Hybrid Modelling and Reachability on Autonomous RC-Cars

Tony Wood, Peyman Mohajerin Esfahani, and John Lygeros *

* *The authors are with the Automatic Control Laboratory, ETH Zürich, Physikstrasse 3, 8092 Zürich, Switzerland. Emails: woodt@ethz.ch, {mohajerin,lygeros}@control.ee.ethz.ch*

Abstract: In this paper we apply reachability analysis to design a controller for an RC-car to drive autonomously on a given circuit. We introduce a hybrid simplification technique to reduce the order of the model; this is crucial for reachability analysis. For a successful implementation on a real system the control problem is divided into two parts: a reachability control strategy is derived for the simplified hybrid model, and a gain-scheduled controller lets the full system track the simplified behaviour. We propose a heuristic algorithm to synthesise a hybrid feedback policy. By considering stochasticity in the model, we improve the performance of the controller which is finally validated on a real physical system.

1. INTRODUCTION

The objective considered in this work is the design and implementation of a method for autonomous control of a remote controlled (RC) car. In particular, the car should drive on a given circuit and reach a predefined target region. The hardware of the control loop contains a 1:43 scale model car, a vision system and a computer system on which the controller is to be implemented; for details we refer to Jones et al. [2010].

A possible approach to address this objective is to apply reachability analysis. More specifically, we consider a so-called reach-avoid problem. This problem consists of determining all initial states from which there exists a control strategy such that the state of the system eventually reaches a target set while not entering an avoid set. We call the set of such initial conditions the reachable set.

There are different methods to investigate reachability. In a direct approach, as in Cardaliaguet [1996], Cardaliaguet et al. [2002] the reachable set is formulated in the context of viability theory. We will however consider an alternative indirect approach that involves level set methods defined by value functions that characterise appropriate optimal control problems. These value functions can be expressed as the viscosity solutions to the standard Hamilton-Jacobi-Bellman (HJB) equations by applying dynamic programming techniques as in Lygeros [2004], Mitchell et al. [2001]. These techniques have been studied further in Fialho and Georgiou [1999], Margellos and Lygeros [2009] in the presence of state constraints and disturbance inputs respectively. Based on these level set methods, numerical algorithms have been developed by Osher and Sethian [1988], Sethian [1999] and have been implemented in efficient computational tools in Mitchell et al. [2001], Mitchell and Tomlin [2000].

In the process of computing the reachable set, a control strategy that steers the system in the desired way can be derived heuristically. Numerically the controller is implemented as a lookup table on a discrete grid. The model used in the design of other controllers for this hardware setup in Wunderli [2011] results in a lookup strategy that is too large to implement on the real system.

In order to deal with this practical limitation, we propose a model reduction technique based on a hybrid framework. Namely, given a convenient particular structure of the nominal model, we introduce a finite set of discrete modes and disregard two states of the full model: one state is set to be a parameter constant in each mode and one state is modelled as the input to the simplified system. We do the reachability analysis on the simplified model, which results in a smaller lookup table on a lower dimensional grid. In a separate task we control the full system to follow the desired simplified behaviour.

Due to the hybrid nature of the simplified system, appropriate notions for time, solutions, reachability and controller synthesis need to be considered. Properties of hybrid models are discussed in Tomlin et al. [1998]. Reachability in a hybrid context is studied in Gao et al. [2007], Lygeros et al. [1999]. We will introduce algorithms to compute the reachable set and synthesise a control strategy for the simplified hybrid system.

The reachability analysis on the simplified model may lead to policies that are not trackable by the dynamics of the full system. Therefore, for a robust controller, a strategy that tends to lead the system away from states that are in some sense close to the avoid set is beneficial. In other words, we aim to differentiate between states in the reachable set according to how "safe" they are. This can be achieved by introducing noise to the dynamics in each discrete mode and extracting a reachability controller from the resulting stochastic systems. The evolution of such a

⋆ This research has been supported by the European Commission under the MoVeS project, FP7-ICT-2009-257005. The test platform ORCA has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number FP7-ICT-2009-4 248940.

stochastic system is modelled by a stochastic differential equation (SDE). Stochastic reach-avoid problems are addressed in Esfahani et al. [2011] and discussed in more detail in Esfahani et al. [2012].

For the tracking of the controlled simplified behaviour by the full system, we consider gain-scheduling control. This is an adaptive control technique covered in Khalil [1992].

In Section 2 we formulate the problem in detail, and then introduce the full model and reachability for deterministic continuous systems. We proceed by defining a simplified hybrid model and discussing the evolution of such a hybrid system in Section 3. We then propose algorithms to compute the reachable set and a corresponding control strategy. In Section 4 we model stochastic systems for each discrete mode of the hybrid system. For each of these stochastic systems, we derive a robust control policy. The control of the full nominal system to track the simplified behaviour is discussed in Section 5. We conclude by illustrating the results of the implementation on the actual physical system in Section 6.

2. MODEL DESCRIPTION AND BASIC DEFINITIONS

The goal is that the car reaches a set of terminal positions from any initial position while staying on the track along the way. This can be formulated as a reach-avoid problem: reach the terminal set $A_p \subset \mathbb{R}^2$ while avoiding the set of positions lying outside the circuit $B_p \subset \mathbb{R}^2$. The terminal and avoid set can be defined in a two dimensional state space. However, the problem is subject to the motion dynamics which are modelled as a four dimensional system. Before we discuss the details of the model, we first introduce some useful definitions.

2.1 Solutions and Reachable States

Consider a continuous time dynamic system expressed in state space description as an ordinary differential equation (ODE)

$$\frac{\mathrm{d}\xi}{\mathrm{d}t} = f(\xi, u); \tag{1}$$

where the vector field $f : \mathcal{X} \times U \to \mathbb{R}^n$ governs the rate of change of the the state $\frac{\mathrm{d}\xi}{\mathrm{d}t} \in \mathbb{R}^n$, in dependence of the state itself $\xi \in \mathcal{X} \subseteq \mathbb{R}^n$ and the input $u \in U \subseteq \mathbb{R}^m$.

The set of solutions $\mathfrak{S}_f^{\xi_0}$ of a system with vector field f and initial state ξ_0 is roughly speaking the set of continuous functions $\xi(\cdot)$ such that for all times $t \geq 0$ there exists a feasible input $u \in U$ such that these functions solve the differential equation $\frac{\mathrm{d}\xi}{\mathrm{d}t}(t) = f(\xi(t), u)$ with initial condition $\xi(0) = \xi_0$.

Definition 1. (Set of Solutions).

$$\mathfrak{S}_f^{\xi_0} := \left\{ \xi(\cdot) \in \mathcal{C}([0,\infty), \mathbb{R}^n) \,\middle|\, \forall t \geq 0 \, \exists u \in U \text{ s.t. } \frac{\mathrm{d}\xi}{\mathrm{d}t}(t) = f(\xi(t), u),\, \xi(0) = \xi_0 \right\}.$$

For a rigorous discussion of solutions of ordinary differential equations, we refer to Khalil [1992]. The set of reachable states $R_f(A, B)$ is the set of initial conditions for which there exists a trajectory $\xi(\cdot)$ in the set of solutions that hits the terminal set A before the avoid set B.

Definition 2. (Reachable Set).

$$R_f(A, B) := \{ z \in \mathcal{X} \mid \exists \xi(\cdot) \in \mathfrak{S}_f^z,\, t \geq 0 \text{ s.t. } \xi(t) \in A,\, \xi(s) \notin B \, \forall s \leq t \},$$

where $A, B \subset R^n$.

2.2 Physical Model Description

In previous projects the RC-cars have been modelled. We consider the simplest model used in Wunderli [2011] where the state space $\mathcal{X}$ is four dimensional. The state vector $\xi \in \mathcal{X} \subset \mathbb{R}^4$ consists of the coordinates of the position of the rear axle centre (x, y), the orientation of the vehicle ψ and its velocity v; thus we have $\xi := [x \ y \ \psi \ v]^T$. There are two inputs to the system: the steering angle δ and the duty cycle of the DC motor d, $u := [\delta \ d]^T$, $u \in U \subset \mathbb{R}^2$. The dynamics are defined by the vector field $f : \mathcal{X} \times U \to \mathbb{R}^4$ as

$$f(\xi, u) := \begin{bmatrix} v\cos(\psi) \\ v\sin(\psi) \\ \frac{1}{L}\delta v \\ (C_{m1} - vC_{m2})\, d - C_d v^2 - C_r - C_\delta \delta^2 v^2 \end{bmatrix} \tag{2}$$

with $C_{m1} = 5.5$, $C_{m2} = 1.4$, $C_d = 0.15$, $C_r = 1.1$, $C_\delta = 1.1$ and $L = 0.062$. Naturally, the orientation is 2π-periodic $\psi \in [0, 2\pi)$, $\psi(t) = \big(\psi(0) + \int \frac{1}{L}\delta(s)v(s)\,\mathrm{d}s\big) \bmod 2\pi$. The model holds for small speeds and steering angles. Thus, we introduce a state constraint $v \in [0, 3]$, and input constraints $\delta \in [-\pi/4, \pi/4]$ and $d \in [-1, 1]$. Note that all numeric values in this paper are in SI units.

2.3 Control Strategy

With the car dynamics given in (2), the task of reaching a goal set while staying on the circuit is a four dimensional reach-avoid problem. The terminal set is $A = \{\xi \in \mathcal{X} \mid (x, y) \in A_p\}$ and the avoid set is $B = \{\xi \in \mathcal{X} \mid (x, y) \in B_p\}$. To design a controller for this task, we compute the reachable set $R_f(A, B)$ introduced in Definition 2 and in the process of this computation derive a feedback control strategy $\mathbf{u}(\xi)$. Let $\mathfrak{R}_f(A, B)$ be an operator that returns the reachable set and a corresponding control strategy $\mathbf{u}$:

$$\mathbf{u} : R_f(A, B) \to U, \quad \mathfrak{R}_f(A, B) := \big(R_f(A, B), \mathbf{u}(\cdot)\big). \tag{3}$$

The computation can be done by solving an appropriate partial differential equation (PDE) as done in Mitchell and Tomlin [2000]. In this work a finite time horizon is considered. For the problem in this paper we take an infinite horizon into account by letting the horizon parameter grow until the reachable set saturates. An infinite horizon allows us to heuristically compute a time invariant state feedback control strategy.

For the numeric computation a level set method Matlab toolbox, Mitchell and Templeton [2005], can be used. This numeric PDE solving method involves gridding and therefore suffers from the curse of dimensionality. When we use this four dimensional model, the lookup table capturing the control strategy is too big to implement on the existing hardware setup. To design an implementable controller, we will therefore reduce the model and formulate the problem in a lower dimensional state space.

3. MODEL SIMPLIFICATION

We need to reduce the order of the model to be able to obtain a lookup table that can be implemented on the physical system. For a lower order model to be useful, the resulting simplified behaviour has to be trackable by the full dynamics. Then, we can solve a reach-avoid problem with this reduced model and use a separate control scheme (see Section 5) to make the system track the desired references.

Since the original objective of reaching a terminal set of positions A_p, while not entering an avoid set of positions B_p, is two dimensional, it is reasonable to suggest a two dimensional simplified model. The obvious choice for the state is the position $\xi := [x\ y]^T$. Note that we use ξ to denote the continuous state in both the full physical model and the simplified model, despite it having a different physical meaning and the corresponding continuous state space $\mathcal{X} \subseteq R^n$ being of different order n. Similarly, the notation $u \in U \subseteq R^m$ will stand for the input of the full and simplified system regardless of what it is physically and its dimension m.

In the full model introduced in Section 2.2 the inputs only act on the orientation ψ and speed v directly. This specific structure of the nominal model allows us to consider the following two ideas for simplification: neglect the dynamics of ψ and v treating them either as inputs or, more restrictively, as constant parameters.

Empirically we see that modelling ψ and v as inputs to the simplified system does not work because the resulting optimal strategy for v is not trackable. We can however model just ψ as an input and v as a parameter of the dynamics. If the velocity v is to be constant, it needs to be selected small enough that the car can drive around the sharp turns of the track. This results in slow driving on parts of the circuit where higher speeds would be possible. Therefore, we design k different regions on the race track. In each of these regions the parameter v can take on a different constant value. This allows us to express a hybrid model with two dimensional continuous state space in Section 3.1.

In Section 3.3 we formulate the task at hand as a hybrid reach-avoid problem, define the reachable set in a hybrid context, and introduce an algorithm to compute it.

3.1 Hybrid Model Description

Let the continuous state $\xi = [x\ y]^T$ be able to take on values in the entire real plane $\xi \in \mathcal{X} := \mathbb{R}^2$. With the velocity v being a piecewise constant parameter taking on k different values, we introduce an additional discrete state $q \in Q = \{1, 2, ..., k\}$ making the state space $S = Q \times \mathcal{X}$ hybrid. The input to the hybrid system is the orientation only, $u := \psi \in U$, where the control set is now $U := [0, 2\pi)$.

The hybrid system is characterised by three functions: the vector field $f : Q \times \mathcal{X} \times U \to \mathbb{R}^2$, the domain map $\mathrm{Dom} : Q \to 2^{\mathcal{X}}$, and the guard map $G : Q \times Q \to 2^{\mathcal{X}}$, where $2^{\mathcal{X}}$ denotes the power set of $\mathcal{X}$. The vector field f determining the dynamics in dependence of the hybrid state (q, ξ) and the input u is defined as

$$f(q, \xi, u) := \begin{bmatrix} v_q \cos(\psi) \\ v_q \sin(\psi) \end{bmatrix} \tag{4}$$

with constants $v_q \in \{v_1, ..., v_k\}$, $0 < v_q \leq 3$. We will also use the shorthand $f_q(\xi, u) := f(q, \xi, u)$.

The race track is made up of k overlapping regions $D_1, D_2, ..., D_k \subseteq \mathbb{R}^2$ (Figure 1). The domain of a discrete state $q \in Q$ is the union of the region D_q with all positions outside the circuit; more formally, for all $q \in Q$, $\mathrm{Dom}(q) := D_q \cup (D_1 \cup D_2 \cup \cdots \cup D_k)^c$; where D^c denotes the the complement of a set D. The guard of a discrete state transition from $q \in Q$ to $g \in Q$ is the domain of q; that is $G(q, g) := \mathrm{Dom}(g)$, for all $q, g \in Q$.

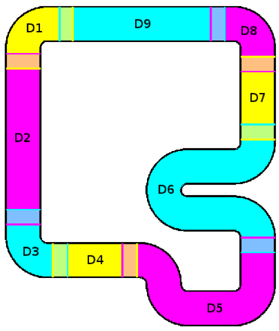


Fig. 1. Regions of the circuit

Standard definitions of hybrid systems contain an additional reset function that determines the continuous state after a discrete transition as in Tomlin et al. [1998]. For this model the reset map $r : Q \times Q \times \mathcal{X} \to \mathcal{X}$ is the identity map for all transitions, $r(q, g, \xi) = \xi\ \forall q, g \in Q$, and is therefore not considered further. Overall, this model can be seen as a special case of a hybrid game automaton defined in Gao et al. [2007]. Hybrid time sets, trajectories and runs are also specified in that work. We will define a similar concept next, before discussing hybrid reachability in Section 3.3.

3.2 Hybrid Time Sets, Trajectories and Executions

To characterise the evolution of the hybrid system, we need to introduce generalised notions of time and trajectory that capture both the continuous change of the continuous state and the transitions of the discrete state as in Gao et al. [2007].

Definition 3. (Hybrid Time Set). A hybrid time set $\tau = \{I_i\}_{i=0}^N$ is a finite or infinite sequence of intervals of the real line, such that

- for all $i < N$, $I_i = [\tau_i, \tau_i']$;
- if $N < \infty$, then either $I_N = [\tau_N, \tau_N')$ (possibly with $\tau_N' = \infty$), or $I_N = [\tau_N, \tau_N']$;
- for all i, $\tau_i \leq \tau_i' = \tau_{i+1}$.

Without loss of generality we can assume that $\tau_0 = 0$.

Definition 4. (Hybrid Trajectory). A hybrid trajectory over a set of variables taking values in a set A is a pair (τ, a) where $\tau = \{I_i\}_{i=0}^N$ is a hybrid time set and $a = \{a_i(\cdot)\}_{i=0}^N$ is a sequence of functions $a_i(\cdot) : I_i \to A$.

Now we can define a hybrid execution which will be the hybrid object corresponding to a solution for continuous systems.

Definition 5. (Execution). A execution of the hybrid system introduced in Section 3.1 is a hybrid trajectory (τ, q, ξ, u) over the state and input variables that satisfies the following conditions:

- Discrete Evolution: for $i < N$,
  (1) $\xi_i(\tau_i') \in G(q_i(\tau_i'), q_{i+1}(\tau_{i+1}))$.
  (2) $\xi_{i+1}(\tau_{i+1}) = \xi_i(\tau_i')$.
- Continuous Evolution: for all i with $\tau_i < \tau_i'$,
  (1) $u_i(\cdot)$ is a Lebesgue measurable function on I_i.
  (2) $q_i(t) = q_i(\tau_i)$ for all $t \in I_i$.
  (3) $x_i(\cdot)$ is the solution of the differential equation
  $$\frac{\mathrm{d}\xi_i}{\mathrm{d}t}(t) = f(q_i(t), \xi_i(t), u_i(t))$$
  over the interval I_i with initial condition $\xi_i(\tau_i)$.
  (4) $\xi_i(t) \in \mathrm{Dom}(q_i(t))$ for all $t \in [\tau_i, \tau_i')$.

Since the discrete state $q_i(\cdot)$ remains constant during continuous evolution, we can use the shorthand $q_i := q_i(t)$ for all $t \in I_i$.

Definition 6. (Set of Finite Executions). An execution is classified as finite if τ is a finite sequence ending with a compact interval. The set of finite executions starting at initial continuous state ξ_0 is denoted as $\mathfrak{E}^{\xi_0}$.

3.3 Hybrid Reach-Avoid Problem

With the model introduced in Section 3.1, the reachability investigation needs to be formulated in a hybrid context. Therefore, we define a new reachable set for the hybrid system, $R^{\mathrm{H}}(A, B)$. Consider the terminal set of positions $A = A_p \subset \mathcal{X}$ and the set of avoid positions $B = B_p = (D_1 \cup D_2 \cup \cdots \cup D_k)^c \subset \mathcal{X}$. The reachable set is the set of initial continuous states for which there exits a finite execution (τ, q, ξ, u) such that the continuous state $\{\xi_i(\cdot)\}_{i=0}^N$ hits the terminal set A before the avoid set B.

Definition 7. (Reachable Set for the Hybrid System).
$$\begin{aligned} R^{\mathrm{H}}(A, B) := \{ & z \in \mathcal{X} \mid \exists (\tau, q, \xi, u) \in \mathfrak{E}^z \\ & \text{s.t. } \xi_N(\tau_N') \in A,\, \xi_i(t) \notin B \,\forall I_i \in \tau \,\forall t \in I_i \}. \end{aligned}$$

The reachable set $R^{\mathrm{H}}(A, B)$ can be computed with the following algorithm applying tools introduced in Section 2.1.

Algorithm 1. (Computation of Reachable Set).

$A_0 := A$, $l = 0$
repeat
  $A_{l+1} := \bigcup_{j \in Q} R_{f_j}\left(A_l \cap D_j, D_j^c\right)$
  $l \leftarrow l + 1$
until $A_l = A_{l-1}$

We define A_∞ as the union of all A_l and claim that it is equal to the reachable set $R^H(A, B)$ introduced in Definition 7.

Theorem 8. (Reachable Set).
$$A_\infty := \bigcup_{l=0}^{\infty} A_l = R^{\mathrm{H}}(A, B).$$

Proof. We first show that $A_\infty \supseteq R^{\mathrm{H}}(A, B)$. Consider an arbitrary reachable state $z \in R^{\mathrm{H}}(A, B)$. From Definition 7 it follows that there exists a finite execution (τ, q, ξ, u) such that $\xi_0(\tau_0) = z$, $\xi_N(\tau_N') \in A$ and $\xi_i(t) \notin B$ for all $I_i \in \tau$ and $t \in I_i$. Thus, for such an execution, we have $\xi_N(\tau_N') \in A_0 := A$ and at the N-th discrete transition we observe that $\xi_{N-1}(\tau_{N-1}') = \xi_N(\tau_N) \in R_{f_{q_N}}\left(A_0 \cap D_{q_N}, D_{q_N}^c\right) \subseteq A_1 := \bigcup_{j \in Q} R_{f_j}(A_0 \cap D_j, D_j^c)$. In the same way we can state that if $\xi_{N-l}(\tau_{N-l}') \in A_l$, it follows that $\xi_{N-(l+1)}(\tau_{N-(l+1)}') = \xi_{N-l}(\tau_{N-l}) \in R_{f_{q_{N-l}}}(A_l \cap D_{q_{N-l}}, D_{q_{N-l}}^c) \subseteq A_{l+1}$. By induction it follows that $\xi_0(\tau_0') \in A_N$ and that $z = \xi_0(\tau_0) \in A_{N+1} \subseteq A_\infty$. Thus, $A_\infty \supseteq R^{\mathrm{H}}(A, B)$.

To show that $A_\infty \subseteq R^{\mathrm{H}}(A, B)$, we now consider an arbitrary $z \in A_\infty$. Since $A_\infty = \bigcup_{i=0}^{\infty} A_i$, there exists an integer l such that $z \in A_l := \bigcup_{j \in Q} R_{f_j}\left(A_{l-1} \cap D_j, D_j^c\right)$. It follows that there exists a discrete state $j \in Q$ such that $z \in R_{f_j}\left(A_{l-1} \cap D_j, D_j^c\right)$. From the definition of $R_f(\cdot, \cdot)$, Definition 2, we know that there exists a trajectory of the continuous state $\Xi(\cdot) \in \mathfrak{S}_{f_j}^z$ with corresponding input trajectory $\upsilon(\cdot)$ and a time $t \geq 0$ such that $\Xi(0) = z$, $\Xi(t) \in A_{l-1} \cap D_j$ and $\Xi(s) \in D_j \subseteq B^c$ with $\upsilon(s) \in U$ for all $s \in [0, t]$. Since for all $s \in [0, t]$ the solution $\Xi(s)$ remains in the domain of mode j, $\Xi(s) \in D_j \subseteq \mathrm{Dom}(j)$, we can start constructing a hybrid trajectory by letting $I_0 = [\tau_0, \tau_0'] = [0, t]$, $q_0 = j$, $\xi_0(\cdot) = \Xi(\cdot)$ and $u_0(\cdot) = \upsilon(\cdot)$. Then, for the next interval we have $\xi_1(\tau_1) = \xi_0(\tau_0') \in A_{l-1}$. Similarly, there exists a discrete state $q_1 \in Q$ such that $\xi_1(\tau_1) \in R_{f_{q_1}}\left(A_{l-2} \cap D_{q_1}, D_{q_1}^c\right)$. Furthermore, there exists an interval of continuous evolution in mode q_1: $I_1 = [\tau_1, \tau_1']$, $\xi_1(\cdot) \in \mathfrak{S}_{f_{q_1}}^{\xi_1(\tau_1)}$ and $u_1(\cdot)$ such that $\xi(\tau_1') \in A_{l-2} \cap D_{q_1}$ and for all $s \in I_1$ we have $\xi_1(s) \in D_{q_1} \subseteq B^c$ and $u_1(s) \in U$. Note that $\xi_0(\tau_0') \in G(q_0, q_1)$ and that $\xi_1(s) \in \mathrm{Dom}(q_1)$ for all $s \in I_1$. Repeating this process inductively leads to a finite execution $(\tau, q, \xi, u) \in \mathfrak{E}^z$ with initial continuous state z and l number of intervals such that $\xi_{l-1}(\tau_{l-1}') \in A_0 = A$ and $\xi_i(t) \notin B$ for all $i \in 0, 1, 2, \ldots, l-1$ and $t \in I_i$. By definition 7 we have $z \in R^{\mathrm{H}}(A, B)$. Thus, $A_\infty \subseteq R^{\mathrm{H}}(A, B)$. ■

Considering the specific choice of the k overlapping regions on the track and the modelled hybrid dynamics, we can state the following fact.

Fact 9. For this specific problem with dynamics (4), Algorithm 1 converges in at most k iterations since all positions on the circuit lie in the reachable set, $R^{\mathrm{H}}(A, B) = B^c$.

3.4 Construction of a Hybrid Feedback Controller

The feedback controller in the hybrid setup is dependent on the continuous and discrete state, $\mathbf{u}^{\mathrm{H}} : Q \times \mathcal{X} \to U$. In this section we propose a heuristic algorithm to synthesise a hybrid feedback policy in the process of computing the reachable set. Applying the operator $\mathfrak{R}_f(A, B)$ introduced in (3), we can rewrite Algorithm 1 to include the computation of the control law $\mathbf{u}^{\mathrm{H}}(q, \xi)$.

Algorithm 2. (Hybrid Controller Construction)

$A_0 := A$, $l = 0$
for all $(q, x) \in Q \times \mathcal{X}$ **do**
  $\mathbf{u}_0^{\mathrm{H}}(q, \xi) = \mathrm{NaN}$
end for
repeat
  for all $j \in Q$ **do**
    $\left(R_{f_j}\left(A_l \cap D_j, D_j^c\right), \mathbf{u}_j(\cdot)\right) = \mathfrak{R}_{f_j}\left(A_l \cap D_j, D_j^c\right)$
  end for
  $A_{l+1} := \bigcup_{j \in Q} R_{f_j}\left(A_l \cap D_j, D_j^c\right)$

10: **for all** $(q,\xi) \in Q \times \mathcal{X}$ **do**
11: **if** $\xi \in A_{l+1}$ **and** $\mathbf{u}^{\mathrm{H}}_{l}(j,\xi) = \mathrm{NaN}\ \forall j \in Q$ **then**
12: $\mathbf{u}^{\mathrm{H}}_{l+1}(q,\xi) = \mathbf{u}_q(\xi)$
13: **else**
14: $\mathbf{u}^{\mathrm{H}}_{l+1}(q,\xi) = \mathbf{u}^{\mathrm{H}}_{l}(q,\xi)$
15: **end if**
16: **end for**
17: $l \leftarrow l+1$
18: **until** $A_l = A_{l-1}$
19: $\mathbf{u}^{\mathrm{H}}(\cdot,\cdot) = \mathbf{u}^{\mathrm{H}}_{l}(\cdot,\cdot)$

where NaN stands for an object not being a number.

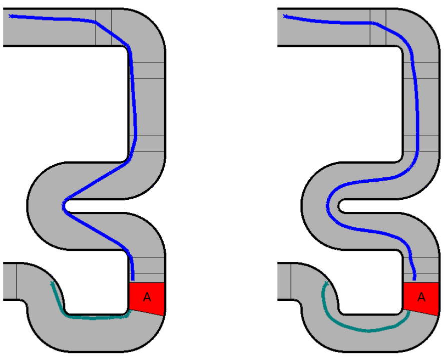
Note that we do not specify the order in which the iteration over the discrete state q in the for-loop starting in line 10 takes place. This order does however have influence on the resulting control strategy $\mathbf{u}^{\mathrm{H}}$. Regardless of the applied order of iteration, for every reachable continuous state $\xi \in R^{\mathrm{H}}_{f}(A,B)$ there always exists a unique discrete state j, for which the resulting control law $\mathbf{u}^{\mathrm{H}}$ is defined:

$$\forall \xi \in R^{\mathrm{H}}_{f}(A,B) \quad \exists! j \in Q \text{ s.t. } \mathbf{u}^{\mathrm{H}}(j,\xi) \neq \mathrm{NaN}\,. \tag{5}$$

Thus for a given reachable continuous state, the control function $\mathbf{u}^{\mathrm{H}}$ contains information on what input u is to be applied and also in what mode q the hybrid system is:

$$q(\xi) = j,\ u(\xi) = \mathbf{u}^{\mathrm{H}}(q(\xi),\xi), \tag{6}$$

where j is the unique mode in (5) for which $\mathbf{u}^{\mathrm{H}}(\cdot,\xi)$, with $\xi \in R^{\mathrm{H}}_{f}(A,B)$, is defined.



(a) 2D-simulation with a controller derived from deterministic reachability

(b) 2D-simulation with a controller derived from stochastic reachability

Fig. 2. Simulation of the simplified system with a controller derived from deterministic reachability (a) and stochastic reachability (b) starting at two different initial positions.

In practice, a controller designed using the deterministic model leads to problems as it does not differentiate between different points of the reachable set. As an illustration, one may obtain a strategy to drive close to the edge of the road in corners (see Figure 2 (a)). Moreover, we empirically observe that instantaneous changes in orientation resulting from the deterministic nature of the model are hard to track. From a practical point of view, it is clear that the difference in behaviour of the full system compared to the simplified system is more problematic the closer the continuous state is to the avoid set. For better results there should therefore be some measure of how "safe" a state is. This motivates the addition of some noise to the simplified dynamics in Section 4. The additional stochasticity also has a positive effect on the trackability of the lower order system; the resulting reference for the orientation is smoother.

4. INCREASING ROBUSTNESS BY INTRODUCING NOISE

In the methods discussed so far we have examined the state space to determine for each continuous state whether it lies in the reachable set or not. To have more robust control we would like to differentiate between the reachable states such that we can quantify how likely it is to reach the terminal set while not hitting the avoid set starting at a particular state. To this end, we add some noise to the deterministic dynamics given in (4). For each discrete mode $q \in Q$, the dynamics are modified by adding a stochastic term modelled by a two dimensional Brownian motion. This leads to a new type of continuous evolution characterised by a SDE in each mode. The individual modes can therefore be considered as stochastic continuous systems. In Section 4.1 we formalise such a system and in Section 4.2 formulate a stochastic reach-avoid problem according to Esfahani et al. [2011]. Quantifying in some sense how "safe" states are allows us to derive a controller for the simplified hybrid system in Section 4.3 that can be tracked by the full system more reliably.

4.1 Stochastic Continuous Time Systems

Consider a probability space $(\Omega,\mathcal{F},\mathbb{P})$ whose filtration $\mathbb{F} = (\mathcal{F}_s)_{s\geq 0}$ is generated by the n-dimensional Brownian motion $(W_s)_{s\geq 0}$ adapted to $\mathbb{F}$. Let the natural filtration of the Brownian motion $(W_s)_{s\geq 0}$ be enlarged by its right-continuous completion; the usual conditions of completeness and right continuity as in Karatzas and Shreve [1991]. Let $U \subset \mathbb{R}^m$ be a compact control set, and let $\mathcal{U}$ denote the set of $\mathbb{F}$-progressively measurable maps into U as in Dellacherie and Meyer [1978]. The stochastic counter part of the ODE in (1) is the $\mathbb{R}^n$-valued SDE

$$\mathrm{d}\Xi_s = f(\Xi_s,u_s)\,\mathrm{d}s + \sigma(\Xi_s,u_s)\,\mathrm{d}W_s, \quad s \geq 0 \tag{7}$$

where $\Xi_0 = \xi$ is given, $f : \mathbb{R}^n \times U \to \mathbb{R}^n$ and $\sigma : \mathbb{R}^n \times U \to \mathbb{R}^{n\times n}$ are measurable maps, and $\mathbf{u} := (U_s)_{s\geq 0} \in \mathcal{U}$.

Under some mild reasonable assumptions it is known from Borkar et al. [2005] that there exists a unique strong solution to the SDE (7). We let $(\Xi^{t,\xi;\mathbf{u}}_{s})_{s\geq t}$ denote the unique strong solution of (7) starting from time t at the state ξ under control policy $\mathbf{u}$.

4.2 Stochastic Reach-Avoid Problem

Given an initial condition (t,ξ), we define the stochastic reachable set $R^{\mathrm{S}}_{f}(t,p;A,B)$ as the set of all initial conditions such that there exists an admissible control strategy $\mathbf{u} \in \mathcal{U}$ such that with probability more than p the state trajectory $\Xi^{t,\xi;\mathbf{u}}_{s}$ hits the set A before set B within the time horizon T.

Definition 10. (Stochastic Reachable Set).

$$\begin{aligned} R^{\mathrm{S}}_{f}(t,p;A,B) := \big\{\xi \in \mathbb{R}^n \,\big|\, \exists \mathbf{u} \in \mathcal{U} \text{ s.t.} \\ p < \mathbb{P}\big(\exists s \in [t,T],\ \Xi^{t,\xi,\mathbf{u}}_{s} \in A \\ \text{and } \forall r \in [t,s]\ \Xi^{t,\xi;\mathbf{u}}_{r} \notin B\big)\big\}. \end{aligned}$$

In Esfahani et al. [2011] it is shown that the stochastic reachable set R_f^{S} is equal to a superlevel set of a certain value function $V : [0,T] \times R^n \to \mathbb{R}$:

$$R_f^{\mathrm{S}}(t,p;A,B) = \{\xi \in \mathbb{R}^n | V(t,\xi) > p\}. \tag{8}$$

Furthermore this value function V is the solution of a PDE in the sense of discontinuous viscosity solutions as in Fleming and Soner [2006]. In Esfahani et al. [2011] this is formalised by the following definition and theorem.

Definition 11. (Differential Operator) Given $u \in U$, we denote by $\mathcal{L}_f^u$ the differential operator associated to the controlled diffusion (7) as

$$\mathcal{L}_f^u \Phi(t,\xi) := \partial_t \Phi(t,x) + \langle f(\xi,u), \partial_\xi \Phi(t,\xi) \rangle + \frac{1}{2} \operatorname{Tr}[\sigma \sigma^T(\xi,u) \partial_\xi^2 \Phi(t,\xi)],$$

where Φ is a real-valued function smooth on the interior of $\mathbb{S} := [0,T] \times \mathbb{R}^n$, with $\partial_t \Phi$ and $\partial_\xi \Phi$ denoting the partial derivatives with respect to t and ξ respectively, andc $\partial_\xi^2 \Phi$ denoting the Hessian matrix with respect to ξ.

Theorem 12. (Dynamic Programming Equation). Consider the system (7) and suppose that it is well behaved (i.e. conditions defined in Esfahani et al. [2011] hold). Then:

- the lower semicontinuous function of the value function V, defined as $V_*(t,\xi) := \liminf_{(t',\xi') \to (t,\xi)} V(t',\xi')$, is a viscosity supersolution of
$$-\sup_{u \in U} \mathcal{L}_f^u V_*(t,\xi) \geq 0 \qquad \text{on } [0,T) \times \overline{O}^c,$$
- the upper semicontinuous function of V, defined as $V^*(t,\xi) := \limsup_{(t',\xi') \to (t,\xi)} V(t',\xi')$, is a viscosity subsolution of
$$-\sup_{u \in U} \mathcal{L}_f^u V^*(t,\xi) \leq 0 \qquad \text{on } [0,T) \times \overline{O}^c,$$

both with boundary conditions

$$\begin{cases} V(t,\xi) = \mathbf{1}_A(\xi) & \forall (t,\xi) \in [0,T] \times \overline{O} \quad \text{(Lateral)}, \\ V(T,\xi) = \mathbf{1}_A(\xi) & \forall \xi \in \mathbb{R}^n \quad \text{(Terminal)}, \end{cases}$$

where $\overline{O}$ denotes the closure of $O := A \cup B$ and $\mathbf{1}_A(\cdot) : R^n \to \{0,1\}$ is the indicator function defined as

$$\mathbf{1}_A(\xi) := \begin{cases} 1 & \text{if } \xi \in A, \\ 0 & \text{if } \xi \notin A. \end{cases}$$

4.3 Robust Controller Derivation

In order to synthesise a more robust feedback control strategy for the simplified hybrid system introduced in Section 3.1, we define the new operator $\mathfrak{R}_f^{\mathrm{S}}$ that returns the saturated stochastic reachable set $R_f^{\mathrm{S}\,*}$ and a corresponding control function $\mathbf{u}^{\mathrm{S}}$ as

$$\mathbf{u}^{\mathrm{S}} : R_f(A,B) \to U, \quad \mathfrak{R}_f^{\mathrm{S}}(p;A,B) := \left(R_f^{\mathrm{S}\,*}(p;A,B), \mathbf{u}^{\mathrm{S}}(\cdot)\right)$$

where $p \in (0,1)$ is fixed. The saturated stochastic reachable set $R_f^{\mathrm{S}\,*}(p;A,B)$ is equal to the reachable set $R_f^{\mathrm{S}}(t,p;A,B)$ introduced in Definition 10 when the time horizon parameter T has been increased until saturation of the value function V in (8). Note that the domain of the stochastic controller $\mathbf{u}^{\mathrm{S}}$ is the deterministic reachable set R_f defined in Definition 2. Therefore, $\mathbf{u}^{\mathrm{S}}$ may be defined for states that do not lie in the saturated stochastic reachable set $R_f^{\mathrm{S}\,*}$.

In this section we shall discuss the details of the explicit computation of the feedback control policy $\mathbf{u}^{\mathrm{S}}(\xi)$ in a stochastic continuous time system. Applying this tool we can heuristically construct a new hybrid feedback control strategy by modifying Line 7 of Algorithm 2: instead of selecting the control function $\mathbf{u}_j$ from $\mathfrak{R}_f$ we set it to be the control function returned by the stochastic operator $\mathfrak{R}_f^{\mathrm{S}}$. The modified Line 7 in an enhanced version of Algorithm 2 then is

$$\left(R_{f_j}^{\mathrm{S}\,*}(p;A,B), \mathbf{u}_j(\cdot)\right) = \mathfrak{R}_{f_j}^{\mathrm{S}}\left(p; A_l \cap D_j, D_j^c\right)$$

where p is chosen a priori.

For the explicit derivation of the control function $\mathbf{u}^{\mathrm{S}}$, consider the dynamics of the simplified hybrid model from Section 3.1 modified by additional Brownian motion such that in every discrete mode we have a stochastic continuous time system as defined in Section 4.1. That is for an arbitrary mode $q \in Q$, we have the continuous state $\xi = [x\ y]^T$, the input $u = \psi$, the dynamics defined in (4), and the diffusion term $\sigma := \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix}$ where σ_x and σ_y are strictly positive. The evolution of the stochastic system is then described by the SDE

$$\begin{bmatrix} \mathrm{d}x_s \\ \mathrm{d}y_s \end{bmatrix} = \begin{bmatrix} v_q \cos(\psi) \\ v_q \sin(\psi) \end{bmatrix} \mathrm{d}s + \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \mathrm{d}W_s$$

and the corresponding differential operator defined in Definition 11 is

$$\mathcal{L}_{f_q}^{\psi} \Phi(t,x,y) = \partial_t \Phi + v_q \cos(\psi) \cdot \partial_x \Phi + v_q \sin(\psi) \cdot \partial_y \Phi + \frac{1}{2} \sigma_x^2 \cdot \partial_x^2 \Phi + \frac{1}{2} \sigma_y^2 \cdot \partial_y^2 \Phi. \tag{9}$$

For a state $\xi \in R_f(A,B)$ we define a time parameter t_R as

$$t_R(\xi) := \begin{cases} \sup\{t \in [0,T^*] \,|\, V(t,\xi) > p\}, & \text{if } V(0,\xi) > p, \\ 0 & \text{otherwise}, \end{cases}$$

where the time horizon parameter T^* is numerically selected such that the value function V is essentially saturated for small t. We heuristically obtain the controller $\mathbf{u}^{\mathrm{S}}(\xi)$ by selecting an input that maximises the differential operator applied to $V(t_R(\xi),\xi)$:

$$\mathbf{u}^{\mathrm{S}}(\xi) \in \operatorname*{arg\,max}_{\psi \in [0,2\pi]} \mathcal{L}_f^{\psi} V(t_R(x,y),x,y).$$

A resulting feedback control function is

$$\mathbf{u}^{\mathrm{S}}(\xi) := \begin{cases} \arctan\left(\dfrac{\partial_y \hat{V}}{\partial_x \hat{V}}\right) & \text{if } \partial_x \hat{V} > 0,\ \partial_y \hat{V} \geq 0, \\ \arctan\left(\dfrac{\partial_y \hat{V}}{\partial_x \hat{V}}\right) + \pi & \text{if } \partial_x \hat{V} < 0, \\ \arctan\left(\dfrac{\partial_y \hat{V}}{\partial_x \hat{V}}\right) + 2\pi & \text{if } \partial_x \hat{V} > 0,\ \partial_y \hat{V} < 0, \\ \pi/2 & \text{if } \partial_x \hat{V} = 0,\ \partial_y \hat{V} \geq 0, \\ 3\pi/2 & \text{if } \partial_x \hat{V} = 0,\ \partial_y \hat{V} < 0, \end{cases}$$

where we use the shorthand notation $\hat{V} := V(t_R(\xi),\xi)$. Roughly speaking, the control input $\mathbf{u}^{\mathrm{S}}(\xi)$ for a state $\xi \in R_f^{\mathrm{S}\,*}(p;A,B)$ is the direction of the steepest increase of the value function $V(t,\xi)$ with respect to ξ at the largest t for which $V(t,\xi)$ is grater than p.

Applying this control strategy to the simplified system in simulation, we observe that the car tends to drive further away from the avoid set and that the trajectories are smoother compared to the deterministic results; comparing Figures 2 (a) and 2 (b) these effects are visible.

The stochastic reachability approach offers also another advantage with respect to deterministic approach: explicite design parameters σ_x, σ_y, and p are introduced that have an intuitive effect.

5. REFERENCE TRACKING

In the preceding sections we have derived a controller for a simplified model. Note that the lookup control policy $\mathbf{u}^{\mathrm{H}}$ is numerically computed on a discretised grid of the continuous state space. Therefore, in practice it is stored for discrete points of the state space only. We denote the closest grid point to the position (x, y) as (x_g, y_g).

We need to control the full system introduced in Section 2.2 such that it follows the simplified behaviour. More specifically, the orientation ψ and velocity v should track their reference values fast enough relative to the change in the reference. The orientation reference denoted as ψ_r is the lookup value resulting from the reachability analysis: $\psi_r(x, y) := \mathbf{u}^{\mathrm{H}}(q(x_g, y_g), x_g, y_g)$. The reference for the velocity v_r is piecewise constant and depends on the evolution of the discrete state: $v_r(x, y) := v_{q(x_g, y_g)}$.

We want to use inputs $\delta \in [\pi/4, \pi/4]$ and $d \in [-1, 1]$ to control ψ and v around their reference values ψ_r and v_r. We define auxiliary variables $\eta := [\psi \ v]^T$ and $\lambda := [\delta \ d]^T$. Recall from (2) that the dynamics of ψ and v, $g(\eta, \lambda) := \left[\frac{\mathrm{d}\psi}{\mathrm{d}t} \ \frac{\mathrm{d}v}{\mathrm{d}t}\right]^T$, are

$$\begin{aligned}\frac{\mathrm{d}\psi}{\mathrm{d}t} &= \frac{1}{L}\delta v\\ \frac{\mathrm{d}v}{\mathrm{d}t} &= (C_{m1} - vC_{m2})d - C_d v^2 - C_r - C_\delta \delta^2 v^2 \qquad (10)\end{aligned}$$

For the task of controlling the physical system to track the desired references, different techniques are considered. A controller that minimises an appropriate Control-Lyapunov function described in Khalil [1992] works well in simulation but, due to delays in the real system, performs poorly in practice with strong visible chattering. For better performance on the real system a controller should process information that in some form measures how "far away" the controlled variables are from their reference values. A possible way of doing this is to apply gain-scheduled feedback control as in Khalil [1992].

5.1 Gain-Scheduled Linear Feedback Control

Gain scheduling is an approach to control non-linear systems, where the system is linearised around a set of operating points that are parametrised by a so-called scheduling variable. We refer to Khalil [1992] for details. We will now briefly derive a feedback controller using this technique.

For a given position (x, y) we linearise $g(\eta, \lambda)$ given in (10) around the desired operating point $\eta_e := [\psi_e \ v_e]^T = [\psi_r(x, y) \ v_r(x, y)]^T$. First we set $g(\eta, \lambda)|_{\eta=\eta_e, \lambda=\lambda_e} = 0$ to see that the following values of $\lambda_e := [\delta_e \ d_e]^T$ are appropriate inputs for the equilibrium: $\delta_e = 0$, $d_e(v_e) = \frac{C_d v_e^2 + C_r}{C_{m1} - v_e C_{m2}}$. Note that $d(v_e)$ is well defined for $v_e \le 3$. Regarding this fact, we can express the linearised system as

$$\frac{\mathrm{d}\hat{\eta}}{\mathrm{d}t} = A(v_e)\hat{\eta} + B(v_e)\hat{\lambda} \qquad (11)$$

where $\hat{\eta} := \eta - \eta_e$, $\hat{\lambda} := \lambda - \lambda_e$,

$$A(v_e) := \frac{\partial g}{\partial \eta}(\eta_e, \lambda_e) = \begin{bmatrix} 0 & 0 \\ 0 & -C_{m2}d_e(v_e) - 2C_d v_e \end{bmatrix},$$

$$B(v_e) := \frac{\partial g}{\partial \lambda}(\eta_e, \lambda_e) = \begin{bmatrix} \frac{1}{L}v_e & 0 \\ 0 & C_{m1} - v_e C_{m2} \end{bmatrix}.$$

Note that A and B depend on v_e but not on ψ_e. We therefore choose $v_e = v_r$ as the scheduling variable. It is constant within each discrete mode.

Since the controllability matrix $[B \ AB]$ has full rank 2 for $0 < v_e \le 3$, the linearised system (11) is controllable. We can design a linear feedback controller K_q for each mode $q \in Q$, $\hat{\lambda} = -K_q\hat{\eta}$. However, recall that there are input constraints. The controllability analysis above does not take these constraints into account; saturation of the inputs may make the controller fail.

We use a linear quadratic regulator (LQR) technique for the design of the gain matrices K_q. That is, we select the gain matrices that minimise the cost function $J = \int_0^\infty (\hat{\eta}^T P \hat{\eta} + \hat{\lambda}^T R \hat{\lambda})\,\mathrm{d}t$, where weight matrices P and R are design parameters. These explicit design parameters allow us to tune the controller such that the saturation of the inputs is minimised.

6. IMPLEMENTATION RESULTS

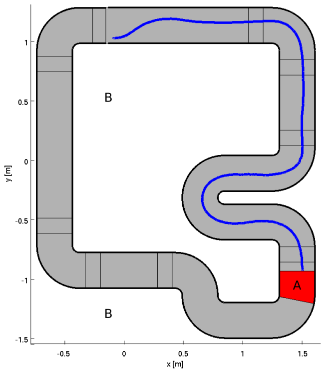
In the previous sections a feedback control strategy was derived by simplifying the full system (2), applying reachability analysis to a resulting lower dimension hybrid system (4), increasing robustness by considering noise and using gain-scheduled linear feedback control to make the actual system behave in the desired way. We have successfully implemented this control strategy on the physical system.

The physical system is the hardware setup of the project Optimal RC Autonomous (ORCA) Racer by Jones et al. [2010]. This configuration contains a vision system that provides an estimation of the full state of an RC-car on the track. The controller is implemented on a computer that communicates with the vision system and the car inputs. We assume that the hardware is fast enough such that we can consider the closed loop system as almost continuous time.

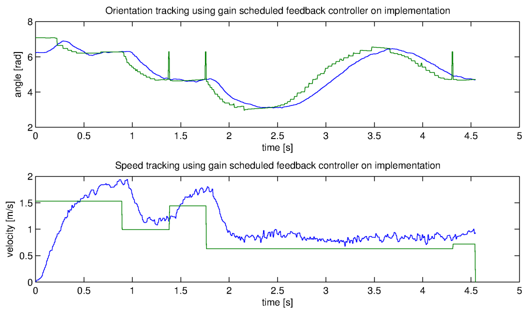
Figure 3 shows measurement data of the implementation on the real system. The applied parameters are: $v_1 = 0.99$, $v_2 = 1.53$, $v_3 = 0.99$, $v_4 = 1.44$, $v_5 = 0.72$, $v_6 = 0.63$, $v_7 = 1.44$, $v_8 = 0.99$, $v_9 = 1.53$, $\sigma_x = \sigma_y = 0.1$, $p = 0.9$, and the grid spacing is $\Delta x = \Delta y = 0.03$.

7. ACKNOWLEDGMENTS

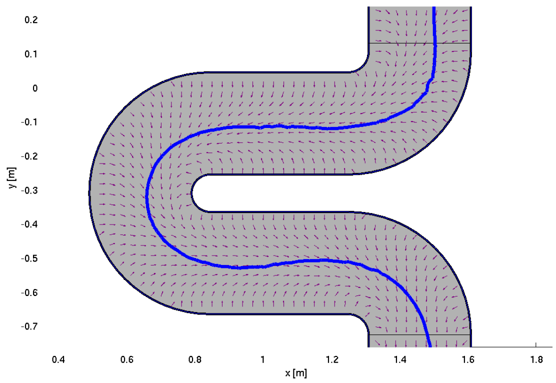
The authors would like to thank S. Summers, K. Margellos and A. Domahidi for helpful discussions and technical support.



(a) Trajectory of the position



(b) Orientation and speed tracking: state (blue), reference (green)



(c) Trajectory of the position and reference vector field $f(q(x,y),x,y,u(x,y))$ in sharp corner

Fig. 3. Measurement data of implementation on the real physical system applying a controller derived from stochastic reachability analysis combined with gain-scheduled reference tracking.

REFERENCES

V.S. Borkar et al. Controlled diffusion processes. *Probability surveys*, 2:213–244, 2005.

P. Cardaliaguet. A differential game with two players and one target. *SIAM journal on control and optimization*, 34:1441, 1996.

P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Differential games with state-constraints. In *ISDG2002, Vol. I, II (St. Petersburg)*, pages 179–182. St. Petersburg State Univ. Inst. Chem., St. Petersburg, 2002.

C. Dellacherie and P.A. Meyer. *Probabilities and potential*, volume 29. North-Holland Pub. Co., 1978.

P. Mohajerin Esfahani, D. Chatterjee, and J. Lygeros. On a problem of stochastic reach-avoid set characterization. In *Conference on Decision and Control (CDC)*, December 2011.

P. Mohajerin Esfahani, D. Chatterjee, and J. Lygeros. On stochastic reach-avoid problem and set characterization for diffusions. Technical report, February 2012. URL `http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=3772`.

I.J. Fialho and T.T. Georgiou. Worst case analysis of nonlinear systems. *Automatic Control, IEEE Transactions on*, 44(6):1180–1196, 1999.

W. H. Fleming and H. M. Soner. *Controlled Markov processes and viscosity solutions*, volume 25 of *Stochastic Modelling and Applied Probability*. Springer, New York, second edition, 2006.

Y. Gao, J. Lygeros, and M. Quincampoix. On the reachability problem for uncertain hybrid systems. *Automatic Control, IEEE Transactions on*, 52(9):1572–1586, 2007.

C.N. Jones, A. Domahidi, and M. Morari. Optimal RC autonomous racer, August 2010. URL `http://orcaracer.ethz.ch`. ETH Zurich.

I. Karatzas and S. E. Shreve. *Brownian motion and stochastic calculus*, volume 113 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1991.

H. K. Khalil. *Nonlinear systems*. Macmillan Publishing Company, New York, 1992.

J. Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40(6):917–927, 2004.

J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *AUTOMATICA-OXFORD-*, 35:349–370, 1999.

K. Margellos and J. Lygeros. Hamilton-jacobi formulation for reach-avoid differential games. *Automatic Control, IEEE Transactions on*, (99):1–1, 2009.

I. Mitchell and C. Tomlin. Level set methods for computation in hybrid systems. *Hybrid Systems: Computation and Control*, pages 310–323, 2000.

I. Mitchell, A. Bayen, and C. Tomlin. Validating a hamilton-jacobi approximation to hybrid system reachable sets. *Hybrid Systems: Computation and Control*, pages 418–432, 2001.

I.M. Mitchell and J.A. Templeton. A toolbox of hamilton-jacobi solvers for analysis of nondeterministic continuous and hybrid systems. *Hybrid Systems: Computation and Control*, pages 480–494, 2005.

S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

J.A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Number 3. Cambridge Univ Pr, 1999.

C. Tomlin, J. Lygeros, and S. Sastry. Synthesizing controllers for nonlinear hybrid systems. *Hybrid Systems: Computation and Control*, pages 360–373, 1998.

L. Wunderli. MPC based trajectory tracking for 1:43 scale race cars. Semester project, April 2011. ETH Zurich.