

Reactive Robot Navigation Using Optimal Timing Control

Henrik Axelsson, Magnus Egerstedt, and Yorai Wardi
{henrik,magnus,ywardi}@ece.gatech.edu
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Abstract—In this paper a solution is presented for the problem of avoiding obstacles while progressing towards a goal for a single robot. In particular, an optimal solution is obtained by allowing the robot to switch between a fixed number of behaviors and optimizing over what behaviors to use and when to switch between them. We moreover show that the structure of the switch law only depends on the distance between the obstacle and the goal. Hence, once initial simulations are done, the structure of the guard is known to the robot and, given that the robot knows the distance between the obstacle and the goal, it knows when to switch to obtain the optimal solution. Therefore the solution lends itself to real-time implementations. The feasibility of the approach is verified in real robotics experiments.

I. INTRODUCTION

In the literature on robot navigation, two distinctly different approaches have emerged. The first approach that we will denote by the *reactive* approach (following the terminology in [1]) consists of designing a collection of behaviors, or modes of operations, such as avoid-obstacle or approach goal. These different behaviors are defined through a particular control law, dedicated to performing a specific task, and the robot switches between different behaviors as obstacles, landmarks, etc. are encountered in the environment. This way of structuring the navigation system has the major advantage that it simplifies the design task. Each controller is designed with only a limited set of objectives under consideration and no elaborate world maps are needed. Unfortunately, very little can be said analytically about such systems, and we contrast them with the second approach under consideration here, namely the *deliberative* approach. Here the motion is carefully planned out in advance and care can be taken as to minimize energy consumption and so on. This plan-based approach has proved very useful in structured environments, e.g. in industrial settings, while unstructured environments pose a challenge. This is due to the fact that there is normally a hefty computational burden associated with path planning and optimal control. And, even if one is willing to pay this cost once, as soon as unmodeled obstacles are encountered, the cost will be incurred again.

In this paper we stay within the reactive navigation

architecture but argue that optimality might still be relevant. Assuming that a number of control modes, or behaviors, have been designed, the question remains as to when to switch between them. This problem can be referred to as the guard design problem for hybrid systems, where a guard enables the transition between different modes of operation. Our approach is thus similar in spirit to the program developed in [2] where the guards were derived based on game theory to ensure safety in a multi-aircraft scenario. Formally, the state of the system evolves in mode i as $\dot{x} = f_i(x)$ until $G_{ij}(x) = TRUE$, at which point the mode changes from i to j , as seen in Figure 1. The particular problem that we will investigate in this paper is the problem of switching between *go-to-goal* and *avoid-obstacle* in an optimal manner. Previously proposed guards typically involve a safety distance Δ so that $\dot{x} = f_g(x)$ (subscript g denotes go-to-goal) as long as $\|x - x_{ob}\| > \Delta$, where x_{ob} is the location of the obstacle ([3], [4], [5], [6]). If $\|x - x_{ob}\| \leq \Delta$ then $\dot{x} = f_o(x)$ (subscript o denotes avoid-obstacle) and hence the guard is defined through a circle centered at x_{ob} with radius Δ . One can thus view the optimal control problem as a problem of determining the optimal Δ , or more generally, to optimize a parameterized surface $g_\alpha(x) = 0$ with respect to α . (See [7], [8] for a discussion of this topic). Unfortunately, no guarantee can be given that this is the optimal surface class and in this paper we take an alternate route and view the optimization problem as a free timing control problem. Subject to certain regularity conditions on the dynamics and the cost, we will show how to obtain the optimal surface by varying the initial conditions. Once such a surface has been obtained, in a potentially costly simulation, it can be implemented as a guard in a real-time reactive navigation system.

The outline of this paper is as follows: In Section 2 we formalize the problem under consideration and we devote Section 3 to its solution. Section 4 is concerned with the development of guards, suitable for implementation, and we conclude the paper with a real robotics example in Section 5.

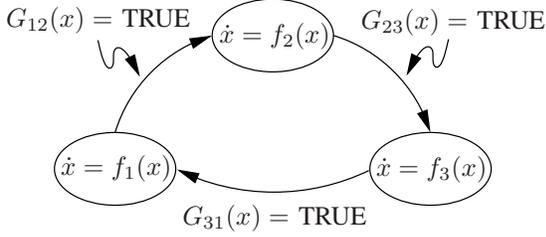


Fig. 1. Mode transition occur as the guard predicates become true.

II. OBSTACLE-AVOIDANCE

Since we are not interested in optimal control for energy minimization, but rather for path-based properties, we simply let the dynamics of the robot be given by an integrator, $\dot{x} = u \in \mathbb{R}^2$. Moreover, denote by x_{ob} the position of the obstacle and by x_g the position of the goal. The general optimal control problem for solving the obstacle avoidance problem thus becomes

$$P1: \quad \min_u J(u) = \int_0^T L(x(t))dt$$

s.t. $\dot{x} = u,$

where T is the time span we are concerned with.

In order for the robot not to collide with any obstacles while moving towards the goal, the cost-function $L(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}_+ \cup \{0\}$ should include a term that penalizes the robot for being far away from the goal and one term that incurs a cost whenever the robot is close to an obstacle. The *goal-attraction* term used in this paper is given by $g(x) = \rho \|x_g - x\|^2$ and the *avoid-obstacle* term is given by $h(x) = \alpha \exp(-\frac{\|x_{ob} - x\|^2}{\beta})$, for some positive constants ρ, α and β . Note that $h(\cdot)$ is a decreasing function in $\|x_{ob} - x\|$. Different cost-functions can be imagined but as the main focus of the paper is to show that feasible robotic controllers can be provided in real-time without giving up on optimality, we do not elaborate on that here. To summarize, we have

$$L(x(t)) = \rho \|x_g - x(t)\|^2 + \alpha e^{-\frac{\|x_{ob} - x(t)\|^2}{\beta}}, \quad (1)$$

where ρ is the gain of the goal attraction term, α is the gain of the obstacle avoidance term, and β decides the range of the obstacle avoidance term. Solving problem $P1$ is in general a hard problem. Hence, in order to solve the obstacle avoidance problem and still be able to have the robot perform in real-time, some simplifications must be made. First, instead of controlling the robot through the control variable u , we assume that the robot is controlled by choosing among a set of desired modes/behaviors. We then get that $\dot{x} = f_i(x)$ for some $i \in I$, where I defines the set of behaviors. We consider using the following three behaviors: *go-to-goal*, *go-around-obstacle-clockwise*, and *go-around-obstacle-counterclockwise*. The respective behaviors

are denoted f_g , f_{\circ} and f_{\ominus} , and are given by

$$f_g(x) = c(x_g - x), \quad f_{\circ}(x) = v \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \frac{x_{ob} - x}{\|x_{ob} - x\|},$$

$$f_{\ominus}(x) = v \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \frac{x_{ob} - x}{\|x_{ob} - x\|},$$

where c and v are positive scalars and $f_{\circ}(x)$ and $f_{\ominus}(x)$ correspond to the robot moving in a circle around the obstacle in the given direction. Throughout the paper, c and v will be set to unity when not mentioned otherwise. It should be pointed out that we are interested in finding the general shape of the guard and that additional tuning of the parameters discussed above might be necessary depending on the application. Problem $P1$ is now reduced to the problem of deciding what behaviors to use and when to switch between them. To this end, we denote the set of all finite length sequences of the three behaviors $f_g, f_{\circ}, f_{\ominus}$ by Σ and let B be an element of Σ . The time at which we switch between the different behaviors in B is denoted by $\bar{\tau}$, e.g. if $B = (f_g, f_{\circ}, f_g)$ then $\bar{\tau} = (\tau_0, \tau_1, \tau_2, \tau_3)$, where τ_1 is the time when we switch between f_g and f_{\circ} , τ_2 is the time when we switch between f_{\circ} and f_g , $\tau_0 = 0$ is the initial time, and $\tau_3 = T$ is the final time. The simplified version of problem $P1$ that we propose to solve becomes

$$P2: \quad \min_{B, \bar{\tau}} J = \int_0^T L(x(t))dt$$

s.t. $\dot{x} = \begin{cases} B(1)(x), & 0 \leq t < \tau_1, \\ B(2)(x), & \tau_1 \leq t < \tau_2, \\ \vdots & \vdots \\ B(N)(x), & \tau_{N-1} \leq t < T, \end{cases}$

$$0 \leq \tau_1 \leq \dots \leq \tau_{N-1} \leq T,$$

where the dimension of $\bar{\tau}$ is induced by B , $B \in \Sigma$ and x belongs to some compact set X such that $x_{ob} \notin X$ (note that at $x = x_{ob}$, f_{\circ} and f_{\ominus} are not differentiable).

III. OPTIMAL CONTROL DERIVATION

In order to obtain a locally optimal solution to problem $P2$ we start by considering the following two subproblems

- 1) Given a sequence of functions B , find the optimal $\bar{\tau}$.
- 2) Given a sequence of functions B with the corresponding optimal $\bar{\tau}$, find if it is beneficial to insert f_g, f_{\circ} or f_{\ominus} for a short interval of time at some time $t \in (0, T)$.

It should be clear that if we can solve subproblems 1 and 2 we can obtain a locally optimal solution to problem $P2$ by first solving subproblem 1 and then search to see if it is beneficial to insert a new function by solving subproblem 2. Repeating this process will give a locally optimal solution to $P2$. In fact, a more general version of

these problems was recently solved in [9] by the authors, and, for the sake of completeness, we recall the major results in the following paragraphs.

Subproblem 1 is solved by deriving an expression for the gradient of the cost with respect to the switching vector, $\nabla J(\bar{\tau})$, and applying a gradient descent algorithm to find the optimal switching vector. An expression for $\nabla J(\bar{\tau})$ was derived in [9], where continuous differentiability of all behaviors and of the cost function $L(x)$ were assumed for $x \in X$. Assuming B consists of a string of N behaviors and relabel the elements of B by defining $f_i = B(i)$, $i = 1, \dots, N$, and setting $B = (f_1, \dots, f_N)$, the following assertion characterizes the derivatives $\frac{dJ}{d\tau_i}$, and hence the gradient $\nabla J(\bar{\tau}) = (\frac{dJ}{d\tau_1}, \dots, \frac{dJ}{d\tau_N})$:

Proposition 3: [9] *For every $i \in (1, \dots, N - 1)$ the following equation is in force,*

$$\frac{dJ}{d\tau_i} = p(\tau_i)^T (f_i(x(\tau_i)) - f_{i+1}(x(\tau_i))), \quad (2)$$

where the costate is given by the following backwards differential equation

$$\dot{p}(t) = - \left(\frac{df_i}{dx}(x(t)) \right)^T p(t) - \left(\frac{dL}{dx}(x(t)) \right)^T, \quad (3)$$

for all $t \in [\tau_{i-1}, \tau_i]$ and for every $i \in \{1, \dots, N\}$, with the given final condition $p(T) = 0$. ■

In order to solve subproblem 2 we need to evaluate how the cost changes if we insert a new behavior at a time t^1 for λ seconds. Inserting a new behavior corresponds to adding two new switching times to $\bar{\tau}$ and adding the inserted behavior to B . Consider a sequence of behaviors B and the associated state trajectory $x(t)$ obtained when solving subproblem 1. Assume that x evolves according to f_g at time t^1 . An expression for the change in the cost obtained by inserting a behavior b at time t^1 was given in [9], with the conclusion that

$$\lim_{\lambda \downarrow 0} \frac{dJ}{d\lambda}(b, t^1) = p(t^1)^T (b(x(t^1)) - f_g(x(t^1))). \quad (4)$$

If $\lim_{\lambda \downarrow 0} \frac{dJ}{d\lambda}(b, t^1) < 0$ and we add the two switching points, corresponding to inserting b at time t^1 , and optimize over the switching times, we will get a descent in J .

Having presented expressions for $\nabla J(\bar{\tau})$ and $\lim_{\lambda \downarrow 0} \frac{dJ}{d\lambda}$ we can now proceed to derive the locally optimal solution to the go-to-goal, obstacle-avoidance problem posed in $P2$. Initializing B to $B = (f_g)$ and calculating $x(t)$ forward with initial condition $x(0) = x_0$, and $p(t)$ backward through (3), with final condition $p(T) = 0$, we can then search to find the function $b \in (f_{\circlearrowleft}, f_{\circlearrowright})$ and the time $t \in [0, T]$ that minimizes (4). If a function b and a time t exists such that $\lim_{\lambda \downarrow 0} \frac{dJ}{d\lambda}(b, t) < 0$, we insert two new switching times, one at $t - dt$ and one at $t + dt$, were dt is a small positive constant, and update B and $\bar{\tau}$ to

$B = (f_g, b, f_g)$ and $\bar{\tau} = (0, t - dt, t + dt, T)$. We then optimize over B to find the optimal switching vector. After we have optimized over $B = (f_g, b, f_g)$ we could try to evaluate (4) again to see if it is beneficial to insert another function at some time $t \in [0, T]$. It turns out that with our particular choice of cost function, given by (1), we do not get any significant descent by performing a second insertion after we have optimized over B given by the first insertion. Therefore, for each obstacle, we only consider one insertion.

The solution to $P2$ is indeed locally optimal but it is not applicable to real-time robotics problems since we need to calculate $x(t)$ and $p(t)$ for each iteration in the optimization algorithm, and this is time consuming. We would like to obtain an optimal solution where the switching times, i.e. when to switch from f_g to f_{\circlearrowleft} or f_{\circlearrowright} and when to switch back to f_g for each obstacle, are given by a geometric guard defined around the obstacle. Moreover, the structure of the guard should only depend on the distance between the obstacle and the goal. In order to arrive at this result, it first needs to be proven that the solution is invariant along trajectories, given that the final time T is big enough. To illustrate this, consider a trajectory $\tilde{x}(t)$, $t \in (0, T)$ with a fixed initial state and a corresponding optimal switch at $\tilde{\tau} \in (0, T)$. If we initialize another trajectory, $\hat{x}(t)$, to start along the path of $\tilde{x}(t)$, that is $\hat{x}(0) = \tilde{x}(\Delta)$ where $\Delta \in (0, \tilde{\tau})$, then the optimal switch of $\hat{x}(t)$ should occur at $\tilde{\tau} - \Delta$ if the solution is invariant along trajectories. To this end, Lemma 3 is presented.

Lemma 3: Given an initial state x_0 , denote by $x(t)$ and by $p(t)$ the state and the costate trajectories obtained when $x(t)$ evolves according to $\dot{x} = h(x)$. Assume h and the costfunction $L(x(t))$ associated with p are continuously differentiable. Denote by $\bar{x}(t)$ and by $\bar{p}(t)$ the state and the costate trajectories associated with the same system, but with an initial condition along the trajectory of $x(t)$, i.e. $\bar{x}(0) = x(\Delta)$ for some $\Delta \in (0, T)$. Assume there exists a finite time T_1 s.t. $\frac{dh}{dx}(x(t))$ is negative definite $\forall t > T_1$, $L(x(t)) : X \rightarrow \mathbb{R}_+$ is bounded above by a constant C , and the final time $T \rightarrow \infty$. Then the state and the costate trajectories satisfy the following two equations

$$\bar{x}(t - \Delta) \rightarrow x(t) \quad t \in [\Delta, T], \quad (5)$$

$$\bar{p}(t - \Delta) \rightarrow p(t) \quad t \in [\Delta, T]. \quad (6)$$

Proof: (5) is true since h does not depend on the initial state and $\bar{x}(0) = x(\Delta)$, hence $x(t) = \bar{x}(t - \Delta)$ for all $t \in (\Delta, T)$ regardless of T . As for (6), we note that the costate for the first system is given by

$$p(t) = \int_t^T \frac{dL}{dx}(x(s)) \Phi(s, t) ds,$$

where $\Phi(s, t)$ is the state transition matrix of the linear, time-varying dynamical system $\dot{z} = \frac{\partial h(x(t))}{\partial x} z$. For the

second system, we have that $\bar{p}(t) = \int_t^T \frac{dL}{d\bar{x}}(\bar{x}(s))\bar{\Phi}(s, t)ds$, where $\bar{\Phi}(s, t)$ is the state transition matrix of the linear, time-varying dynamical system $\dot{z} = \frac{\partial h(\bar{x}(t))}{\partial \bar{x}}z$. Furthermore

$$\begin{aligned}\bar{p}(t - \Delta) &= \int_{t-\Delta}^T \frac{dL}{d\bar{x}}(\bar{x}(s))\bar{\Phi}(s, t - \Delta)ds = \\ &= \int_t^{T+\Delta} \frac{dL}{d\bar{x}}(\bar{x}(s - \Delta))\bar{\Phi}(s - \Delta, t - \Delta)ds,\end{aligned}$$

by a change of variables. Noting that $\bar{x}(s - \Delta) = x(s)$ and $\bar{\Phi}(s - \Delta, t - \Delta) = \Phi(s, t)$, we get that $\bar{p}(t - \Delta) = \int_t^{T+\Delta} \frac{dL}{dx}(x(s))\Phi(s, t)ds$. The following expression for the difference holds for all times $t \in (\Delta, T)$,

$$\begin{aligned}\|p(t) - \bar{p}(t - \Delta)\| &= \left\| \int_t^T \frac{dL}{dx}(x(s))\Phi(s, t)ds - \right. \\ &\left. - \int_t^{T+\Delta} \frac{dL}{dx}(x(s))\Phi(s, t)ds \right\| \leq C \int_T^{T+\Delta} \|\Phi(s, t)\|ds.\end{aligned}\quad (7)$$

The following equation is in force for $\Phi(s, t)$,

$$\frac{d}{ds}\Phi(s, t) = \frac{dh}{dx}(x(s))\Phi(s, t)\quad (8)$$

$$\Phi(t, t) = I.\quad (9)$$

As $\frac{dh}{dx}(x(s))$ is negative definite for $t > T_1$, $\lim_{s \rightarrow \infty} \Phi(s, t) = 0$ by (8) and (9). Since Δ and t are finite it follows from (7) that $\bar{p}(t - \Delta) \rightarrow p(t)$ as $T \rightarrow \infty$. ■

It should be noted that variants of Lemma 3 have appeared in the literature ([10], [11]). Nevertheless it is important for our future presentation and is therefore presented.

IV. GUARD GENERATION

As an application of Lemma 3, consider subproblem 1 and let $B = (f_g)$. Let $x^1(t)$ be the state trajectory associated with solving subproblem 1 starting at x_0 . Likewise, let $x^2(t)$ be the state trajectory when we start at $x^1(\Delta)$ for some time $\Delta \in (0, T)$. From Lemma 3 we know that (5,6) are in force and hence $x^1(t) \rightarrow x^2(t - \Delta)$ and $p^1(t) \rightarrow p^2(t - \Delta)$ for all finite times t as $T \rightarrow \infty$. Denote by J^1 the cost associated with $x^1(t)$ and by J^2 the cost associated with $x^2(t)$ and define $\bar{\Delta}$ to be the vector with the same dimension as $\bar{\tau}$ with each element equal to Δ . Since $\nabla J(\bar{\tau})$ and $\lim_{\lambda \downarrow 0} \frac{dJ}{d\lambda}$ only depend on the state $x(t)$ and the costate $p(t)$ it follows that both $\|\nabla J^1(\bar{\tau}) - \nabla J^2(\bar{\tau} - \bar{\Delta})\|$ and $|\lim_{\lambda \downarrow 0} \frac{dJ^1}{d\lambda}(b, t) - \lim_{\lambda \downarrow 0} \frac{dJ^2}{d\lambda}(b, t - \Delta)|$ are close to zero for all finite times $t \in (\Delta, T)$. The implication of this is that $\frac{dJ^1}{d\lambda}(b, t)$ will be minimized at the same time $\bar{t} \in (\Delta, T)$ and with the same behavior b as $\frac{dJ^2}{d\lambda}(b, t - \Delta)$. Hence, given that $\frac{dJ^1}{d\lambda}(b, t) < 0$, the insertion of a new behavior will occur at the same point for both systems. After the insertions we will still have $\nabla J^1(\bar{\tau}) = \nabla J^2(\bar{\tau} - \bar{\Delta})$. Hence the optimization algorithm will terminate at two distinct switching vectors: $\bar{\tau}^1$ associated with $x^1(t)$ and $\bar{\tau}^2$ associated with $x^2(t)$ such that $\bar{\tau}^1 = \bar{\tau}^2 - \bar{\Delta}$. Moreover, the switches occur at the same point in the state space.

Thus, we have shown that the solution to subproblem 1 is invariant along trajectories, e.g. if we start along a trajectory we will switch at the same points in the state space independently of where on the trajectory we start. This is exactly the result we need in order to generate the guards for when to switch from the *go-to-goal* behavior to the *obstacle-avoidance* behavior. Left to do is to show that problem P2 indeed does satisfy the assumptions in the lemma. We also want to prove that (5) and (6) are close to being true, in the sense that $\|x(t) - \bar{x}(t - \Delta)\|$ and $\|p(t) - \bar{p}(t - \Delta)\|$ are small, even if T is finite but big enough to guarantee that we indeed end up close to the goal.

In order to prove this we assume that the obstacle and the goal are far enough apart. This results in (1) being small at the final time since $x(T)$ is close to x_g . Furthermore, since $x \in X$, where the set X were defined in Section 2, $L(x(t))$ is bounded for all times $t \in (\Delta, T)$. Hence the constant C defined in (7) is small. Next, we need to prove the existence of a positive time T_1 such that if $x(t)$ evolves according to $\dot{x} = h(x(t))$ after time T_1 then $\frac{dh}{dx}$ is negative definite for all times $t \in (T_1, T)$. To show this we note that $\frac{df_g}{dx} = -cI$, where I denotes the identity matrix and $c > 0$, is negative definite. Next, two cases are needed in order to capture possible B s. The first case corresponds to $B = (f_g)$, i.e. we do not switch. In this case we can choose $T_1 = 0$ since $\frac{df_g}{dx}$ is negative definite. The second case corresponds to $B = (f_g, b, f_g)$, where $b = f_{\circlearrowleft}$ or $b = f_{\circlearrowright}$. Here we might choose $T_1 = \tau_2$ and by the assumption that the obstacle and the goal are far enough apart it follows that $T_1 < T$. Note that the case corresponding to $B = (f_g, b)$ is excluded by virtue of the same assumption. Since $\Phi(s, t)$ decays exponentially when we evolve according to $\dot{x} = f_g$, we conclude that $\|x(t) - \bar{x}(t - \Delta)\|$ and $\|p(t) - \bar{p}(t - \Delta)\|$ both will be close to zero even if T is finite.

To illustrate the discussion above, a solution to problem P2 for two different initial states (denoted by a star and a circle) along the same trajectory is depicted in Figure 2(a). Both trajectories evolve according to $B = (f_g)$ before the insertion. Here, the final time is 5 seconds, $x_{ob} = (0, 2)^T$, $x_g = (0, 4)^T$, and the values for ρ , α and β are $\rho = 1/100$, $\alpha = 2$, and $\beta = 0.1$. It is clear, from Figure 2(a), that the solution to problem P2 for both initial states switches at the same place in the state space even though the final time is finite. Furthermore, in Figure 2(b), $\lim_{\lambda \downarrow 0} \frac{dJ}{d\lambda}(f_{\circlearrowleft}, t)$ is depicted for both state trajectories. As seen from the figure, both curves are similar in shape but there is a time offset between them corresponding to the fact that the dotted curve starts along the trajectory of the solid curve. From Figure 2(b) it is moreover clear that the insertion will occur at the same point in the state space.

To summarize, we have proved that problem P2 satisfies

the assumptions of Lemma 3 and that the Lemma is valid for practical purposes even if the final time is finite. In

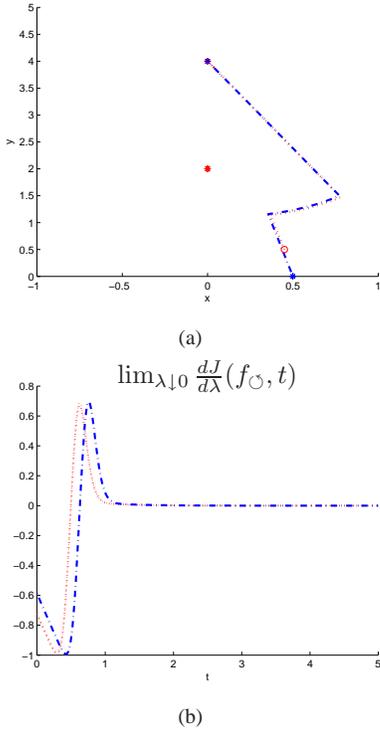


Fig. 2. Different Initial States: In the top figure, the optimal state trajectories for both initial states are shown. In the bottom figure $\lim_{\lambda \downarrow 0} \frac{dJ}{d\lambda}(f_{\circ}, t)$ is shown for both trajectories as functions of t when $B = (f_g)$.

order to get the data needed to generate the guard for a given distance between the obstacle and the goal we execute the following program:

- *Step 0:* Select a finite set of representative initial states X_0 and a small constant δt , and create a set of times $G = \{t \mid t \in [0, T] \text{ and } t = k\delta t, k = 1, 2, \dots\}$
- *Step 1:* If $X_0 = \emptyset$ STOP. Else, select an initial state $x_0 \in X_0$, calculate the state forward as described in problem $P2$ with initial condition $x(0) = x_0$, and $p(t)$ backwards through (3), with final condition $p(T) = 0$. Remove x_0 from X_0 .
- *Step 2:* Evaluate $\lim_{\lambda \downarrow 0} \frac{dJ}{d\lambda}(b, t)$ for $\forall t \in G$ and $b \in \{f_{\circ}, f_{\circ}\}$. If $\min_{b,t} \frac{dJ}{d\lambda}(b, t) \geq 0$ go to *Step 1*. Else, insert the one out of f_{\circ} and f_{\circ} that minimizes $\lim_{\lambda \downarrow 0} \frac{dJ}{d\lambda}(b, t)$ for an interval of length δt seconds around $r = \arg \min_t \{\lim_{\lambda \downarrow 0} \frac{dJ}{d\lambda}(b, t)\}$. Update B and \bar{r} .
- *Step 3:* Run the optimization algorithm to obtain a locally optimal switching vector. Go to *Step 1*.

An example of the state trajectories obtained by running this program is depicted in Figure 3(a), where $x_{ob} = (0, 2)^T$, $x_g = (0, 4)^T$, and the values for ρ , α and β are as before. By examining Figure 3(a), we see that whenever the robot is inside the region denoted by Guard I in Figure 3(b), it is

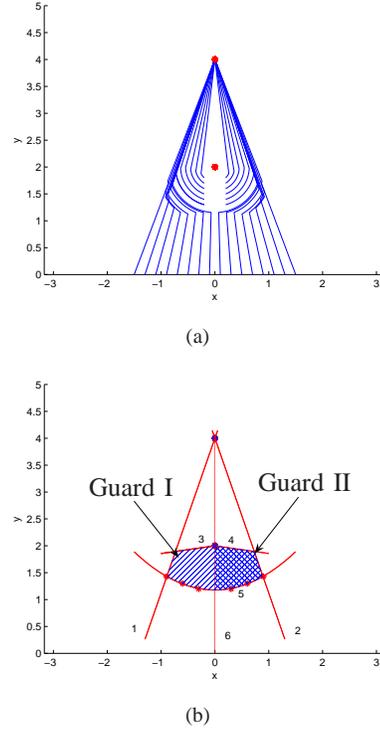


Fig. 3. State trajectories and associated guard structure: In the top figure the result for executing the program for a set of initial states along the x -axis is shown. In the bottom figure, an approximation of the associated guards using only lines and a parabola is depicted. Guard I is represented by the striped region and Guard II is represented by the boxed region. Number 1 to 6 represent the respective lines and the parabola used to approximate the guards.

optimal to let the robot evolve according to f_{\circ} . Likewise, if the robot is inside the region denoted by Guard II in Figure 3(b), it is optimal to let the robot evolve according to f_{\circ} . Everywhere else it is optimal to use f_g . The regions where it is beneficial to evolve according to f_{\circ} or f_{\circ} are approximated by a parabola and five lines as shown in Figure 3(b). At this point it should be noted that given the cost-function and a set of behaviors, the structure of the guards depends *only* on the distance between the goal and the obstacle. We denote the distance between the goal and the obstacle by d where $d = \|x_g - x_{ob}\|$. As we change d it might be conceivable that we no longer can approximate the guards the same way as done above, e.g. the error from assuming that the guards can be approximated with lines and parabolas might be substantial. Simulation shows that this is not the case for our range of distances. Hence, we denote by $a_{i,d}$ and $b_{i,d}$ the parameters for line 1 to 4, in Figure 3(b), such that line i is described by $y = a_{i,d}x + b_{i,d}$, $i = \{1, 2, 3, 4\}$ and denote by $a_{5,d}$ and $b_{5,d}$ the parameters of the parabola $y = a_{5,d}x^2 + b_{5,d}$. Here, subscript d denoted the distance between the goal and the obstacle. Then, under the assumption that the obstacle and the goal lie on the y -axis, the guard associated with f_{\circ} can

be expressed as

$$G_{f_{\odot},d} = \begin{cases} TRUE, & \text{if } \begin{cases} y < a_{1,d}x + b_{1,d} \text{ and} \\ y < a_{3,d}x + b_{3,d} \text{ and} \\ y > a_{5,d}x^2 + b_{5,d} \text{ and} \\ x < 0, \end{cases} \\ FALSE, & \text{otherwise.} \end{cases}$$

Likewise, the guard associated with f_{\ominus} is given by

$$G_{f_{\ominus},d} = \begin{cases} TRUE, & \text{if } \begin{cases} y < a_{2,d}x + b_{2,d} \text{ and} \\ y < a_{4,d}x + b_{4,d} \text{ and} \\ y > a_{5,d}x^2 + b_{5,d} \text{ and} \\ x > 0, \end{cases} \\ FALSE, & \text{otherwise.} \end{cases}$$

If the goal and obstacle do not line up, a simple rotation and translation of the above equations is needed. From the assumption that x_{ob} and x_g are far enough apart, it follows from the structure of problem $P2$ that there are only three distinct sequences of switching vectors that can occur in the optimal solution. These corresponds to $B = (f_g)$, $B = (f_g, f_{\odot}, f_g)$ and $B = (f_g, f_{\ominus}, f_g)$, from which it follows that we never switch between f_{\odot} and f_{\ominus} . Therefore the optimal solution is given in terms of the guards $G_{f_{\odot},d}$ and $G_{f_{\ominus},d}$ and the optimal solution can be cast on the form of Figure 4. This solution is suitable for realtime applications since the guards are easily stored and evaluated.

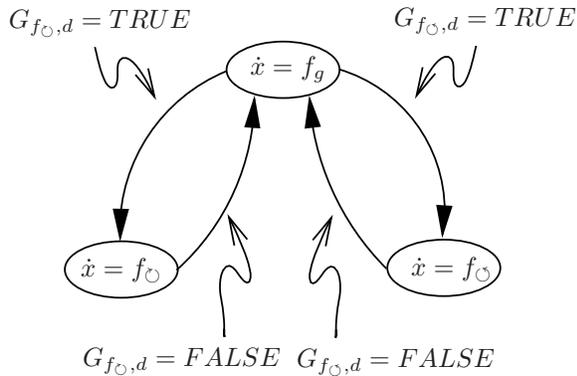


Fig. 4. The optimal solution to problem $P2$ given in terms of guards associated with each obstacle encountered in the robots path.

V. ROBOTICS IMPLEMENTATION

In order to verify that the proposed navigation system performs well when implemented on a real robotics platform, we tested it on the MAGELLAN PRO platform from IROBOT. In the experiment, the goal was located at $(0, 4)$, while the obstacle was at $(0.3, 2)$. To detect the obstacle the robot uses infrared sensors that have an effective range from 0 to 1 meter. When the robot gets inside one of the regions defined by the guards $G_{f_{\odot},2}$ and $G_{f_{\ominus},2}$ it switches to the respective behavior. From Figure 5 it can be seen that the robot detects the obstacle, and then switches to f_{\odot} to avoid the obstacle. The robot switches back to the *go-to-goal*-behavior when the guard predicate no longer is true. At

this point it should be noted that even though our proposed model performs well in experiments, we do not claim to have solved any of the classical problems [1] associated with the reactive approach.

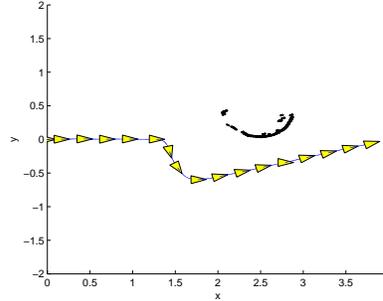


Fig. 5. Robot Experiment: The robot's path together with the sensor data is shown.

VI. CONCLUSIONS

In this paper we presented a technique for computing guards using optimal timing control. In particular, we showed how to construct switching surfaces that dictate when to switch between *go-to-goal* and *avoid-obstacle* in reactive navigation systems for mobile robots. Once the initial computational price has been paid in simulation, the guards can be implemented on a real platform with very little computational overhead.

REFERENCES

- [1] R. Arkin, *Behavior-Based Robotics*, MIT Press, Cambridge, Massachusetts, 1998.
- [2] J. Lygeros, C. Tomlin and S. Sastry, "Controllers for reachability specifications for hybrid systems", *Automatica*, **35**, pp. 349-370.
- [3] M. Egerstedt, K.H. Johansson, J. Lygeros and S. Sastry, "Behavior Based Robotics Using Regularized Hybrid Automata", *IEEE Conference on Decision and Control*, Phoenix, AZ, 1999.
- [4] J.H. Reif and H. Wang, "Social Potential fields: A distributed behavioral control for autonomous robots", *Robotics and Autonomous Systems*, **27**, 171-194, 1999.
- [5] E.W. Large, H.I. Christensen and R. Bajcsy, "Dynamic Robot Planning: Cooperation through Competition", *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Vol. 35, pp. 2306-2312, 1997.
- [6] D. Kortenkamp, R.P. Bonasso and R. Murphy, *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, AAAI Press, Cambridge, Massachusetts, 1998.
- [7] Y. Wardi, M. Egerstedt, M. Boccadoro and E. Verriest, "Optimal Control of Switching Surfaces", in *43rd IEEE Conference on Decision and Control*, Atlantis, Bahamas, 2004.
- [8] M. Boccadoro, M. Egerstedt and Y. Wardi, "Optimal Control of Switching Surfaces in Hybrid Dynamic Systems", *IFAC Workshop on Discrete Event Systems*, Reims, France, Sept. 2004.
- [9] M. Egerstedt, Y. Wardi, and H. Axelsson, "Optimal Control of Switching Times in Hybrid Systems", *9th IEEE International Conference on Methods and Models in Automation and Robotics*, Miedzysdroje, Poland, 2003.
- [10] A.E. Bryson, Jr. and Y.-C. Ho, *Applied Optimal Control*, Ginn and Co., 1969.
- [11] E.B. Lee and L. Markus, *Foundations of Optimal Control Theory*, Wiley, New York, 1967.