calculate analytically the optimal solution in a simple case: more than a way to verify the validity of the numerical algorithm, this analytical calculation presents a general methodology to calculate optimal trajectories in robotics. An important point of section III is actually the use of results from the theory of calculus of variations and not from the theory of optimal control. The beginning of section IV discusses why trajectory optimization in our case is better posed as a problem of calculus of variations rather than a problem of optimal control, what is the direct generalization of section III. After this disussion we derive a numerical algorithm based on the discretization of the time law through an interpolation with non-uniform cubic splines. This algorithm has shown robust and efficient convergence properties and the trajectories thus generated were executed successfully on a Stäubli Rx90 with a CS8 controller. It appeared then obviously that specifically optimizing the trajectories of manipulator robots with respect to their real physical limitations allows a superior usage of their real capacities, with faster execution times together with a milder management of these limitations.

References

[1] Y. Bestaoui, "On line reference trajectory definition with joint torque and velocity constraints," *The International Journal of Robotics Research*, 1992.

[2] A. D. Luca, L. Lanari, and G. Oriolo, "A sensitivity approach to optimal spline robot trajectories," *Automatica*, 1991.

[3] J. Hollerbach, "Dynamic scaling of manipulator trajectories," *Journal of Dynamic Systems, Measurement, and Control*, 1984.

[4] J. Bobrow, S. Dubowsky, and J. Gibson, "Time optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, 1985.

[5] C.-S. Lin, P.-R. Chang, and J. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Transactions on Automatic Control*, 1983.

[6] F. Ltd, "Displaying method for duty of industrial robot," *Japan Patent JP7087787*, 1995.

[7] D. Corp., "Motor controller," *Japan Patent JP2001292586*, 2001.

[8] J. Taine and J.-P. Petit, *Transferts thermiques, Mécanique des fluides anisothermes*. Dunod, 1998.

[9] A. Bryson and Y.-C. Ho, *Applied Optimal Control, Optimization, Estimation and Control*. Taylor and Francis, 1975.

[10] E. Pinch, *Optimal Control and the calculus of Variations*. Oxford Science Publications, 1993.

[11] W. Khalil and E. Dombre, *Modélisation, identifi cation et commande des robots*. Hermes Science Publications, 1999.

[12] C. Lemaréchal, "Optimisation," *Les techniques de l'ingénieur : Automatique*, 1995.

[13] O. von Stryck, "Optimal control of multibody systems in minimal coordinates," *Zeitschrift fur Angewandte Mathematik und Mechanik 78, Suppl 3*, 1998.

[14] J. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, 1997.

[15] S. Thompson and R. Patel, "Formulation of joint trajectories for industrial robots using b-splines," *IEEE Transactions on industrial Electronics*, 1987.

[16] J.-F. Bonnans, J.-C. Gilbert, C. Lemaréchal, and C. Sagastizabal, *Numerical Optimization : Theoritical and Practical Aspects*, 2003.

[17] A. Durrbaum, W. Klier, and H. Hahn, "Comparison of automatic and symbolic differentiation in mathematical modeling and computer simulation of rigid-body systems," *Multibody Systems Dynamics*, 2002.

[18] L. Žlajpah, "On time optimal path control of manipulators with bounded joint velocities and torques," *Proceedings IEEE Int. Conf. on Robotics and Automation*, 1996.

[19] C. Lawrence, J. Zhou, and A. Tits, "User's guide for cfsqp version 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, Tech. Rep., 1997.