Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Optimal Control for Autonomous Task Execution

Debora Botturi and Paolo Fiorini
Department of Computer Science
University of Verona, Italy
{botturi,fiorini}@metropolis.sci.univr.it

Abstract— **The research described in this paper addresses the problem of automatic execution of a robotic task. The objective is to develop a control methodology that can be used to give autonomy to a robotic device during the execution of a specific task. In particular we focus on the application domain of robotic surgery and we want to explore the possibility of adding autonomous capability to the robotic device performing the surgery, to be of help to the surgeon. In this context, by "task" we mean a small sequence of coded surgical gestures, that is well described in the medical literature and that can, potentially, be described in algorithmic form. We first model the task with a suitable Hybrid System, then we compute the nominal controls by minimizing various performance indices, and finally we define a quality measure to provide feedback during real time execution. We model a complex task with a hybrid automaton, whose elementary states represent distinct actions in the task, and we account for uncertainty with appropriate state transitions. The desired behavior is represented by the optimal trajectory computed off-line, whereas the on-line compensation aims at zeroing trajectory errors and the cost of the jumps between states. We conclude the paper with some simulation and experimental results proving the feasibility of this approach.**

I. INTRODUCTION

Due to the complex and diverse nature of a generic surgical task, the use of a single control law for the whole task is not feasible. Therefore, we propose to use an approach based on *Hybrid System* theory, whereby a task is modeled as a sequence of states each endowed with its own controller.

The work described in this paper addresses the problem of controlling such a hybrid system. Our model for control computation is the hybrid automaton. A hybrid automaton is a state machine with transitions between states that are governed by discrete logic decision.

The properties of a control algorithm are stated in terms of satisfying stability. However, in our case stability analysis is not enough to ensure either good performance or task termination. Our purpose is then also to determine a set of parameters that can be used to determine the performance of the control system and whether or not the task is likely to terminate. The main difficulty of proposing a new control method for an application such as robotic surgery, is to ensure complete and total safety of the procedure. This implies the need to address a number of aspects, which, in a less demanding application, may be overlooked in a first development. In fact, this work will deal with uncertain and variable environments, as represented by the different patient anatomies, with the true complexity of a surgical operation, without imposing unrealistic simplifications. The work will also deal with the presence and the interaction of humans to provide input to the system, and with the difficulties of implementing the results.

In the following we will present an overview on autonomous task execution using a Hybrid System representation in Section II. In Section III, Hybrid Optimal Control of the Hybrid system is introduced. Optimum Task Control and its mathematical background is presented in Section IV. In Section V we present the formulation of the problem addressed in this work and in Section VI we discuss the algorithm used in the nominal trajectory computation. In Sections VII and VIII respectively some simulation and experimental results are presented, and Section IX concludes the paper with some consideration on the results presented.

II. AUTONOMOUS TASK EXECUTION

Hybrid Systems are starting to be used in the representation, validation and execution of complex tasks. Hybrid systems permit to encode the structure of a complex task with an appropriate sequence of continuous and discrete states; maintain the hierarchical structure of the standard robotic control architecture by encapsulating the continuous variables in the hybrid states; and analyses and test the overall system performance by using formal tools derived from Computer Science and Automatic Control.

In the literature, automatic execution of robotic tasks is mainly studied for mobile mobile robots. In [1] the task to be carried out by the robot is well defined and various phases of the task are controlled by the underlying hybrid system monitor. In [2], [3] a hybrid automaton is used to model the cooperation of multiple mobile robots to perform a coordinated manipulation.

Autonomous capability in manipulation are described in the integration of vision, grasp planning and execution [4].

Another area of application of Hybrid System theory, and of great potential impact, is the control of the navigation and interaction of autonomous vehicles. An example is [5], in which the focus is on intelligent multi agent systems that eventually will replace centralized control systems, as in the case of air traffic management, or will enhance human resources, as in the case of automatic vehicle control. The Hybrid System framework is ideally suited for autonomous, or semi-autonomous, agent control. In fact, at the continuous level, each agent chooses its own optimal strategy, while discrete coordination is used to solve conflicts.

This brief summary shows that Hybrid System are starting to be used to control complex tasks. However the equivalent feedback signal forcing the task to its completion is not fully specified. In Section V we show that a distance between a nominal trajectory and the actual execution can be used for this purpose.

III. Hybrid Optimal Control

In the context of Hybrid Systems, the computation of the optimal solution involves the selection of a path among the possible states of the hybrid system and the computation of the optimal times to jump from one state to the next.

Controlling the switching times, when possible, and choosing among several possible states, whenever such choices are available, gives rise to a rich class of optimal control problems. This has motivated efforts to extend classical optimal control principles [6] and to apply dynamic programming techniques [7], [8].

The solutions of the Hybrid Optimal Control Problem (HOCP) are deterministic open-loop trajectories.

The numerical solution of closed-loop hybrid feedback control problems, however, is at even a much earlier stage and the primarily finite-element based solution strategies that have been presented for their solution [8] cannot readily handle nonlinear systems of more than three dimensions due to the well-known curse of dimensionality. However, the approach is not applicable to our class of HOCPs with nonlinear dynamics equations subject to nonlinear constraints. For these reasons we adapt Pontryagin's Maximum Principle to the specific requirements of Hybrid Systems as discussed in the next Section.

IV. Optimal Task Control

In this Section we present the approach we propose to control the autonomous execution of a complex task. The task that we consider is a puncturing task, using the model shown in Figure 1. This task, although simple, has the characteristics of a complex task, since it requires different control modes and involves various logical conditions. To guarantee the correct task execution we define as a global error the difference between a nominal trajectory and the value of the trajectory at run time. The nominal trajectory of the state variables is computed off-line using optimization techniques. At execution time, uncertainties are compensated by a regulator customized for each subtask that eliminates the error on the continuous states and on the jump conditions. Roughly speaking, the computation of the nominal trajectory is an instance of optimal control problem, i.e. to drive the system to a desirable state while minimizing a cost function that depends on the path followed. It typically involves a *terminal cost* (depending on the terminal state), an *integral cost* accumulated along continuous evolution, and a series of *jump cost* associated with discrete transitions. This is a classical problem for continuous systems, extended more recently to discrete systems [9], and to a class of hybrid systems with simple continuous dynamics. The approach has been extended to general hybrid systems using a dynamic programming formulation [6], and a variational formulation, extending Pontryagin Maximum Principle [10].

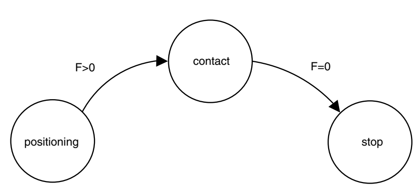


Fig. 1. State transition diagram.

We first focalize our attention on the control problem and the optimization techniques for determining the nominal control $\mathbf{u}(t)$.

We consider the problem of selecting a continuously differentiable function $x: [t_0, t_f] \rightarrow \Re$ to minimize the cost function

$$J(x) = \int_{t_0}^{t_f} \phi[x(t), \dot{x}(t), t] dt \tag{1}$$

with respect to the set of real-valued, continuously differentiable functions u on the interval $[t_0, t_f]$. Such functions are referred to as *admissible trajectories*. Throughout this Section, we assume that ϕ is continuous in $x, \dot{x}$, and t and has continuous partial derivatives with respect to x and $\dot{x}$.

A. State Nominal Trajectory

The computation of the nominal trajectory can be formulated mathematically as an optimization problem in the following way. Find the control $u(t) \in \mathbf{U}$ in $t_0 \leq t \leq t_f$ that minimizes the performance index J:

$$J = \theta[\mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} \phi[\mathbf{x}(t), \mathbf{u}(t), t] dt \tag{2}$$

subject to:

- **kinematic constraints**:

$$\text{initial manifold:} \quad \Gamma[\mathbf{x}(t_0), t_0] = \mathbf{0} \tag{3}$$

$$\text{terminal manifold:} \quad \Omega[\mathbf{x}(t_f), t_f] = \mathbf{0} \tag{4}$$

- **dynamic constraints**:

$$\text{systems dynamics:} \quad \dot{\mathbf{x}} = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] \tag{5}$$

$$\text{admissible controls:} \quad \mathbf{u}(t) \in \mho, t \in [t_0, t_f], \mho \subset \Re^m$$

$$\mathbf{g}[\mathbf{u}, t] \geq \mathbf{0} \tag{6}$$

$$\text{state constraints:} \quad \mathbf{x}(t) \in \mathbf{X}, t \in [t_0, t_f], \mathbf{X} \subset \Re^n$$

$$\mathbf{h}[\mathbf{x}(t), t] \geq \mathbf{0} \tag{7}$$

In the context of hybrid systems the initial and terminal manifolds represent the boundaries of the states on which the system must perform a jump. Clearly in the optimal case the cost of the jump is zero as normally assumed in the literature of HOCP, however this is not the case during the execution and the cost of jumps must be accounted for in the performance index. We consider the problem of determining an admissible control function $\mathbf{u}$ in order to minimize the criterion function of (2) subject to these constraints where each component of $\mathbf{h}$ is assumed to be continuously differentiable in state space. We assume also that the terminal manifold equation is a function of terminal time, and the terminal time is unspecified. The initial time and the initial state vector are specified. Therefore the problem becomes one of minimizing the cost function (2) for the system described by (5) with $\mathbf{x}(t_0) = \mathbf{x}_0$ where, at the unspecified terminal time $t = t_f$, the terminal manifold equation (4) and the control and state constraints are satisfied.

We may convert this inequality constraint (6), where $\mathbf{g} : \Re^{m+1} \to \Re^r$, into an equality constraint by writing for each component of $\mathbf{g}$

$$y_i{}^2 = g_i[\mathbf{u}(t), t] \qquad i = 1, 2, ..., r \tag{8}$$

We adjoin, via Lagrange multipliers, constraints (4), (5), which are embedded in the Hamiltonian, and (8) to the cost function (2) to obtain

$$\begin{aligned} J \;=\; & \theta[\mathbf{x}(t_f), t_f] + \nu^T \Omega[\mathbf{x}(t_f), t_f] + \\ & \int_{t_0}^{t_f} \{\mathscr{H}[\mathbf{x}(t), \mathbf{u}(t), \lambda(t), t)] - \\ & \lambda^T(t)\dot{\mathbf{x}} - \gamma^T(t)\{\mathbf{g}[\mathbf{u}(t), t] - y^2\}\}dt \end{aligned} \tag{9}$$

where $\mathscr{H}$ is the Hamiltonian defined as

$$\mathscr{H}[\mathbf{x}(t), \mathbf{u}(t), \lambda(t), t)] = \phi[\mathbf{x}(t), \mathbf{u}(t), t] + \lambda^T \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] \tag{10}$$

There are several methods whereby we may convert the inequality constraint represented by the $s-$vector equation (7) to an equality constraint. We choose to define a new variable x_{n+1} by

$$\begin{aligned} & \dot{x}_{n+1} = \\ & [h_1(\mathbf{x}, t)]^2 H(h_1) + [h_2(\mathbf{x}, t)]^2 H(h_2) + \cdots \\ & \cdots + [h_s(\mathbf{x}, t)]^2 H(h_s) \end{aligned} \tag{11}$$

where $H[h_s(\mathbf{x}, t)]$ is a modified Heaviside step.

We now apply the Euler-Lagrange equations to the cost function (9) in order to obtain the necessary conditions for a minimum, due to McShane [11] and Pontryagin [12], and others. It is thus convenient to define a scalar function Φ, the Lagrangian for no inequality state constraint, as

$$\begin{aligned} & \Phi[\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t), \lambda(t), \gamma(t), \mathbf{y}(t), t] = \\ & \mathscr{H}[\mathbf{x}(t), \mathbf{u}(t), \lambda(t), t)] - \lambda^T(t)\dot{\mathbf{x}} - \gamma^T(t)\{\mathbf{g}[\mathbf{u}(t), t] - y^2\} \end{aligned} \tag{12}$$

From (12) it follows that the Lagrangian for the problem at hand is

$$\tilde{\Phi} = \Phi + \lambda_{\mathbf{n+1}}[\mathbf{f_{n+1}} - \dot{\mathbf{x}}_{\mathbf{n+1}}] \tag{13}$$

We write the Euler-Lagrange equations as

$$\begin{aligned} \frac{d}{dt}\frac{\partial \Phi}{\partial \dot{\mathbf{x}}} - \frac{\partial \Phi}{\partial \mathbf{x}} - \frac{\partial f_{n+1}}{\partial \mathbf{x}}\lambda_{n+1} &= 0 \\ \frac{d}{dt}\frac{\partial \Phi}{\partial \mathbf{u}} &= 0 \\ \frac{d}{dt}\frac{\partial \Phi}{\partial \mathbf{y}} &= 0 \end{aligned} \tag{14}$$

We call each piecewise continuously differentiable solution of the Euler-Lagrange equations (14) extremal trajectory of the associated variational problem. The transversality conditions for this problem are

$$\begin{aligned} \Omega[\mathbf{x}(t_f), t_f] &= \mathbf{0} \\ \mathbf{x}(t_0) &= \mathbf{x_0} \\ \frac{\partial \theta}{\partial t_f} + \frac{\partial \Omega^T}{\partial t_f}\nu + \mathscr{H} &= 0 \quad \text{for} t = t_f \\ \frac{\partial \theta}{\partial \mathbf{x}} + \frac{\partial \Omega^T}{\partial \mathbf{x}}\nu - \lambda &= 0 \quad \text{for} t = t_f \\ x_{n+1}(t_0) = x_{n+1}(t_f) &= 0 \end{aligned} \tag{15}$$

The optimization problem given by equations (14) and (15) is essentially a Multi Point Boundary Value Problem, since its solution must satisfy the terminal manifold Ω, the state and co-state constraints. The optimal solution $u_i^*(t)$ for the i-subtask is reached when the above necessary conditions of optimality, derived from Pontryagin's Maximum Principle, are satisfied. We compute the global optimal solution, i.e. nominal trajectory of the complete task

$$u^*(t) = \sum_{i=1}^{n} \sum_{t=0}^{N} u_i^*(t)$$

The optimality of the u^* is assured by null jump cost θ_i for all i. These conditions (15) can be used to check the i optimal solution because we know for each subtask i the terminal configuration manifold Ω.

In particular, they must be satisfied by the trajectory computed by the numerical method. It will be shown in Section VI, that a steepest descent algorithm computes a solution satisfying the necessary conditions and, therefore, converges to a target behavior.

V. Problem Formulation

The control problem is cast in a form compatible with Pontryagin's Principle with state-dependent control constraints. We consider the case where the terminal manifold equation is a function of the terminal time, and the terminal time is unspecified. The optimal control is computed minimizing the cost of a performance index function of the continuous state and of the jumps necessary to reach the subsequent state in the optimal configuration respect to the subtask. We iterate this concept for each subtask. By collecting the nominal trajectories computed for each state, we obtain the optimal control law for the complete task.

Remark 5.1: The continuous evolution is assured by the kinematic constraints, i.e. the configuration at time t_f belongs to the intersection of the terminal configuration manifold $\Omega[\mathbf{x}(t_f)]$ of state $i-1$ and the initial configuration manifold $\Gamma[\mathbf{x}(t_0)]$ of state i.

We use the variational approach where the terminal time is not fixed and where the control and state vectors are smooth functions.

Remark 5.2: Among state constraints we include Lyapunov functions which ensure that state trajectories are smooth and the stability of the system.

From a practical point of view, we are given a system with known relation between states and control input, and we desire to find the control which changes the state $\mathbf{x}$ so as to accomplish some desirable objective.

Since our problem is to find an optimal controller for hybrid system, modeled with hybrid automaton, we use the cost function (9), to calculate the optimal control for a continuous state and the following jump computation to minimize the cost of the jump.

Definition 5.1: We define the cost of the jump as the distance between the initial position on manifold $\Gamma[\mathbf{x}(t_0)]$ of the state i and the final position of terminal manifold $\Omega[\mathbf{x}(t_f)]$ of the state $i-1$. We instantiate the function $\theta[\mathbf{x}(t_f), t_f]$ as such distance function.

Remark 5.3: θ is an indicator of successful completion of the task: the bigger θ the lower the probability to complete the task.

We define:

Definition 5.2: Safety in a task execution is the property of completing the task without uncontrolled behaviors.

Thus we can state the following:

Remark 5.4: The distance θ is inversely proportional to the safety of the complete system execution.

For our problem the function $\theta[\mathbf{x}(t_f), t_f]$ is the distance between two points on the same manifold because we assume that:

Remark 5.5: The final configuration manifold $\Omega[\mathbf{x}(t_f)]$ of state $i-1$ and the initial configuration manifold $\Gamma[\mathbf{x}(t_0)]$ of state i intersect at $t_f = t_0$. The final point of the trajectory of the state $i-1$ and the initial point of the trajectory of the state i are on this manifold.

For curved or more complicated surfaces, the *metric* used to compute the distance between two points is an integration. With distance we mean the shortest distance between two points, the *geodesic*. The geodesics in a space depend on the Riemannian metric, which affects the notions of distance and acceleration, see [13].

Definition 5.3: Let $\delta(s)$ be a smooth curve on a manifold $\mathscr{M}$ from a to b with $\delta(0) = a$ and $\delta(1) = b$. Then $\delta\prime(s) \in \mathscr{T}_{\delta(s)}$ where $\mathscr{T}_a$ is the tangent space of $\mathscr{M}$ at a. The curve length of δ with respect to the Riemannian structure is given by $\int_0^1 |\,\delta\prime(s)\,|_{\delta(s)} dt$ and the distance $d(a,b)$ between a and b is the shortest distance between a and b given by

$$d(a,b) = \inf_{\delta: a \to b} \int_0^1 |\,\delta\prime(s)\,|_{\delta(s)} dt \qquad (16)$$

Thus, we can describe the θ function as a geodesic, where $\mathscr{M}[\mathbf{x}(t_f)] = \mathbf{0}$ is the manifold representing a subset of the configuration space, considering 5.5. Following Eq. (16), we can write

$$\theta[\mathbf{x}(t_f)] = \inf_{\delta: a \to b} \int_0^1 |\,\delta\prime(s)\,|_{\delta(s)} dt \qquad (17)$$

where

$$\delta = |\,\Gamma_i[\mathbf{x}(t_0)] - \Omega_{i-1}[\mathbf{x}(t_f)]\,| \quad \text{with } t_f = t_0 \qquad (18)$$

VI. The Algorithm

The procedure used to solve the multi-point boundary-value problem transforms the variational problem into a minimization problem with constraints and minimizes a functional. The output of the minimization is the control law.

The mathematical background of this problem is well defined but very few implementation results are present in the literature for such kind of minimization, where we do not have an objective function to minimize, neither we are checking for optimal parameters of this function but we compute a functional. We adapt Brayson's algorithm [14] to our problem formulation. The algorithm gives the possibility to choose between a fixed time with interval $0 < t < T$ and open final time, where t is variable. In the second case the optimization algorithm returns an optimal control law, with $0 < t < T$ where T is determined from the problem constrains. We resolve a dynamic optimization problem with open final time, with state and control constraints. The inequality in the state and control constrains are added to the problem as shown in Equations (19), (20) and (21). Consider an optimization problem with only one control variable $u(t)$, where u is bounded between u_{min} and u_{max}. We first introduce a change of control variables from u to u_1 where

$$u = \frac{(u_{max} + u_{min})}{2} + u_1 \frac{(u_{max} - u_{min})}{2} \qquad (19)$$

Using the new control variable u_1, we have

$$-1 \le u_1 \le 1 \qquad (20)$$

To handle this constraint, we next introduce a *slack control* variable u_2 where

$$u_1 = \cos(u_2) \qquad (21)$$

Numerical solutions using u_2 as control are then relatively straightforward using the gradient projection algorithm `seqopt`. However, if the exact solution contains abrupt switches from $u_1 = 1$ to $u_1 = -1$ or vice-versa, the solution using u_2 will have rapid but not abrupt changes, so it is an approximate solution. By increasing the number of integration steps, one can come as close as desired to the exact solution. If the inequality constraint involves only the state variables, we have to introduce one or more *slack state variables* as well as a slack control variable.

VII. Nominal Trajectory Computation

In this Section we study a simple robotic task, whose hybrid model is shown in Figure 1. This diagram represents a simplified surgical action, i.e. the initial steps of a suture.

The first state represents the subtask of positioning the end effector of the manipulator in contact with a compliant surface; in the second state an elastic force is opposing the tool tip until the force applied by robot breaks the surface. At the breaking point there is a transition to the third state in which the end effector of the manipulator moves through the surface and stops as soon as possible. For the sake of simplicity, we consider a 1-DOF manipulator in order to have simple dynamics and simplified constraints. Given the cost function (2) and the dynamic and kinematic constraint, we instantiate them for each state in the automaton. In the first positioning state we want to minimize the dissipated energy, i.e. $\phi[u(t)] = u^2$ while satisfying system dynamics, i.e. for the 1DOF manipulator:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{\tau_1}{m_{l_1} l_1{}^2 + I_{l_1}} \end{aligned} \tag{22}$$

The control bounds are due to the standard range of joint torque, i.e. $-10Nm < u < 10Nm$. The state constraints take into account system stability. In the more general case when an infinite number of switchings occurs, we add as state constraint a non-increasing Lyapunov function. Since in our system we force the jumps fixing the position of the jump occurrence, we have a finite number of switches and we assure the convergence with a restriction on the state of the "stop" subtask. Thus, only in the last subtask computation we take into account state constraints for stability purpose. The jump position is related to the system considered, time and amplitude are dependent from the optimal control problem, i.e. small amplitude and time correspond to small distance of the nominal trajectory to the real one.

The kinematic constraints are related to the terminal manifold, since the initial manifold is given. In the first state we have

$$\begin{aligned} x_2(t_f) &= 0 \\ x_1(t_f) &= c \end{aligned} \tag{23}$$

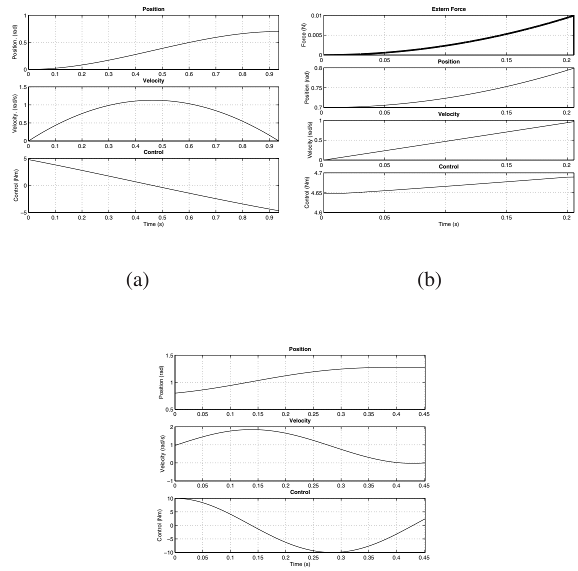
where c is the desired initial position for the contact state, that should be the final position of the first state. Figure 2.a shows the results of the computation algorithm. In the position plot is shown that a predefined position is reached smoothly, i.e. without spending too much effort. The velocity profile shows that the manipulator starts and arrives at the target with zero velocity. The control, third plot, changes direction in correspondence of the point where the velocity starts to decrease. In the second state we consider $\phi[u(t)] = u^2$ and the constraints due to the interaction with a compliant surface. The computation takes into account the external force applied by the membrane (F). The dynamics of the system is

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{\tau_1 - F}{m_{l_1} l_1{}^2 + I_{l_1}} \end{aligned} \tag{24}$$

We bound the velocity, Eq. (25), in order to have a smoother behavior during the contact and a better performance in the following stop state.

$$\begin{aligned} x_2(t) &< v_{max} \\ x_2(t) &> v_{min} \end{aligned} \tag{25}$$

The results of the simulation are shown in Figure 2.b. It is possible to recognize the breaking point (final point in the plot) where the force F is zero. In this subtask, position, control and velocity profile increase according to the subtask



(a) (b)

(c)

Fig. 2. (a) Positioning subtask: position, velocity and control function. (b) Contact subtask: position, velocity and control function. (c) Stop subtask.

behavior, i.e. the manipulator does not stop at the end of this subtask. In the third state we consider a different cost function because we want to minimize the time to stop after the membrane breaking, i.e. we use $\phi = 1$. The kinematic constraint is $x_2(t_f) = 0$ because the desired behavior for this subtask is to arrive at t_f with zero velocity satisfying system dynamics (22) and the stability constraint (27). We impose the convergence of the Lyapunov function, representing the energy of the system, (26) to the equilibrium point, i.e. zero.

$$V(\mathbf{x}) = \frac{1}{2} x_2{}^2 + (1 + m \cos x_1) \tag{26}$$

$$u \leq m \sin x_1 \tag{27}$$

where m is a multiplicative factor. The results are shown in Figure 2.c. The position increases, the velocity instead is still influenced by conditions of the end of the previous state, and it shows an initial increase, after which it goes to zero. The control decreases from the initial time of the subtask, as we expect. In the final part of the state the velocity is controlled so it starts to decrease still maintaining a negative value.

VIII. EXPERIMENTS

To carry out the experiments we use a PUMA 560 manipulator As a dynamical model for the optimization we use a model obtained with identification techniques, instead of considering the complete manipulator dynamics. The identification makes the simulation very near to the real system results. The control input is the torque τ applied by the motor, and the measured output is the angular position on the manipulator.

After the calculation of the off-line nominal trajectory is performed, on-line feedback must be used to correct the error due to environment and model uncertainties and to guarantee the completion of the task. Figure 3 shows how the automaton of the puncturing task is modified to take into account the on-line execution. The feedback represented by

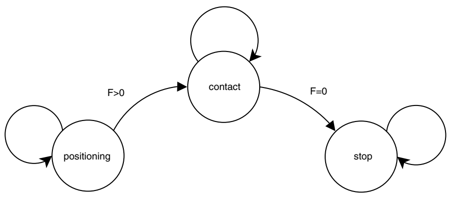


Fig. 3. Model of the task with feedback.

a loop in each state, correct the error respect to the nominal trajectory, i.e. the distance θ between manifolds Ω' and target Ω is zeroing. In this case each hybrid state has an optimal target coming from the optimization and a different kind of feedback control parameters is set up consequently. A velocity control is developed for the the positioning and stop states, and a force control is used in the contact state.

Thus the results (shown in Figure 4) verify the feasibility of this approach to the autonomous execution of a puncturing task.

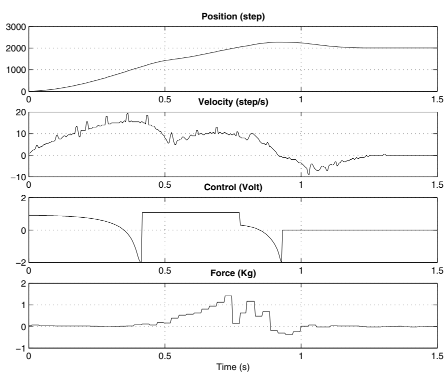


Fig. 4. Experiment data for autonomous membrane puncturing.

IX. Discussion and Conclusions

In the work described in this paper we represent a task as the deterministic action sequence of a hybrid automaton, and the computation of the optimal trajectory accounts for a single switch at each jump condition. Hence, the computation of the nominal trajectory can consider the discrete part of the hybrid system as constraints for the continuous optimal trajectory computation. These choices are motivated by the strict structure of the task that we are studying, but the theoretical framework proposed has a general validity and it will part of our future work to extend it to tasks with an uncertain structure.

The general approach to Hybrid Optimal Control Problem computes the optimal solution by making the best choice among the many switching possibilities at each state jump. In the problem studied here, we use a priori knowledge in the switching selection and our main concerns are good performance within each subtask and safety of the complete execution. These concerns are transformed into the constraints of the off-line optimization used to compute the nominal trajectory of the desired task. During on-line execution, task feedback is represented by the distance θ, between nominal and current performance. If we define "task safety" during execution as the property of the task to be completed without uncontrolled behaviors, then the distance θ, between nominal and actual trajectories represents the safety of the complete task execution.

With the approach proposed in this paper we have shown in simulation and with experiments that a complex task execution can be completed autonomously, is safe, with good performance and efficient. In the method presented here, a priory knowledge plays a big role, since it defines the hybrid automaton structure and the jump sequences, but this is not a restriction to the solution of more general problems. In fact, we transformed the hybrid optimal control problem into an optimal control problem and we used the hybrid system models only as a solution strategy to autonomous task execution.

References

[1] M. Campos, G. Pereira, S. Vale, A. Bracarense, G. Pinheiro, and M. Oliveira, "A mobile manipulator fo rinstallation and removal of aircraft warning spheres on aerial power transmission lines," in *IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp. 3559–3564.

[2] L. Chaimowicz, M. Campos, and V. Kumar, "Dynamic role assigment for cooperative robots," in *IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp. 293–298.

[3] ——, "Hybrid systems modeling of cooperative robots," in *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003, pp. 4086–4091.

[4] D. Kragic, A. Miller, and P. Allen, "Realtime tracking meets on line grasping planning," in *IEEE International Conference on Robotics and Automation*, Seoul, Korea, May, 21-26 2001, pp. 2460–2465.

[5] C. Tomlin, G. Pappas, J. Kosecka, J. Lygeros, and S. Sastry, *Air traffic automation: a case study in distributed decentralized control*, ser. Lecture Notes in Control and Information Sciences. Springer, 1998, pp. 261–292.

[6] M. S. Branicky, V. S. Borkar, and S. K. Mitter, "A unified framework for hybrid control: Model and optimal optimal control theory," *IEEE Transaction on Automatic Control*, vol. 43, no. 1, pp. 31–45, April 1998.

[7] X. Xu and J. Antsaklis, "A dynamic programmin approach for optimal control of switched systems," in *39th IEEE Conf. On Decision and Control*, Dec. 2000, pp. 1822–1827.

[8] S. Hedlund and A. Rantzer, "Optimal control of hybrid systems," in *38th IEEE Conf. On Decision and Control*, Dec 1999, pp. 3972–3977.

[9] R. Sengupta and S. Lafortune, "An optimal control theory for discrete event systems," *SIAM Control and Optimization*, vol. 36, no. 2, pp. 488–541, 1998.

[10] G. Grammel, "Maximum principle for hybrid system via singular pertrubations," *SIAM Control and Optimization*, vol. 37, no. 4, pp. 1162–1175, 1999.

[11] E. McShane, *On Multipliers for Lagrange Problems*. New York: American J. Math., 1939, vol. 61.

[12] L. Pontryagin, *The Mathematical Theory of Optimal Processes*. New York: Wiley, 1962.

[13] P. M. Do Carmo, *Differential Geometry of Curves and Surfaces*. Upper Side river, New Jersey: Prentice-Hall, 1976.

[14] A. Bryson and Y.-C. Ho, *Applied Optimal Control*. Hemisphere Publishing Corporation, 1975.