Reachability-based Synthesis of Feedback Policies for Motion Planning Under Bounded Disturbances

Jerry Ding, Eugene Li, Haomiao Huang, and Claire J. Tomlin

Abstract— **The task of planning and controlling robot motion in practical applications is often complicated by the effects of model uncertainties and environment disturbances. We present in this paper a systematic approach for generating robust motion control strategies to satisfy high level specifications of safety, target attainability, and invariance, under unknown but bounded time-varying disturbances. The motion planning task is decomposed into the two sub-problems of finite horizon reach while avoid and infinite horizon invariance. The set of states for which each of the sub-problems is robustly feasible is computed via iterative reachability calculations under a differential game framework. We will discuss how the results of this computation can be used to inform selections of control inputs based upon state measurements at run-time and provide an algorithm for implementing the corresponding feedback control policies. Finally, we demonstrate an experimental application of this method to the control of an autonomous helicopter in tracking a moving ground vehicle.**

I. Introduction

An important consideration in ensuring safe and reliable deployment of mobile robots and autonomous vehicles is whether the desired objectives can be met despite uncertainties in the operating condition. In this paper, we consider the problem of constructing feedback policies to satisfy motion planning task specifications of the following form: from a set of initial robot configurations, reach a set of goal configurations, and then remain there indefinitely, subject to system dynamics, safety constraints, and unknown but bounded, time-varying disturbances. The controller is assumed to operate at a high level control layer, where the control commands may be symbolic, in the form of a finite set of maneuvers (e.g. move forward, turn left, turn right), parameterized by a continuous variable (e.g. velocity, acceleration, turn rate). Under the setting of a sampled data system, the selection of controls is based upon state measurements at discrete sampling instants and the inputs are held constant on sampling intervals.

It is well-known that motion planning under state and input constraints can be a difficult problem [1], [2], even in a deterministic setting. Global search methods include visibility graphs [3], [4] and cell decomposition [5], [6]. Real-time planning methods such as probabilistic roadmaps [7], [8] and artificial potential fields [9], [10] reduce the computational complexities through probabilistic sampling and reactive local planning. These methods are shown experimentally to perform well under small deviations from the assumed nominal conditions. However, under the effects of model inaccuracy, sensor and actuator noise, or external disturbances, deterministic guarantees about the performance of the previously mentioned methods become difficult to formulate and prove.

Motion planning under uncertainty is a significantly more complicated problem [11], due to the fact that under an unknown disturbance, the actual motion undertaken by the robot could vary from one execution to another, even for the same starting configuration and control input sequence. Thus, instead of considering the feasibility of a single deterministic trajectory, one needs to consider the feasibility of a family of possible trajectories consequent on the chosen motion strategy. This problem is addressed in [12]–[15] using *preimage backchaining*, in the context of fine-motion planning. In this approach, geometric arguments are used to construct preimages of the goal set, namely the set of configurations that are guaranteed to reach the goal set under some sequence of piecewise constant velocity vectors, regardless of control and sensing errors. However, with more complicated nominal dynamics, the computation of the preimage becomes involved. For such situations, a method is proposed in [16] for constructing local feedback controllers whose domain of attraction is approximated by Lyapunov functions. Motion planning is then performed by backchaining of the local controllers. More recently, closed-loop rapidly-exploring random trees (CL-RRT) [17] have been proposed to provide feasibility guarantees under bounded disturbances, using conservative estimates of the trajectory tracking error to tighten constraints on the feasible planning space and input set.

The use of a discrete set of controllers to perform motion planning is motivated by reasons of computation and implementation. In the work of [18] and [19], the authors propose methods for constructing a library of flight maneuvers for an autonomous helicopter, with compatible initial and terminal conditions, so as to reduce a difficult continuous planning problem to a selection of maneuver sequence and switching times. In the work of [20] and [21], the authors consider the problem of synthesizing motion plans that realize high level objectives specified in linear temporal logic. Discrete ab-

This work is supported by the "MURI - Frameworks and Tools for High Confidence Design of Adaptive, Distributed Embedded Control Systems" project administered by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-06-1-0312.

J. Ding and C. J. Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA {jding,tomlin}@EECS.Berkeley.Edu

E. Li is with salesforce.com, San Francisco, CA 94105, USA eugene.li@salesforce.com. Research performed while at Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA

H. Huang is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA haomiao@stanford.edu

stractions are constructed of the continuous robot motion by partitioning the state space and appropriately designing continuous controllers for each partition. The above approaches assume that the continuous dynamics are not perturbed by time-varying disturbances, which complicates the task of ensuring the correctness of the discrete abstractions.

We propose in this work a systematic approach for computing the robust feasible sets of the motion planning problem via iterative reachability calculations, and then using the results of the computation to synthesize feedback policies for closed-loop control of robot motion. Our approach is based upon the game theoretic framework for hybrid controller design described in [22] and [23]. This framework has been applied successfully to problems of flight envelope protection, automated highway platooning, and provably safe aerobatic maneuvers [24]–[26], as a means for open-loop system design and verification. The focus of this paper is instead on automatic synthesis of closed-loop policies.

The advantages of the proposed method are several-fold. First, the type of continuous dynamics that can be handled is quite general, although restricted to low dimensions (3-4 continuous states) due to computational concerns. Second, performance guarantees are ensured under worst case disturbance behavior, using techniques from differential games. Third, in comparison with open-loop methods for motion planning, feedback policies mitigate the effects of disturbance by allowing the system to respond to changes in the system state at run-time. Finally, the control solution for reaching the goal region can be viewed as a *robust minimum time to reach* controller.

Part of the methodology described in this paper is presented in the general context of switched nonlinear systems in [27]. Here we specialize the model and results for application to motion planning problems. The controller synthesis procedure is also extended to allow the robot to remain indefinitely in a goal region, as opposed to merely reaching it in finite time. Finally, rather than a numerical simulation, experimental results will be presented of the application of the proposed approach to the control of a quadrotor helicopter under dynamic uncertainty.

II. Problem Formulation

The configuration of the robot at any given time is assumed to be summarized by a state vector $x \in \mathbb{R}^n$, where n is the dimension of the state space. To effect changes to this configuration, it is further assumed that we have access to a finite set of discrete modes $Q = \{q_1, q_2, ..., q_m\}$ describing the set of high level commands (e.g. move forward, turn left, turn right). Within each mode q_i, one can choose a scalar valued control input u_i, which can be a parametrization of the individual maneuver (e.g. velocity, acceleration, turn rate). For computational purposes and practical implementation considerations, we discretize the input range of u_i into a finite number of levels, namely $U_i = \left\{u_i^1, u_i^2, ..., u_i^{L_i}\right\} \subset \mathbb{R}$, where L_i is the number of quantization levels. Finally, a vector valued time-varying function d_i is used to capture the effect of disturbance on the system dynamics in mode q_i. It is assumed that d_i takes on a bounded range $D_i \subset \mathbb{R}^{M_i}$.

Starting from some initial time $t = 0$ and initial configuration x_0, we model the continuous motion of the robot, as per Newtonian dynamical models, by the ordinary differential equation

$$\dot{x}(t) = f_{q(t)}(x(t), u(t), d(t)), \;\; x(0) = x_0 \tag{1}$$

where $q(t) \in Q$, $u(t) \in U_{q(t)}$, and $d(t) \in D_{q(t)}$, which can be used to model the effects of model inaccuracy, actuator noise, environment disturbances, or the unknown inputs of a moving object, as long as the bounds on d are known beforehand or can be conservatively estimated.

It is assumed that measurements of the system state x are received at sampling instants kT, $k = 1, 2, ...$, where T is the sampling period, and that the inputs $(q(t), u(t))$ are held constant on sampling intervals. Specifically, based upon a state measurement $x(kT)$, we select $q_k \in Q$ and $u_k \in U_{q_k}$ and apply the input $q(t) \equiv q_k$ and $u(t) \equiv u_k$, for $t \in [kT, (k+1)T)$. On the other hand, d is not restricted to be piecewise constant, as long as $d(t) \in D_{q_k}$ is satisfied for $t \in [kT, (k+1)T)$.

We denote by $W \subset \mathbb{R}^n$ the desired goal set, and by $A \subset \mathbb{R}^n$ the avoid set (or unsafe set), describing for example physical obstacles or unsafe operating regions. The problem we would like to solve is then, roughly speaking, to construct a control policy as a rule of selecting (q_k, u_k) based upon measurement $x(kT)$, so as to drive the system state x into the set W within a finite number of steps N, while avoiding the set A, and then remain in W for all time steps $k > N$, regardless of the possible realizations of disturbance d at run-time. This can be naturally decomposed into the two sub-problems of reaching W in finite time and then staying inside W over infinite time. To state this more precisely, a brief discussion on the permissible control and disturbance policies is needed.

For some finite horizon $[kT, NT)$, define a control policy which only depends on the state measurements at sampling instants by the sequence $\pi_{k\to N} = (\mu_k, \mu_{k+1}, ...\mu_{N-1})$ of state feedback maps $\mu_j : \mathbb{R}^n \to V$ from the state space to the control input space $V = \bigcup_{q_i \in Q} q_i \times U_i$. We denote the set of such admissible control policies by $P_{k\to N}$.

Under a differential game setting, we assume that the disturbance has full knowledge of the control input selected on any sampling interval. Then, a permissible disturbance strategy over the time horizon $[kT, NT)$ is defined by the sequence $\gamma_{k\to N} = (\nu_k, \nu_{k+1}, ...\nu_{N-1})$ of maps $\nu_j : V \to \bigcup_{q_i \in Q} \mathcal{D}_i$, satisfying $\nu_j(q_i, u_i) \in \mathcal{D}_i$, where $\mathcal{D}_i$ is the set of possible realizations of d_i on sampling interval $[kT, (k+1)T)$. We denote the set of such permissible disturbance strategies by $D_{k\to N}$.

It should be noted that this information structure allows $d(t)$ to be chosen rationally based upon both the state $x(t)$ and the input $u(t)$ [28], which could be the case if d is used to model the action of another robot or autonomous vehicle with competing objectives. However, this may be a conservative assumption for random but bounded noise.

With these definitions, the desired task specification can be met by solving the following two sub-problems in sequence.

Problem 1: (Infinite Horizon Invariance) Given system (1) and set W: 1) compute the set of states $E_{Inv} \subset W$ such that for $x_0 \in E_{Inv}$, there exists an admissible control policy $\pi_{0\to\infty} \in P_{0\to\infty}$ so that for any disturbance strategy $\gamma_{0\to\infty} \in D_{0\to\infty}$, the closed-loop state trajectory $x_{cl}(\cdot)$ satisfies $x_{cl}(t) \in E_{Inv}$ for all $t \in [0, \infty)$; 2) synthesize a time-invariant state feedback law $F_{Inv}(x)$ such that for any initial condition $x_0 \in E_{Inv}$, the closed-loop trajectory satisfies the above conditions.

Using the solution of Problem 1, we choose some set $R \subset E_{Inv}$ and then solve the problem below.

Problem 2: (Finite Horizon Reach-Avoid) Given system (1) and sets R, A: 1) compute the set of states $E_{RA} \subset \mathbb{R}^n$ such that for $x_0 \in E_{RA}$, there exists an admissible control policy $\pi_{0\to N} \in P_{0\to N}$ so that for any disturbance strategy $\gamma_{0\to N} \in D_{0\to N}$, the closed-loop state trajectory $x_{cl}(\cdot)$ satisfies $x_{cl}(kT) \in R$ for some $k \in \{0, 1, \ldots, N\}$, and $x_{cl}(t) \notin A$ for all $t \in [0, kT]$; 2) synthesize a time-varying state feedback law $F_{RA}(x, k), k = 0, 1, \ldots, N-1$ such that for any initial condition $x_0 \in E_{RA}$, the closed-loop trajectory satisfies the above conditions.

In the statements of the two sub-problems, $x_{cl}(\cdot)$ is the closed-loop trajectory determined by the initial condition x_0, control policy π, and disturbance strategy γ, under dynamics (1). Due to the unpredictable nature of the disturbance, we choose to synthesize a *feedback policy* π rather than a sequence of inputs (q_k, u_k), $k = 0, 1, ...$ It is possible that a different sequence of inputs is obtained for each execution, using the same policy π, due to different realizations of the disturbance.

III. Reachable Set Computation and Controller Synthesis

In this section, an algorithm will be given to compute, in an automated fashion, the set of initial conditions E_{RA} for which the finite horizon reach-avoid problem is feasible, using Hamilton-Jacobi reachability analysis [28]. It will be shown how this algorithm can be extended to compute the set E_{Inv} solving the invariance problem. From the results of these computations, we proceed to synthesize F_{RA} and F_{Inv} as set-valued state feedback laws and discuss how they can be implemented in closed-loop control to accomplish the overall motion planning objectives.

For the discussion in this section, we denote by $C_{k\to N}$ the set of initial conditions for which the reach-avoid problem is feasible over the time horizon $[kT, NT)$. The offline computation will focus on generating representations of the sets $C_{k\to N}$, while the online implementation uses the information provided by $C_{k\to N}$ to make input selections based upon state measurements.

A. Finite Horizon Reach-Avoid Set Computation

In order to formulate an iterative algorithm for computing $C_{k\to N}$, we first consider the simpler problem of driving the state x of the robot into a set $R \subset \mathbb{R}^n$ at the end of a sampling interval, while avoiding a set $A \subset \mathbb{R}^n$, namely $x(T) \in R$ and $x(t) \notin A$, $\forall t \in [0, T]$. The one-step feedback policy is described by a selection of $(q_i, u_i) \in V$ for each feasible initial condition $x_0 \in \mathbb{R}^n$, while the one-step disturbance strategy corresponds to a selection of $d_i(\cdot) \in \mathcal{D}_i$ on $[0, T]$. We denote the set of feasible initial conditions for this problem by $\mathcal{RA}(R, A, T)$.

For a choice of $(q_i, u_i) \in V$, we define $Pre_d^{q_i,u_i}(A, T)$ as the set of initial conditions x_0 for which there exists some choice of disturbance $d_i(\cdot) \in \mathcal{D}_i$ on $[0, T]$, such that the state trajectory satisfies $x(t) \in A$ for some $t \in [0, T]$, where $x(\cdot)$ is the solution of the ODE $\dot{x} = f_q(x, u, d_i)$, $x(0) = x_0$, $(q(t), u(t)) \equiv (q_i, u_i)$ on the sampling interval $[0, T]$. Suppose A is the set of obstacles or unsafe operating regions, this describes the set of initial conditions that can be rendered unsafe within one sampling interval under the input (q_i, u_i).

Under suitable technical conditions given in [28], this set can be computed using a Hamilton-Jacobi-Isaacs (HJI) partial differential equation (PDE). To perform this computation, we use level set representation of sets. Specifically, suppose a set $G \subset \mathbb{R}^n$ is the set of all states x such that $\phi_G(x) \leq 0$ for some function $\phi_G : \mathbb{R}^n \to \mathbb{R}$, then ϕ_G is a level set representation of G. For example, a disc centered on the origin with radius r can be represented by $\phi_{D(0,r)}(x_1, x_2) = \sqrt{x_1^2 + x_2^2} - r$.

Let $\phi : \mathbb{R}^n \times [-T, 0] \to \mathbb{R}$ be the viscosity solution of the terminal value HJI PDE

$$\frac{\partial \phi}{\partial t} + \min\left[0, H\left(x, \frac{\partial \phi}{\partial x}\right)\right] = 0, \ \phi(x, 0) = \phi_A(x) \quad (2)$$

where the optimal Hamiltonian is given by

$$H(x, p) = \min_{d_i \in D_i} p^T f_{q_i}(x, u_i, d_i)$$

Then by a special case of the argument presented in [28], $Pre_d^{q_i,u_i}(A, T) = \{x \in \mathbb{R}^n, \phi(x, -T) \leq 0\}$. A numerical toolbox as described in [29] can be used to compute the solution to equation (2).

Next, we define for a choice of $(q_i, u_i) \in V$ the set $\overline{Pre}_u^{q_i,u_i}(R, T)$ of initial conditions x_0 for which regardless of the realization of the disturbance $d_i(\cdot) \in \mathcal{D}_i$ on $[0, T]$, the state trajectory satisfies $x(T) \in R$, where $x(\cdot)$ is the solution of the ODE $\dot{x} = f_q(x, u, d_i)$, $x(0) = x_0$, $(q(t), u(t)) \equiv (q_i, u_i)$ on the sampling interval $[0, T]$. Suppose R is the target set, then this describes the set of initial conditions that can reach the target set after one sampling interval under the input (q_i, u_i). This set can be computed by a slight modification of equation (2).

Putting these definitions together, we find that the set of initial condtions that can be driven into a set $R \subset \mathbb{R}^n$ at the end of a sampling interval, while avoiding a set $A \subset \mathbb{R}^n$ using a choice of $(q_i, u_i) \in V$ has the concise representation

$$\mathcal{RA}^{q_i,u_i}(R, A, T) = \overline{Pre}_u^{q_i,u_i}(R, T) \backslash Pre_d^{q_i,u_i}(A, T).$$

We note that for sets G_1, G_2 with level set representations ϕ_{G_1} and ϕ_{G_2}, the representation for $\phi_{G_1} \backslash \phi_{G_2}$ is computed simply by taking pointwise maximization $\max\{\phi_{G_1}, -\phi_{G_2}\}$.

It can be directly inferred that the set of feasible initial conditions for the reach-avoid problem over $[0,T]$ is

$$\mathcal{RA}(R,A,T)=\bigcup_{(q_i,u_i)\in V}\mathcal{RA}^{q_i,u_i}(R,A,T).$$

Since the input set V is finite, the above union is taken over a finite number of sets represented by level set functions. For sets G_1,G_2 with level set representations ϕ_{G_1} and ϕ_{G_2}, the representation for $\phi_{G_1}\cup\phi_{G_2}$ is computed by taking pointwise minimization $\min\{\phi_{G_1},\phi_{G_2}\}$.

The operator $\mathcal{RA}$ gives us a recipe for computing the sets $C_{k\to N}$. Namely, $\mathcal{RA}(R,A,T)$ gives the set $C_{N-1\to N}$, which could then be used to initialize the computation of $C_{N-2\to N}$. Stated more precisely, consider Algorithm III.1

Algorithm III.1 Computation of Exact Finite Horizon Reach-Avoid Set

Require: $R,A\subset\mathbb{R}^n$
1: $S_0\Leftarrow R$
2: **for** $j=0$ to $N-k-1$ **do**
3: $\quad S_{j+1}\Leftarrow\mathcal{RA}(S_j,A,T)\cup S_j$
4: **end for**
5: **return** S_{N-k}

By a special case of the argument presented in [27], we have the following result.

Proposition 3.1: The output S_{N-k} of Algorithm III.1 satisfies $S_{N-k}=C_{k\to N}$, $k=0,1,...,N-1$. In particular, S_N is the N step finite horizon reach-avoid set $C_{0\to N}$.

As a result of this proposition, it is clear that the set E_{RA} as required by Problem 2 is given by $E_{RA}=S_N$.

B. Infinite Horizon Invariant Set Computation

Now suppose the goal is instead to make the robot stay in W over either a finite or infinite time horizon, then we can take $R=W$ and $A=W^C$ and perform the reach-avoid set computation as in Algorithm III.1, except replacing the statement $S_{j+1}\Leftarrow\mathcal{RA}(S_j,A,T)\cup S_j$ by $S_{j+1}\Leftarrow\mathcal{RA}(S_j,A,T)$. Then by a very slight modification of the proof given in [27], S_k is the set of initial conditions for which there exists a control policy, such that regardless of the disturbance strategy, the closed-loop state trajectory $x_{cl}(\cdot)$ of (1) satisfies $x_{cl}(kT)\in W$ and $x_{cl}(t)\notin W^C$, for all $t\in[0,kT]$, or equivalently, $x_{cl}(t)\in W$, for all $t\in[0,kT]$. In other words, S_k is an k step invariant set. This is summarized in Algorithm III.2.

Algorithm III.2 Computation of Finite Horizon Invariant Set

Require: $W\subset\mathbb{R}^n$
1: $\tilde{S}_0\Leftarrow W$
2: **for** $j=0$ to $k-1$ **do**
3: $\quad \tilde{S}_{j+1}\Leftarrow\mathcal{RA}(\tilde{S}_j,W^C,T)$
4: **end for**
5: **return** $\tilde{S}_k$

Clearly, the existence of an invariant set over infinite horizon depends on the convergence of Algorithm III.2. For our application, we assume that the algorithm converges numerically to a fixed point at some iteration k_0, namely $\tilde{S}_{k_0+1}=\mathcal{RA}(\tilde{S}_{k_0},W^C,T)=\tilde{S}_{k_0}$. Then by straightforward induction, $\tilde{S}_k=\tilde{S}_{k_0}$, $\forall k\geq k_0$. This implies that the invariant set E_{Inv} as required by Problem 1 is given by $E_{Inv}=\lim_{k\to\infty}\tilde{S}_k=\tilde{S}_{k_0}$.

C. Reach-Avoid and Invariance Controller Synthesis

Given a finite horizon N, automated computations using Algorithm III.1 and III.2 can be performed offline, yielding the sets $\mathcal{RA}^{q_i,u_i}(S_j,A,T)$ for $(q_i,u_i)\in V$, and $j=0,1,...,N-1$, and the sets $\mathcal{RA}^{q_i,u_i}(E_{Inv},W^C,T)$ for $(q_i,u_i)\in V$. There are $N_R=(N+1)(\sum_{i=0}^{m}L_i)$ such sets, where m is the number of modes and L_i is the number of quantization levels in q_i.

Now consider a feasible initial condition $x(0)\in S_N$. First, we find the smallest index $N_0\leq N$ such that $x(0)\in S_{N_0}$. Suppose $N_0=N$, then by the property of the set S_N, $\exists(q_i,u_i)\in V$, such that $x(0)\in\mathcal{RA}^{q_i,u_i}(S_{N-1},A,T)$. Hence, by choosing $(q(0),u(0))=(q_i,u_i)$ as our control input in step $k=0$, we ensure that regardless of the choice of disturbance input $d(\cdot)\in\mathcal{D}_{q(0)}$, the state trajectory satisfies $x(T)\in S_{N-1}=C_{1\to N}$ and $x(t)\notin A$, $\forall t\in[0,T]$.

At the next time step, we obtain a state measurement $x(T)$, find the smallest index $N_1\leq N-1$ such that $x(T)\in S_{N_1}$, and select controls $(q(T),u(T))$ using a similar procedure. As can be seen, the choice of $(q(T),u(T))$ depends upon the realization of the disturbance $d(\cdot)\in\mathcal{D}_{q(0)}$. Proceeding in this manner, we obtain $(q(kT),u(kT))$, $k=0,1,...,N-1$ as a function of the state measurement at each time step and ensures that $x(kT)\in R$ for some $k\in\{0,1,\ldots,N\}$ and $x(t)\notin A$ for all $t\in[0,kT]$, regardless of the choice of disturbance strategy $\gamma_{0\to N}\in D_{0\to N}$.

Now by assumption, the set R is a subset of the invariant set E_{Inv}. Then by the properties of E_{Inv}, $x(kT)\in R$ implies that $\exists(q_i,u_i)\in V$, such that $x(kT)\in\mathcal{RA}^{q_i,u_i}(E_{Inv},W^C,T)$. Hence, by choosing the control input $(q(kT),u(kT))=(q_i,u_i)$ in time step k, we ensure that regardless of the choice of disturbance input $d(\cdot)\in\mathcal{D}_{q(kT)}$, the state trajectory satisfies $x((k+1)T)\in E_{Inv}$ and $x(t)\in W$, $\forall t\in[kT,(k+1)T]$. This procedure can be repeated over all subsequent time steps to keep the trajectory inside W, thus accomplishing the overall objectives.

To state this more formally, define $I=\{0,1,\ldots,N-1\}$ and let 2^V be the power set of V. Then the explicit finite horizon reach-avoid control policy is given by the time-varying, set-valued state feedback law $F_{RA}:\mathbb{R}^n\times I\to 2^V$

$$F_{RA}(x,k)=\{(q_i,u_i),x\in\mathcal{RA}^{q_i,u_i}(S_{N_k-1},A,T)\}\qquad(3)$$

where $N_k\leq N-k$ is the smallest index such that $x\in S_{N_k}$. On the other hand, the infinite horizon invariance control policy is given by the time invariant, set-valued state feedback law $F_{Inv}:\mathbb{R}^n\to 2^V$

$$F_{Inv}(x)=\left\{(q_i,u_i),x\in\mathcal{RA}^{q_i,u_i}(E_{Inv},W^C,T)\right\}\qquad(4)$$

In pseudo-code, these feedback policies can be implemented as in Algorithm III.3.

Algorithm III.3 Online Implementation of Reach-Avoid and Invariance Control Policies

Require: $x(0) \in S_N$

InvMode $\Leftarrow 0$
for $k = 0, 1, ...$ **do**
  $F_k \Leftarrow \emptyset$
  Measure state $x(kT)$
  if $x(kT) \in R$ or InvMode $\equiv 1$ **then**
    InvMode $\Leftarrow 1$
    for all $(q_i, u_i) \in V$ **do**
      if $x(kT) \in \mathcal{RA}^{q_i,u_i}(E_{Inv}, W^C, T)$ **then**
        Add (q_i, u_i) to F_k
      end if
    end for
  else
    Find smallest $N_k \leq N - k$ such that $x(kT) \in S_{N_k}$
    for all $(q_i, u_i) \in V$ **do**
      if $x(kT) \in \mathcal{RA}^{q_i,u_i}(S_{N_k-1}, A, T)$ **then**
        Add (q_i, u_i) to F_k
      end if
    end for
  end if
  Apply $(q(kT), u(kT)) \in F_k$
end for

Remark: For any initial condition $x_0 \in S_k \backslash S_{k-1}$, $k = 1, ..., N$, F_{RA} is guaranteed to control x_0 into R within k time steps, regardless of the disturbance strategy. Furthermore, given that $x_0 \notin S_{k-1} = C_{0\to k-1}$, there does not exist an admissible control law that can accomplish the same task within $k-1$ time steps. In this sense, F_{RA} can be interpreted as a robust minimum time to reach control law.

IV. Experimental Results

To illustrate this methodology, we will consider the problem of controlling a quadrotor helicopter first to some position on top of a stationary ground vehicle, while satisfying constraints on the velocity, and then hovering over the vehicle as it starts moving. The internal disturbances appear in the form of model uncertainties and actuator noise, while the external disturbance is the movement of the ground vehicle, which is not planned ahead of time. The control policies are implemented on the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC) (see Figure 1), an unmanned aerial vehicle platform consisting of six quadrotor helicopters each equipped with onboard computation, sensing, and control capabilities [30].

Under a previously designed inner control loop, the position-velocity dynamics in the x and y directions can be assumed to be decoupled, with pitch and roll angles as the respective control inputs. Then from the point of view of the high level controller, the dynamics of the vehicle under pitch and roll commands can be approximated as

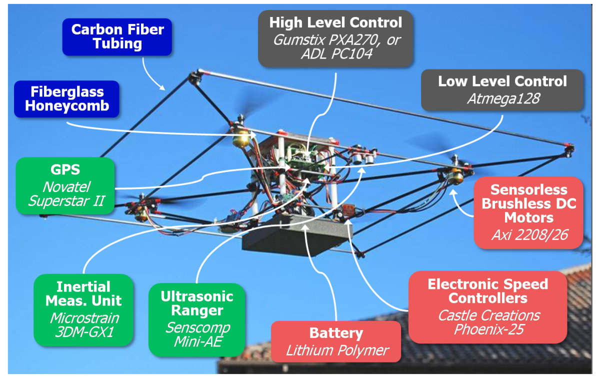


Fig. 1. **An overview of components on STARMAC.**

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{y_1} \\ \dot{y_2} \end{bmatrix} = \begin{bmatrix} x_2 + d_1 \\ g\sin(-\phi) + d_2 \\ y_2 + d_3 \\ g\sin(\psi) + d_4 \end{bmatrix} \quad (5)$$

where x_1, x_2 and y_1, y_2 are the position and velocity of the quadrotor in the x-axis and y-axis, respectively, with respect to the position and velocity of the ground vehicle, g is the gravitational constant, ϕ is the pitch command, ψ is the roll command, and $d = (d_1, d_2, d_3, d_4)$ are the disturbance terms. For d_1 and d_3, the disturbance bound is chosen to be $\pm 0.1 m/s$, corresponding to $\pm 10\%$ of the maximum allowed vehicle velocity. For d_2 and d_4, a slightly larger disturbance bound of $\pm 0.5 m/s^2$ is selected, corresponding to about $\pm 30\%$ of the maximum allowed acceleration, in order to capture the unknown acceleration of the ground vehicle.

For our experiments, the hover region W is chosen to be a squared shaped region centered on the ground vehicle, with some tolerance on the relative velocity, while the avoid region A is the set of all relative velocities violating a velocity constraint. The precise problem parameters are summarized below:

- Hover Region (W): $|x_1|, |y_1| \leq 0.3$ m for position, and $|x_2|, |y_2| \leq 0.5$ m/s for velocity
- Avoid Region (A): $|x_2|, |y_2| > 1$ m/s for velocity
- Time Step (T): 0.1 seconds
- Time Horizon for Reaching W (N): 25 time steps
- Range of Attitude Commands (ϕ, ψ): -10, -7.5, -5, -2.5, 0, 2.5, 5, 7.5, 10 degrees

The hover region and avoid region are plotted in the position-velocity plane in Figure 2. Given that the dynamics in the x and y directions are decoupled, we can perform the reachability analysis in two dimensions to save computation time and use the result to synthesize control policies for the pitch and roll angles, up to a sign change in the commanded angle. For the rest of this section, the results of the reachable set computation will be shown in the position-velocity plane, with the understanding that the same sets apply to movement in both the x and y directions.

Using the computation procedure described in Section III-B, we compute an invariant subset E_{Inv} of the hover region.

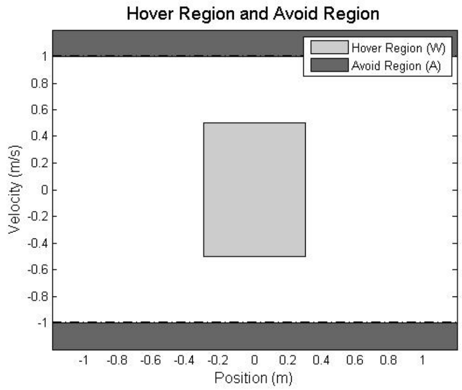


Fig. 2. **Plot of Hover Region and Avoid Region in the Position-Velocity Plane.**

The result is shown in Figure 3. Intuitions suggest that if the vehicle is at a large positive displacement, with high positive velocity, it is likely to exit the hover region. Indeed these states are excluded from E_{Inv}, along with states corresponding to large negative displacement and velocity. To accomplish the reach and hover objectives, we choose a subset $R = \{(x_1, x_2, y_1, y_2) : |x_1|, |y_1| \leq 0.2\text{m}, |x_2|, |y_2| \leq 0.2\text{m/s}\} \subset E_{Inv}$ to be the target zone for the finite horizon reach-avoid algorithm.

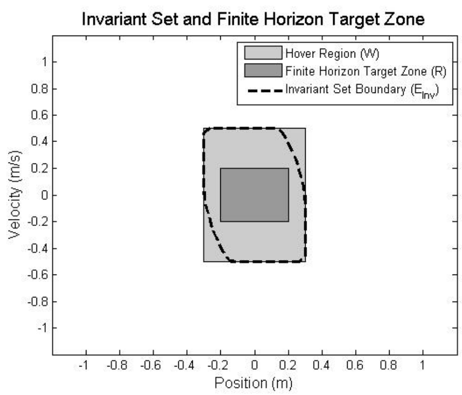
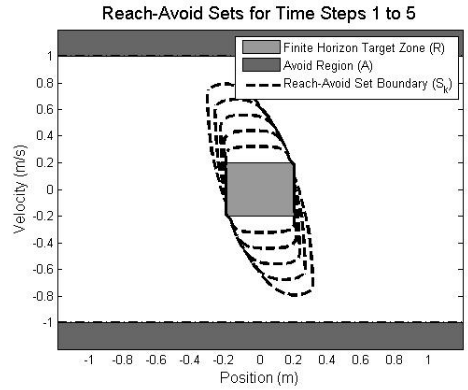


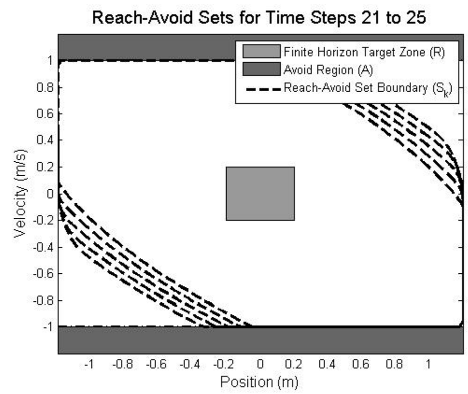
Fig. 3. **Plot of Invariant Set and Finite Horizon Target Zone in the Position-Velocity Plane.**

Under Algorithm III.1, we initialize the computation of the finite horizon reach-avoid set using R and progress backwards in time over sampling intervals. The resulting reach-avoid sets are shown in Figure 4 (a) and (b), for time steps 1 through 5 and time steps 21 through 25, respectively. As intuitions suggest, at a positive displacement, one needs negative velocity to arrive at the origin, and vice versa for negative displacement. Thus, the growth of the sets are angled towards the second and fourth quadrants of the position-velocity state space in the first few time steps. Over longer time horizons, the velocity constraints come into effect and limits the growth of the reach-avoid sets.

The control policy satisfying the desired objectives is implemented at a high-level control layer onboard the quadrotor vehicle according to Algorithm III.3. For the actual experi-



(a)



(b)

Fig. 4. **(a) Reach-avoid Sets at Time Steps 1 through 5; (b) Reach-avoid Sets at Time Steps 21 through 25.**

ments, the reachable sets are pre-loaded onto the on-board memory. During run-time, the vehicle consults these sets when selecting the appropriate pitch and roll commands based upon position and velocity measurements. Taking into account that there may be sources of noise and disturbance not accounted for in the disturbance bound estimates, for example occasional acceleration bursts by the ground vehicle, we add an extra condition that if the invariant set is ever violated during run-time, the controller is switched back into finite horizon reach-avoid mode.

In Figures 5 and 6 the trajectory flown by STARMAC is shown for an experimental trial where the quadrotor is initialized at approximately $(x_1, x_2, y_1, y_2) = (1, 0, 1.1, 0)\text{m}$, relative to a stationary ground vehicle placed at the origin. The squares in these plots marks the initial condition of the quadrotor, while the circles marks the relative states at 1.8 seconds into the experiment, when the finite horizon target zone is first attained. After the target zone is attained, the quadrotor hovers over the stationary ground vehicle for about 35 seconds, upon which time the experiment is terminated. The trajectory plots show that the vehicle indeed achieves the desired objectives of reaching the target zone within 2.5 seconds while satisfying safety limits on the allowed velocity, despite perturbations by time varying disturbances. Furthermore, except for a few brief violations, the quadrotor remained within the desired hover region over the course of 35 seconds. The source for these violations can be attributed to time-lag in the vehicle response, noise in the velocity estimation, and numerical errors in the computation of the reachable sets, which are not formally taken into account. However, despite these non-ideal operating conditions, the quadrotor is shown to quickly recover itself inside the hover region using the finite horizon reach-avoid controller.

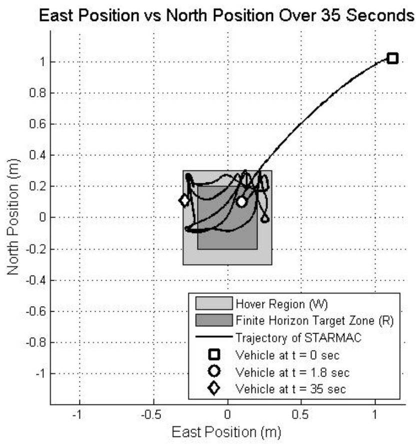


Fig. 5. **East Position (m) vs North Position (m) of STARMAC for 35 Seconds.**

Figure 7 shows another experimental trial where STARMAC first hovers over a remote-controlled ground vehicle using finite horizon reach-avoid controller and then follows the vehicle as it starts moving using infinite horizon invariant controller, over the course of 44 seconds. As depicted in the plot, the quadrotor reaches the finite horizon target zone within 2.1 seconds and begins to hover over the ground vehicle. The subsequent trajectory of the ground vehicle was human controlled, and not planned ahead of time. As such, STARMAC has to react at run-time to remain inside the hover region over the vehicle. Snapshots of the quadrotor and ground vehicle trajectory at about 20 seconds and 33 seconds into the experiment are shown in Figure 8. As can be seen, the quadrotor vehicle remains for the most part inside the hover region, except for a few brief violations, which in this case can be also attributed to occasional bursts in acceleration of the remote controlled ground vehicle.

V. Conclusion and Future Work

In this work, we proposed a reachability-based methodology for motion planning under dynamic uncertainty. Using a game theoretic formulation of the problem, we gave a systematic procedure for synthesizing feedback control policies to achieve high level objectives of safety, target attainability, and invariance, under bounded, time-varying disturbances, and discussed the implementation of the policies in real-time control of an autonomous vehicle.

There are several possible directions for future work. First, it can be seen from the experimental results that the formal guarantees relies on the availability of disturbance bounds.

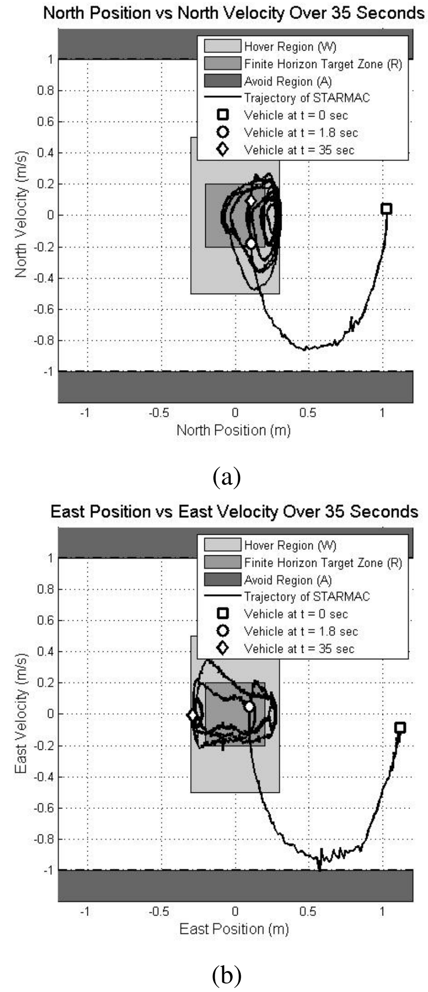


Fig. 6. **(a) East Position (m) vs East Velocity (m/s) of STARMAC for 35 Seconds; (b) North Position (m) vs North Velocity (m/s) of STARMAC for 35 Seconds.**

Methods are needed to rigorously characterize such bounds given certain physical models and operating conditions. Second, we note that in the current formulation of Hamilton-Jacobi equation, only a terminal cost is used to solve the reachability problem. A possible future direction would be to consider adding running cost, so as to choose control strategies which minimize a particular performance index. Finally, in order to apply this technique to higher dimensional systems, efficient schemes for approximation and representation of reachable sets will need to be investigated for particular forms of system dynamics.

References

[1] J. H. Reif, "Complexity of the mover's problem and generalizations," in *Foundations of Computer Science, 1979., 20th Annual Symposium on*, Oct. 1979, pp. 421 –427.

[2] J. T. Schwartz and M. Sharir, "On the piano movers' problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers," *Communications on Pure and Applied Mathematics*, vol. 36, no. 3, pp. 345–398, May 1983.

[3] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

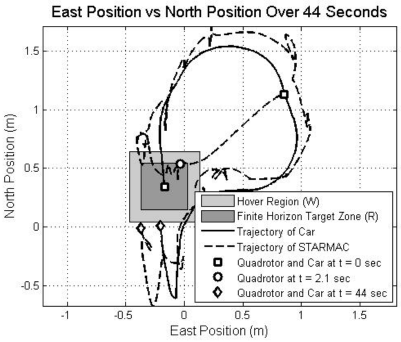
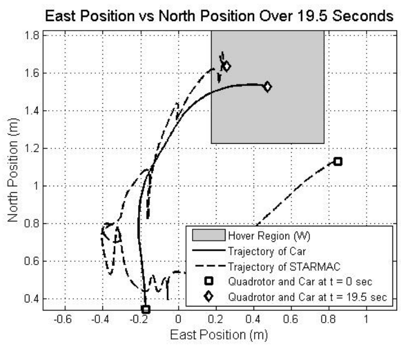
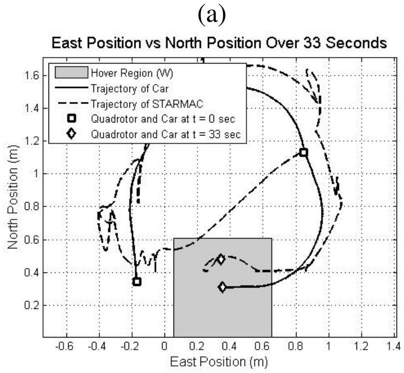


Fig. 7. **East Position (m) vs North Position (m) of STARMAC and ground vehicle for 44 seconds in car following experiment.**



(a)



(b)

Fig. 8. **Snapshots of East Position (m) vs North Position (m) of STARMAC and ground vehicle in car following experiment over: (a) 19.5 seconds; (b) 33 seconds.**

[4] T. Asano, T. Asano, L. Guibas, J. Hershberger, and H. Imai, "Visibility of disjoint polygons," *Algorithmica*, vol. 1, no. 1, pp. 49–63, 1986.

[5] T. Lozano-Perez, "Automatic planning of manipulator transfer movements," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 11, no. 10, pp. 681 –698, oct. 1981.

[6] K. Fujimura and H. Samet, "A hierarchical strategy for path planning among moving obstacles [mobile robot]," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 1, pp. 61–69, Feb 1989.

[7] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566 –580, aug. 1996.

[8] S. M. Lavalle and J. J. Kuffner, "Randomized kinodynamic motion planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[9] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[10] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 2, pp. 224 –241, Mar/Apr 1992.

[11] J. C. Latombe, *Robot Motion Planning*. Boston, Mass.: Kluwer Academic Publishers, 1991.

[12] T. Lozano-Prez, M. T. Mason, and R. H. Taylor, "Automatic Synthesis of Fine-Motion Strategies for Robots," *The International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.

[13] M. Erdmann, "Using Backprojections for Fine Motion Planning with Uncertainty," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 19–45, 1986.

[14] J. Canny, "On computability of fine motion plans," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, may. 1989, pp. 177 –182 vol.1.

[15] J.-C. Latombe, A. Lazanas, and S. Shekhar, "Robot motion planning with uncertainty in control and sensing," *Artificial Intelligence*, vol. 52, no. 1, pp. 1 – 47, 1991.

[16] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.

[17] B. D. Luders, S. Karaman, E. Frazzoli, and J. P. How, "Bounds on tracking error using closed-loop rapidly-exploring random trees," in *American Control Conference (ACC), 2010*, Jun. 2010, pp. 5406 – 5412.

[18] E. Frazzoli, M. Dahleh, and E. Feron, "Robust hybrid control for autonomous vehicle motion planning," in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 1, 2000, pp. 821–826.

[19] T. J. Koo, G. J. Pappas, and S. Sastry, "Mode switching synthesis for reachability specifications," in *Lecture Notes in Computer Science, Hybrid Systems: Computation and Control*, vol. 2034. Berlin, Germany: Springer-Verlag, 2001, pp. 333–346.

[20] H. Kress-Gazit, G. Fainekos, and G. Pappas, "Temporal-logic-based reactive mission and motion planning," *Robotics, IEEE Transactions on*, vol. 25, no. 6, pp. 1370 –1381, dec. 2009.

[21] C. Belta, V. Isler, and G. Pappas, "Discrete abstractions for robot motion planning and control in polygonal environments," *Robotics, IEEE Transactions on*, vol. 21, no. 5, pp. 864 – 874, oct. 2005.

[22] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349 – 370, 1999.

[23] C. Tomlin, J. Lygeros, and S. Shankar Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949 –970, Jul 2000.

[24] J. Lygeros, D. Godbole, and S. Sastry, "Verified hybrid controllers for automated vehicles," *Automatic Control, IEEE Transactions on*, vol. 43, no. 4, pp. 522 –539, Apr 1998.

[25] A. Bayen, I. Mitchell, M. Oishi, and C. Tomlin, "Aircraft autolander safety analysis through optimal control-based reach set computation," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 68–77, January-February 2007.

[26] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 3-7 2010, pp. 1649 –1654.

[27] J. Ding and C. J. Tomlin, "Robust reach-avoid controller synthesis for switched nonlinear systems," in *Decision and Control, 2010. CDC 2010. 49th IEEE Conference on*, Dec. 2010, accepted.

[28] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *Automatic Control, IEEE Transactions on*, vol. 50, no. 7, pp. 947–957, July 2005.

[29] 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*, vol. 3414/2005, pp. 480–494, Feb. 2005.

[30] G. Hoffmann, H. Huang, S. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proceedings of the AIAA Conference on Guidance, Navigation and Control*, Hilton Head, South Carolina, August 2007.