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

Optimal Control of Hybrid Systems Using Statistical Learning

C. Collaro, C.T. Abdallah, A. Tornambè, and U. Dole

Abstract— **In this paper we discuss how statistical learning methods may be used to obtain solutions to various optimal control strategies for hybrid systems. The results are illustrated on the problem of autonomous navigation for mobile robots in the presence of obstacles.**

I. INTRODUCTION

When dynamical systems are composed of the interaction of continuous and discrete-dynamics, they become *hybrid systems*. While such systems have been around for a long time (temperature control, process control, telecommunications systems, manufacturing systems, are examples of hybrid systems), and more realistically model various physical phenomena, it was recently that their study has become a mainstay of control theory [1], [11]. The modeling and control of such systems relies on concepts from control theory, computer science and simulation, as well as mathematics. At one end of the spectrum are the usual continuous-time or discrete-time dynamics well understood by control theorists, while at the other end are the event-driven dynamics (Automata) favored by computer scientists. The combination of these points of views leads to various hybrid models such as hybrid automata, hybrid Petri nets, piecewise-affine models, etc. It turns out however, that many questions related to the modeling and control of hybrid systems are hard, and more specifically many decision problems (is a system stabilizable?, is it reachable?) are NP-hard [2]. In earlier work by one of the authors [10] and others [14], [12], it was recommended that statistical learning techniques may offer useful, albeit approximate answers to some NP-hard problems. It was also shown in [10], [14], [12] and others that some difficult optimization and robust control problems may be approximately solved using statistical learning methods. It is exactly this track that has led us to apply such methods in various optimal control scenarios of hybrid systems.

The remainder of this paper is organized as follows: Section II provides a quick overview of statistical learning concepts. In sectionIII we present a list of optimal control problems for hybrid systems that might benefit from the statistical learning methods. In section IV the autonomous control problem for mobile robots, in the presence of obstacles and uncertainties is studied and statistical learning methods are proposed for its solution. Section V presents our results and our conclusions are summarized in section VI.

The work of C.T. Abdallah is partially supported by NSF-0233205

U. Dole and C.T. Abdallah are with the Electrical and Computer Engineering Department, MSC01 1100, The University of New Mexico, Albuquerque, NM 87131-0001, USA. {umesh,chaouki}@ece.unm.edu

C. Collaro and A. Tornambé are with DISP, University of Roma, Tor Vergata, Via del Politecnico 1 00133, Roma - ITALY. {collaro,tornambe}@disp.uniroma2.it

II. STATISTICAL LEARNING OVERVIEW

In order to introduce statistical learning concepts, let $(S, \mathcal{A})$ be a measurable space and let $\{X_n\}_{n\geq 1}$ be a sequence of independent identically distributed (i.i.d) observations in this space with common distribution P. We assume that this sequence is defined on a probability space $(\Omega, \Sigma, \mathbb{P})$. Denote by $\mathcal{P}(S) := \mathcal{P}(S, \mathcal{A})$ the set of all probability measures on $(S, \mathcal{A})$. Suppose $\mathcal{P} \subset \mathcal{P}(S)$ is a class of probability distributions such that $P \in \mathcal{P}$. In particular, if one has no prior knowledge about P, then $\mathcal{P} = \mathcal{P}(S)$. In this case, we are in the setting of *distribution free learning.* One of the central problems of statistical learning theory is *the risk minimization problem.* It is crucial in all cases of learning (standard concept or function learning, regression problems, pattern recognition, etc.). It also plays an important role in randomized (Monte Carlo) algorithms for robust control problems, as has been shown by Vidyasagar [14]. Given a class $\mathcal{F}$ of $\mathcal{A}$-measurable functions f from S into $[0, 1]$ (e.g., decision rules in a pattern recognition problem or performance indices in control problems), the risk functional is defined as

$$R_P(f) := P(f) := \int_S f dP := \mathbb{E}f(X), \ f \in \mathcal{F}.$$

The goal is to find a function f_P that minimizes R_P on $\mathcal{F}$. Typically, the distribution P is unknown (or, as it occurs in many control problems, the integral of f with respect to P is too hard to compute) and the solution of the risk minimization problem is to be based on a sample $(X_1, \ldots, X_n)$ of independent observations from P. In this case, the goal of statistical learning is more modest: given $\varepsilon > 0, \delta \in (0, 1)$, find an estimate $\hat{f}_n \in \mathcal{F}$ of f_P, based on the data $(X_1, \ldots, X_n)$, such that

$$\sup_{P\in\mathcal{P}} \mathbb{P}\{R_P(\hat{f}_n) \geq \inf_{f\in\mathcal{F}} R_P(f) + \varepsilon\} \leq \delta. \quad (1)$$

In other words, one can write that with probability $1 - \delta$, $R_P(\hat{f}_n)$ is within ε of $\inf_{f\in\mathcal{F}} R_P(f) = R^*$. Denote by $\tilde{N}^L_{\mathcal{F},\mathcal{P}}(\varepsilon; \delta)$ the minimal number $n \geq 1$ such that for some estimate $\hat{f}_n$ the bound (1) holds, and let $\tilde{N}^U_{\mathcal{F},\mathcal{P}}(\varepsilon; \delta)$ be the minimal number $N \geq 1$ such that for some sequence of estimates $\{\hat{f}_n\}$ and for all $n \geq N$ the bound (1) holds. Let us call the quantity $\tilde{N}^L_{\mathcal{F},\mathcal{P}}(\varepsilon; \delta)$ *the lower sample complexity* and the quantity $\tilde{N}^U_{\mathcal{F},\mathcal{P}}(\varepsilon; \delta)$ *the upper sample complexity* of learning. These quantities show how much data we need in order to guarantee certain accuracy ε of learning with certain confidence level $1-\delta$. Clearly, $\tilde{N}^L_{\mathcal{F},\mathcal{P}}(\varepsilon;\delta) \le \tilde{N}^U_{\mathcal{F},\mathcal{P}}(\varepsilon;\delta)$, and it is easy to show that the inequality can be strict. The upper sample complexity is used rather frequently in statistical learning theory and is usually referred to simply as the sample complexity.

A method of *empirical risk minimization* is widely used in learning theory. Namely, the unknown distribution P is replaced by *the empirical measure* P_n, defined as

$$P_n(A) := \frac{1}{n}\sum_{k=1}^{n} I_A(X_k),\ A \in \mathcal{A}$$

where $I_A(x) = 1$ for $x \in A$ and $I_A(x) = 0$ for $x \notin A$. The risk functional R_P is replaced by the empirical risk R_{P_n}, defined by

$$R_{P_n}(f) := P_n(f) := \int_S f dP_n := \frac{1}{n}\sum_{k=1}^{n} f(X_k),\ f \in \mathcal{F}.$$

The problem is now to minimize the empirical risk R_{P_n} on $\mathcal{F}$, and we let $f_{P_n} \in \mathcal{F}$ be a function that minimizes R_{P_n} on $\mathcal{F}$.

In what follows, f_{P_n} is used as our learning algorithm, i.e. $\hat{f}_n := f_{P_n}$. Determining the sample complexity of the empirical risk minimization method is definitely one of the central and most challenging problems of statistical learning theory (see, e.g., [10], or Vidyasagar [14] for the relevant discussion in the context of robust control problems). A reasonable upper bound for the sample complexity can be obtained by finding the minimal value of n for which the expected value $\mathbb{E}f(X)$ is approximated uniformly over the class $\mathcal{F}$ by the empirical means with given accuracy ε and confidence level $1-\delta$. More precisely, denote

$$\begin{aligned} N(\varepsilon,\delta) &:= N^L_{\mathcal{F},\mathcal{P}}(\varepsilon,\delta) \qquad (2)\\ &:= \min\Big\{n \ge 1 : \sup_{P\in\mathcal{P}} \mathbb{P}\big\{\|P_n - P\|_{\mathcal{F}} \ge \varepsilon\big\} \le \delta\Big\}\end{aligned}$$

where $\|\cdot\|_{\mathcal{F}}$ is the sup-norm in the space $\ell^\infty(\mathcal{F})$ of all uniformly bounded functions on $\mathcal{F}$. Let us call the quantity $N(\varepsilon;\delta)$ *the (lower) sample complexity of empirical approximation* on the class $\mathcal{F}$.

Unfortunately, the quantity $N^L_{\mathcal{F},\mathcal{P}}(\varepsilon,\delta)$ is itself unknown for most of the nontrivial examples of function classes, and only rather conservative upper bounds for this quantity are available. These bounds are expressed in terms of various entropy characteristics and combinatorial quantities, such as VC-dimensions, which themselves are not always known precisely and are replaced by their upper bounds. Recent work however [12], [14], [10] has shown the applicability of randomized algorithms in order to find bounds on the number of samples to come within ϵ from an optimum point with high confidence. It is exactly these techniques we intend to apply on the optimal control problems discussed next.

An algorithm that was proved in [10] provides a general solution to the optimization problems encountered so far, and gives a bound on the number of samples n that will guarantee some accuracy and confidence. It is this algorithm that is used later to solve our optimization problem for a hybrid system. Note that the algorithm does not require the function f to be optimized to be differentiable, or even continuous. It is only required to be measurable.

III. Optimal Control of Hybrid Systems

In order to illustrate the optimal control problems that might benefit from such algorithms, let us first focus on the class of hybrid systems described in [3]. Note that the robots used in the experimental set-up in section IV can actually fit this model, when the coordination goal is to take each robot through a path in a minimum time while satisfying a quality performance objective (tracking errors, collision avoidance, etc). Consider N systems each evolving according to the state equation

$$\dot{z}_i = g_i(z_i, u_i, t);\ z_i(\tau_i) = \zeta_i;\quad i = 1, \cdots, N \qquad (3)$$

The control signal u_i is used to attain a desired final state belonging to the target quality set $\Gamma_i(u_i)$. If $s_i(u_i)$ is the service time (the time during which system i is active), we have the following:

$$s_i(u_i) = \min\{t \ge 0; z_i(\tau_i + t) \in \Gamma_i(u_i)\} \qquad (4)$$

Meanwhile, the temporal state x_i which represents the time when state z_i is active, evolves according to the so-called Lindley equation

$$x_i = \max(x_{i-1}, a_i) + s_i(u_i) \qquad (5)$$

The optimal control problem for such a hybrid system is then stated as follows,

$$\min_{u_1,\cdots,u_N} J = \sum_{i=1}^{N} L_i(x_i, u_i) \qquad (6)$$

subject to the dynamic equations (3) and (5) and where the function $L_i(x_i, u_i)$ is the cost associated with system i. As detailed in the paper [3], one of the main difficulties is that the max function is non-differentiable exactly at the points $x_{i-1} = a_i$. Recall that differentiability was not needed when applying algorithm **??**. In fact, if one defines the augmented cost

$$\begin{aligned} \bar{J} \quad & (x,\lambda,u) = \qquad (7)\\ \textstyle\sum_{i=1}^{N} \quad & [L_i(x_i,u_i) + \lambda_i(\max(x_{i-1},a_i) + s_i(u_i) - x_i)]\end{aligned}$$

where L_i and s_i are differentiable functions, and λ is the co-state, immediate difficulties arise in applying the first-order necessary conditions for optimality because of the presence of max. One has to resort then to using generalized gradients and non-smooth optimization techniques [4] in order to proceed further [3]. On the other hand, a randomized algorithm rooted in statistical learning, will provide an efficient albeit approximate solution to the minimization problem.

IV. Autonomous Control of Mobile Robots

In order to experimentally validate our algorithms, we focus in the remainder of the paper specifically on the problem of autonomous navigation of mobile robots. It is important for the autonomy of such robots that they interact with their environment. Typically, mobile robots collect sensor information, process them to extract features or detect obstacles, and execute appropriate control laws for optimal, obstacle-free navigation. However, different controllers activate different behaviors such as obstacle avoidance, approach target, etc. and they need to be coordinated for a given navigation task. It is this coordination that when implemented with a *hybrid automaton*, helps activate an event-specific behavior. A *path following* navigation task is considered here. It involves the subtasks of *obstacle avoidance* and *approach target* [5]. These two behaviors define the nodes of the hybrid automaton and there is a minimum safety distance, d_{OA} governing the discrete transitions between them. However, the real problem is how to coordinate these behaviors from the point of view of safety and optimality. If we fuse them such that they are simultaneously active and the system control vector is the sum of their control vectors, then in spite of achieving smooth performance, we undermine our goals of reducing complexity and dependency. If, on the other hand, we use hard switches [5] then the system might experience chattering, which would definitely degrade the overall system performance. The solution to this problem lies in removing the *Zeno* properties of the system by adding nodes to the hybrid automaton in order to *regularize* the overall system. Let us suppose the system dynamics described in the plane by $[\dot{x} \;\; \dot{y}] = F(x, y)$. If F_{OA} and F_{GA} represent the continuous dynamics of the obstacle avoidance and goal attraction behaviors, respectively, then the *sliding solution* that will regularize the system is chosen such that $f_S \perp F_{OA}$ [5]. This is due to the fact that the repulsive potential field under the typical *reactive* obstacle avoidance behavior is orthogonal to the surface on which the behavior becomes active [5]. The problem now is to design a controller such that the control variable drives the mobile robot around the obstacles at the preferred safety margin and avoids high curvatures in the path which cross the physical limits of the signals that the actuators can track. The main idea is to generate near-optimal trajectories around the obstacles to the goal, and then convert this navigation problem into a *trajectory tracking* one. With the given tracking algorithm, the steady-state tracking error can be made very small as the robot's steering angle ϕ exponentially converges to its desired value ϕ_d [6].

Let us suppose that the environment is characterized by the presence of three obstacles of different dimensions as seen in Figure 12. Since the robot has to approach the final target along a minimum path and safely, it has to avoid the obstacles every time they are detected. We then need to define optimal coordination between two different behaviors: 1) *Goal Attraction* or attractive behavior, and 2) *Obstacle Avoidance* or reactive behavior. We assume that the reactive behavior is valid only in a circular safety region around the obstacle (represented by the outer circles in Figure 1), while the attractive one is true elsewhere. Based on our assumptions:

- $(X_t, Y_t) \rightarrow$ Final Destination;
- $(X_o, Y_o) \rightarrow$ Initial Position;
- $(X_{ob_i}, Y_{ob_i}) \rightarrow$ Obstacle Position in the plane;
- $(\overline{Gap_i}, \underline{Gap}_i \rightarrow$ Upper and lower safety Gap Bounds;
- $i = \{A, B, C\} \rightarrow$ Three Obstacles.

Thus, we represent each behavior (continuous dynamics) of the system as a specific node of a *Control Graph*, while the *Guard Conditions*, associated with each edge, define discrete transitions between these different dynamics.

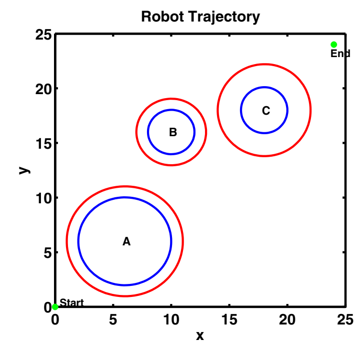

Fig. 1. Navigation Environment

In order to design a Hybrid Automaton of the Mobile Robot Navigation system, we need to define the three different behaviors that characterize the robot during its navigation through the obstacles:

1) *Goal Attraction* behavior:

$$F_{GA} = [\; F_{GA_x} \quad F_{GA_y} \;],$$

where $F_{GA_x} = M\ddot{x} = -K_{d1}\dot{x} - K_{p1}(x - X_t)$, $F_{GA_y} = M\ddot{y} = -K_{d2}\dot{y} - K_{p2}(y - Y_t)$.

2) *Obstacle Avoidance* behavior:

$$F_{OA} = [\; F_{OA_x} \quad F_{OA_y} \;],$$

where

$$F_{OA_x} = M\ddot{x} = \frac{1}{(x - X_0)^n}$$

$$F_{OA_y} = M\ddot{y} = \frac{1}{(y - Y_0)^n}$$

3) *Sliding* behavior:

$$F_S = [\; -F_{OA_y} \quad F_{OA_x} \;]$$

The Sliding behavior is used to regularize the system by adding nodes to the control graph that characterized the Hybrid Automaton model. The sliding dynamics are orthogonal to the repulsive potential field (obstacle avoidance) which is true inside the *Safety Area* as shown in Figure 2.

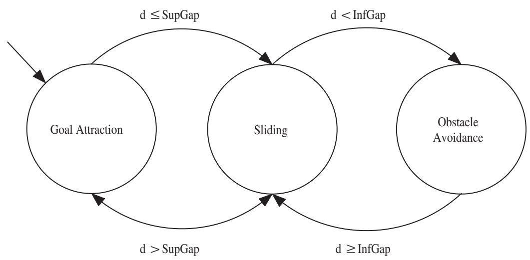

Fig. 2. Hybrid Automaton

V. SIMULATION

The aim of the simulation is to show how statistical learning methods may be used in order to obtain optimal solutions to autonomous navigation of a mobile robot in the presence of obstacles. Given the model described before, let's suppose the dimension of the safety areas (both the *Safety Gap* and the *Safety Region*) changing according to some environment conditions, for example the shape of the obstacles or the plant of the area where the mobile robots are presumed to move.

Since the coordination goal is to take the robot through a minimum path avoiding undesired obstacle collisions and satisfying some other qualities performance conditions (as minimizing the number of commutations of the robot), let's define the objective function of the mobile robot problem as a linear combination of two different parameters: $f_o = \frac{1}{3}D + C$where *Distance*=D is the length of the path of the robot from an initial position to a final destination, and *Commutation*=C is the number of changes of direction requested to the robot to reach the final target safely. While the safety areas are circumferences, the idea is trying to change their dimension (by modifying the radius of each circle) for two of the three obstacles considered in the model (A and C), and to find the optimal condition for the navigation robot mobile problem (3). The optimal control problem for such a Hybrid System is then stated as: $Min_{i=1}^{N}(\frac{1}{3}Distance_i + Commutaion_i)$. Let us suppose that environment conditions

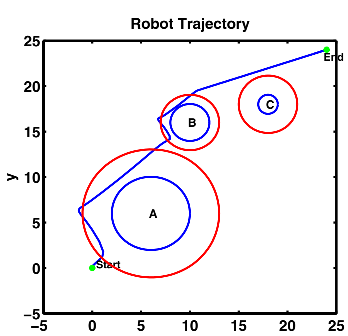

Fig. 3. Obstacle Avoidance

allow one to consider for each circle (that identifies the Safety Region and the Safety Gap around the obstacles A and C) different possible dimensions Figure 4. These dimensions are given by a combination of four different set of input values (one set for each circle $\overline{Gap}_{A,C}$ and $\underline{Gap}_{A,C}$). After

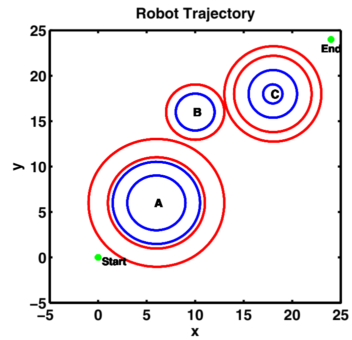

Fig. 4. Safety area combinations

defining an upper bound and a lower bound for each set $\overline{Gap}_A \in [3,7]$, $\underline{GapA} \in [2,6]$, $\overline{Gap}_C \in [1.5, 5.5]$, and $\underline{Gap}_C \in [0.5, 4.5]$, we define nine possible values for each interval. It turns out that there are $N = n \times m \times k \times l = 9^4$

Table 1: Input values for Robot Mobile Navigation

$\overline{Gap}_A$	3	3.5	4	4.5	5	5.5	6	6.5	7
$\underline{Gap}_A$	2	2.5	3	3.5	4	4.5	5	5.5	6
$\overline{Gap}_C$	1.5	2	2.5	3	3.5	4	4.5	5	5.5
$\underline{Gap}_C$	0.5	1	1.5	2	2.5	3	3.5	4	4.5

possible combinations that characterize a sample of independent observations (Table 1). Each observation is obtained by a simulation of the robot mobile's model, according to a particular combination of the input values. The output of each simulation provides different values of the main parameters: *length of the path* and *number of commutation*, so that we can evaluate the optimal condition from a set of possible solutions. We did however apply our learning algorithm with 3600 samples (obtained from a prior experiment), and used the following values: $K_{di} = 5, K_{pi} = 1$ to navigate from $(X_), Y_0) = (0,0)$ to $(X_t, Y_t) = (24,24)$ while avoiding obstacles at $(X_{oA}, Y_{oA}) = (6,6)$,$(X_{oB}, Y_{oB}) = (10,16)$, and $(X_{oC}, Y_{oC}) = (18,18)$. In order to reach our goal, we used the interaction properties between Simulink and Stateflow (fig. 5). We note each combination of the input parameters $\underline{Gap}_A, \underline{Gap}_C\}$, and $\overline{Gap}_A, \overline{Gap}_C\}$ define a particular solution to the safe navigation problem. The general trend is that the more the Safety Gap and the Safety Area are decreased, the larger the number of commutations, and the longer the length of the path. This is why the Sliding behavior, used to regularize the system, is valid in a smaller area (see Figures fig:dessin8,fig:dessin9,fig:dessin10).

It turns out that most of the combinations provide a number of commutation between values four to six (see Figure 9), and a length of the path between 41 to 46 (Figure 10). The robot never collides with the obstacles, and it trajectories are fairly efficient.

By choosing the parameters that minimize the objective

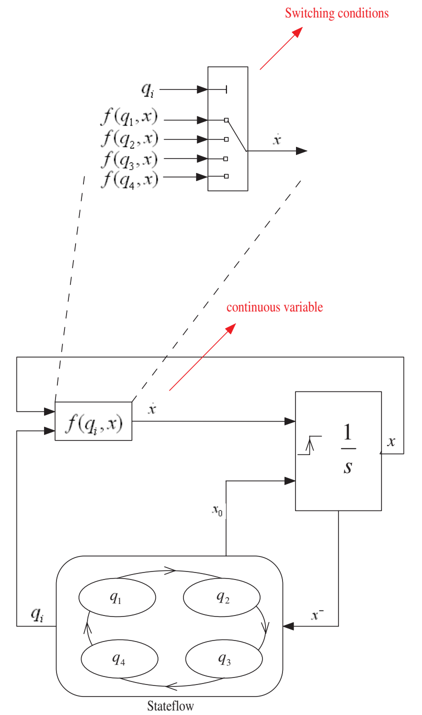

Fig. 5. modeling with Simulink and Stateflow

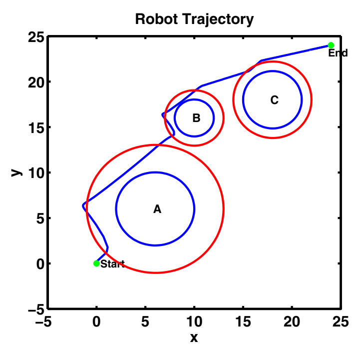

Fig. 6. Set [7,4,4,3]

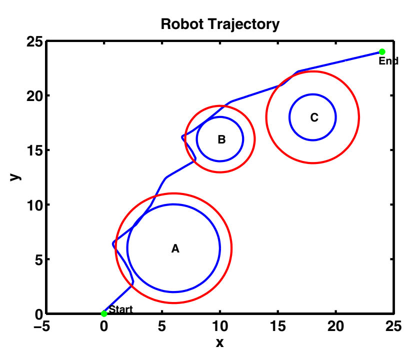

Fig. 7. Set [5,4,4,2]

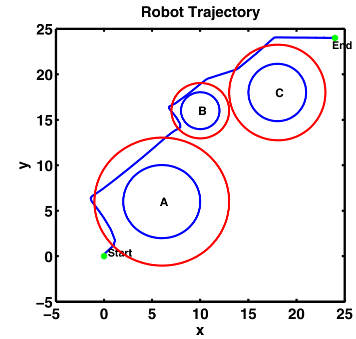

Fig. 8. Set [7,4,5,3]

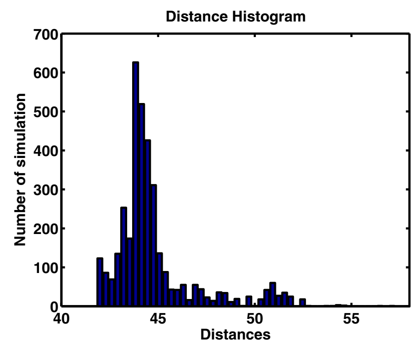

Fig. 9. length's path Distribution

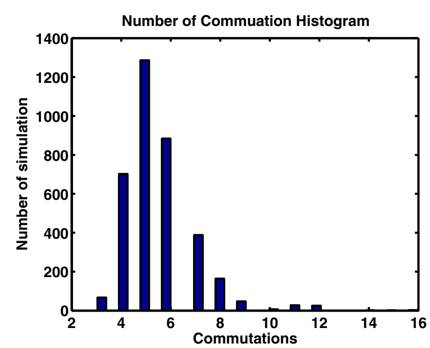

Fig. 10. Commutation Distribution

function, we can see that the optimal conditions for the navigation of mobile robot problem are given by more than a combination (Table 2).

In particular, we can see how the dimension of the Safety Region, relative to the Obstacle C, has to be less than 3.5, otherwise the robot must avoid also this obstacle, increasing the number of commutation and the path. In the other hand, the dimension of Obstacle A, has to be fixed to the values $\overline{Gap}_A = 6, \underline{Gap}_A = 5$).

Suppose these conditions are true, then the robot never collides any obstacle, and it reaches the target in a minimum path (fig.13). However, by choosing ($Sup_{Gap_A} = 6, Inf_{Gap_A} = 5$) and ($Sup_{Gap_C} = 4.5, Inf_{Gap_C} = 3.5$) as input values, we can see how the trajectory of the mobile

Table 2: Optimal combinations

$\overline{Gap}_A$	$\underline{Gap}_A$	$\overline{Gap}_C$	$\underline{Gap}_C$	D	C	Fo
6	5	1.5	0.5	43.354	3	16.006
6	5	1.5	1	43.354	3	16.006
6	5	1.5	1.5	43.354	3	16.006
6	5	2	0.5	43.354	3	16.006
6	5	2	1	43.354	3	16.006
6	5	2	1.5	43.354	3	16.006
6	5	2	2	43.354	3	16.006
6	5	2.5	0.5	43.354	3	16.006
6	5	2.5	1	43.354	3	16.006
6	5	2.5	1.5	43.354	3	16.006
6	5	2.5	2	43.354	3	16.006
6	5	2.5	2.5	43.354	3	16.006
6	5	3	0.5	43.354	3	16.006
6	5	3	1	43.354	3	16.006
6	5	3	1.5	43.354	3	16.006
6	5	3	2	43.354	3	16.006
6	5	3	2.5	43.354	3	16.006
6	5	3.5	0.5	43.354	3	16.006
6	5	3.5	1	43.354	3	16.006
6	5	3.5	1.5	43.354	3	16.006
6	5	3.5	2	43.354	3	16.006
6	5	3.5	2.5	43.354	3	16.006
6	5	3.5	3	43.354	3	16.006
6	5	3.5	3.5	43.354	3	16.006

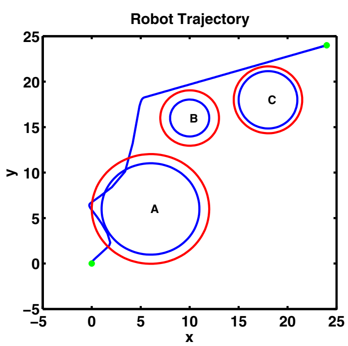

Fig. 11. Optimal condition for the Robot Mobile Navigation Problem

robot is no more optimal (fig.14).

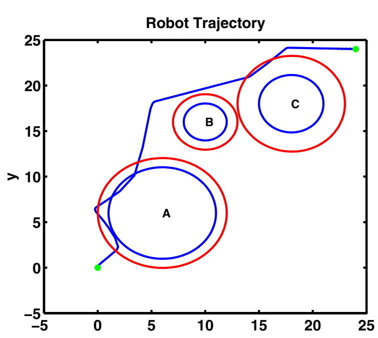

Fig. 12. Optimal condition for the Robot Mobile Navigation Problem

VI. Conclusions

In this paper, we have proposed to apply statistical learning algorithms to various optimal control problems for hybrid systems. Such problems are particularly suited for randomized algorithms due to the presence of non-differentiable cost functions, along with continuous and discrete dynamics. We have focused our attention on the problem of autonomous navigation of mobile robots.

In the behavior-based method above, the robot model is assumed certain. However, in practice, the wheel radi r_1 and r_2 of the left and right wheels change, and as a result the robot model usually has *parametric uncertainty*. Such an imprecise model cannot have a unified controller that yields optimal performance as described in [8]. If the uncertain parameters take values over a finite set $\mathcal{P}$ then we can design a family of *estimators* and *controllers* by discretizing the parameter space [9]. The two approaches mentioned above address the problem of hybrid systems based mobile robot navigation from two different perspectives. The first approach includes a behavior-based framework and shows how different behaviors can be integrated and implemented as one hybrid system. The second approach considers a single robotic task, that of parking, and uses the hybrid systems approach to make the overall system robust to parametric uncertainty. A possible scenario where the two approaches are combined will lead into a situation where the statistical learning approaches may be applied.

References

[1] Antsaklis, P. Editor, *Hybrid Systems: Theorey and Applications*, Special Issue of Proceedings of the IEEE, Vol. 8, No.7, 2000.

[2] Blondel,V., Tsitsiklis, J., “Complexity of Stability and Controllability of Elementary Hybrid Systems,” *Automatica*, Vol. 35, No. 3, pp. 479-489, 1999.

[3] Cassandras, C.G., Pepyne, D.L., Wardi, Y., “Optimal Control of a Class of Hybrid Systems,” *EEE Transactions on Automatic Control*, Vol. 46, No. 3, March. 2001, pp. 398-415.

[4] Clarke, F.H., *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, 1983.

[5] Egerstedt, M., “Behavior Based Robotics Using Hybrid Automata” in Lecture Notes in Computer Science: Hybrid Systems III: Computation and Control, pp. 103-116, Pittsburgh, PA, Springer-Verlag, March 2000.

[6] Egerstedt, M., Hu, X., Stotsky, A., “Control of Mobile Platforms Using a Virtual Vehicle Approach”, *IEEE Transactions on Automatic Control*, Vol. 46, No. 11, Nov. 2001, pp. 1777-1782.

[7] Garey, M.R., Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness* Freeman, New York, 1979.

[8] Hespanha, J. P., Morse, A. S., “Certainty Equivalence Implies Detectability”, *Systems & Controls Letters*, 1999, Vol. 36, pp. 1-13.

[9] Hespanha, J. P., Liberzon, D., Morse, A. S., “Logic-based Switching Control of a Nonholonomic System with Parametric Modeling Uncertainty”, *Systems & Controls Letters*, Nov. 1999,Vol. 38, pp. 167-177.

[10] Koltchinskii, V., Abdallah, C.T., Ariola, M., Dorato, P., Panchenko, D., “Improved Sample Complexity Estimates for Statistical Learning Control of Uncertain Systems,” *IEEE Transactions on Automatic Control*, Vol. 46, pp. 2383-2388, 2000.

[11] Liberzon, D., *Switching in Systems and Control,* Birkhaüser, Boston, 2003.

[12] Tempo, R., Calafiore, G., Dabbane, F., *Randomized Algorithms for Analysis and Control of Uncertain systems*, Springer-Verlag, London, 2005.

[13] Tomlin, C.J., Lygeros, J., Sastry, S.S., “A Game Theoretic Approach to Controller Design for Hybrid Systems,” *Proceedings of the IEEE*, Vol. 88, No. 7, pp. 949-970, 2000.

[14] Vidyasagar, M., *Learning and Generalization: With Applications to Neural networks*, Springer-Verlag, New York 2002.

[15] Vidyasagar, M., Blondel, V., “Probablilistic Solutions to some NP-hard Matrix Problems,” *Automatica*, Vol. 37, 2001, pp. 1397-1405.