Real-Time Safe Path Tracking of Quadrotors

authors

Abstract— **Quadrotors have become very popular in research and industry for tasks that require exploration of unknown environments. However, path planning for autonomous vehicles is a computationally intensive task, and balancing speed of planning with safety and dynamic feasibility is challenging. Simplified models of quadrotor dynamics are easy to plan, but do not capture nonlinear behavior. Guaranteed safe paths can be computed for more realistic and complicated dynamics of quadrotors, but these paths require heavy computational load. We propose a method that combines these two approaches. First we precompute a guaranteed tracking error bound for the realistic dynamics following a simplified dynamics model using reachability analysis. We then perform path planning in real-time using the simplified model. This results in an invariant set around the simplified model that our quadrotor is guaranteed to remain in. Path planning method, results, etc.**

I. Introduction

Currently there is a great interest both in research and industry to find methods of path planning and model-predictive control (MPC) for autonomous quadrotors and other vehicles. These vehicles must be able to plan and execute a path in real-time without violating safety constraints. This is a very difficult challenge: the need for fast planning is generally at odds with the need for maintaining safety. In order to achieve real-time planning and model-predictive control (MPC) for any environment with static obstacles researchers typically must use highly simplified model dynamics. This results in a tracking error between the planned path and the true high-dimensional system. This concept is illustrated in Figure 1, where the path was planned using vehicle 1, but the real vehicle 2 cannot follow this path exactly. In addition, most current planners do not consider the effect of external disturbance (e.g. wind) on the resulting tracking error. This tracking error due to the simplified dynamics and lack of disturbances can lead to unsafe situations in which the planned path may be safe, but the actual vehicle trajectory crashes into an obstacle or other unsafe region.

We propose precomputing a bound on the possible tracking error between the path planned by the simplified model and the true high-dimensional vehicle dynamics. We compute this by using Hamilton Jacobi reachability analysis to analyze a capture-avoid game in relative coordinates between the true high-dimensional vehicle dynamics and a 'virtual' vehicle that uses the simplified model dynamics. Additional disturbances can be included in this analysis. The result is an invariant set of relative states around the virtual simplified vehicle that provides bounds on the possible tracking error between the two systems. This precomputed set also provides a look-up table to determine the optimal control required for the true vehicle to remain as close as possible to the virtual simplified vehicle. This set captures all deviations due to nonlinearities and disturbance in the true system.

This work has been supported in part by NSF under CPS:ActionWebs (CNS-931843), by ONR under the HUNT (N0014-08-0696) and SMARTS (N00014-09-1-1051) MURIs and by grant N00014-12-1-0609, by AFOSR under the CHASE MURI (FA9550-10-1-0567). The research of M. Chen has received funding from the "NSERC PGS-D" Program. The research of S. Herbert has received funding from the NSF GRFP and the UC Berkeley Chancellor's Fellowship Program

Fig. 1: filler image about planes chasing each other

We can couple this precomputed set with any model-preditive control method (MPC) that uses the simplified dynamics. As the MPC plans with the virtual vehicle, the true vehicle will use the relative state between itself and the virtual vehicle to look up the optimal control that will reduce tracking error. We can guarantee safety with the MPC method by expanding all encountered obstacles by the precomputed tracking error bound. The only additional computation required in real-time will be to access a look-up table with the optimal control for a given state.

We show our results in blablabla

II. Related Work

work on fast planning
work on safe planning
work on both
how ours is different

III. Computing Tracking Safety Radius

To precompute the tracking bound we must set up a capture-avoid game between the real and virtual vehicles, which we then analyze using HJ reachability. In this game, the real system will try to "capture" the virtual system, while the virtual system is doing everything it can to avoid capture. By using reachability to analyze this game we will get a guaranteed bound on how far apart the two vehicles will ever be even when the virtual system is acting as inconveniently as possible.

A. Individual and Relative Dynamics

Let z_1 be the state variable of the virtual system used for MPC planning, and z_2 be the state variable of the real system. The evolution of these states satisfies their respective ordinary differential equations:

$$\begin{aligned} \frac{dz_i}{ds} = \dot{z}_i = f_i(z_i, u_i), s \in [t, 0] \\ z_i \in \mathcal{Z}_i, u_i \in \mathcal{U}_i, i = 1, 2 \end{aligned} \tag{1}$$

Fig. 2: filler image about the implicit surface function

Fig. 3: filler image about the value function

We assume that the system dynamics $f_i : \mathcal{Z}_i \times \mathcal{U}_i \rightarrow \mathcal{Z}_i$ are uniformly continuous, bounded, and Lipschitz continuous in z_i for fixed control u_i. The control functions functions $u_i(\cdot)$ are drawn from the set of measurable functions[1]:

$$u_i(\cdot) \in \mathbb{U}_i(t) = \{\phi : [t, 0] \rightarrow \mathcal{U}_i : \phi(\cdot) \text{ is measurable}\} \quad i = 1, 2 \tag{2}$$

Under these assumptions there exists a unique trajectory solving 1 for a given $u_i(\cdot) \in \mathcal{U}_i$ [2]. The trajectories of 1 that solve this ODE will be denoted as $u(\cdot)$ as $\zeta_i(s; z_i, t, u_i(\cdot)), i = 1, 2$. These trajectories will satisfy the initial condition and the ODE 1 almost everywhere:

$$\begin{aligned} \frac{d}{ds}\zeta_i(s; z_i, t, u_i(\cdot)) &= f_i(\zeta_i(s; z_i, t, u_i(\cdot)), u_i(s)) \\ \zeta_i(t; z_i, t, u_i(\cdot)) &= z_i, \ \ i = 1, 2 \end{aligned} \tag{3}$$

We now have the dynamics for the individual systems, but to set up the capture-avoid game we must first define the relative states and dynamics. We place the virtual vehicle at the origin by subtracting its states (z_1) from the real system's states (z_2). In this frame of reference (z_r) we are given the states of the real system relative to the virtual system.

$$\begin{aligned} z_r =& z_2 - z_1 \\ g(z_r, u_1, u_2) =& f_2(z_2, u_2) - f_1(z_1, u_1) \end{aligned} \tag{4}$$

The relative dynamics will include the relative position states x_r, y_r and any other relevant states such as relative angles and velocities.

B. Formalizing the Capture-Avoid Game

Now that we have the relative dynamics between the two systems we must define a metric for the tracking error bound between these systems. We do this by defining an implicit surface function as a cost function in the new frame of reference. Because the metric we care about is distance to the origin, this cost function is a simple signed distance function in position space centered at the origin:

$$l(x_r, y_r) = \|[x_r, y_r]\|_2 \tag{5}$$

This can be seen in Figure 2, where the rings represent varying level sets of the cost function. The real vehicle will try to minimize this cost to reduce the relative distance, while the virtual vehicle will do the opposite.

We want to find the farthest distance (and thus highest cost) that this game will ever reach when both players are acting optimally. Therefore we want to find a mapping between the initial state of the system and the maximum cost achieved over the time horizon. This mapping is through our value function, defined as:

$$V(z_r) = \min_{u_2} \max_{u_1} \max_{t \in [0,T]} l(x_r(t), y_r(t)) \tag{6}$$

This is a modified version of the Hamilton-Jacobi formulation as described by cite jaime,mo, time varying reachability. explain why this is equivalent to taking the min (or in the way I wrote it, max) between the current value function and the target set

run to desired time or convergence conditions for convergence

results in Figure 3

guarantee to remain within current level set (proof) if p1 does not act optimally, can get into a closer bound eventual limit

can also add external disturbances easily

Goals overview: Compute invariant sets, given by the optimization solution

Properties of solution:

C. Trajectory Tracking as a Pursuit-Evasion Game

Dynamics of system

$$\dot{x} = f(x, u, d) \tag{7}$$

Dynamics of trajectory

$$\dot{y} = g(y, w) \tag{8}$$

Assume y is a subset of x. Let s be the set of indices such that $y_i = x_{s_i}$.

Relative state:

$$z = x - Ry \tag{9}$$

Relative dynamics

$$\dot{z} = r(z, u, w, d) \tag{10}$$

Tracking error: z_s.

The goal of the system is to minimize the tracking error.

The goal of the trajectory, which is a "virtual" vehicle, is to maximize the tracking error.

D. Optimization Problem

Define an error function $l(z)$

$$V(z, T) = \max_{u(\cdot)} \min_{w(\cdot), d(\cdot)} \min_{t \in [-T, 0]} l(\xi_r(t; z, -T, u(\cdot), w(\cdot), d(\cdot))) \tag{11}$$

Theorem 1: Let $T_c \geq 0$, and suppose

$$V_\infty(z) = V(z, T) = V(z, T_c) \ \forall T \geq T_c. \tag{12}$$

Then $\forall t_1, t_2$ with $t_1 \geq t_2$,

$$V_\infty(z) \leq V_\infty(\xi_r(t_2; z, t_1, u^*(\cdot), w^*(\cdot), d^*(\cdot))) \tag{13}$$

[1]A function $f : X \rightarrow Y$ between two measurable spaces (X, Σ_X) and (Y, Σ_Y) is said to be measurable if the preimage of a measurable set in Y is a measurable set in X, that is: $\forall V \in \Sigma_Y, f^{-1}(V) \in \Sigma_X$, with Σ_X, Σ_Y σ-algebras on X,Y.

where

$$
\begin{aligned}
u^*(\cdot) &= \arg\max_{u(\cdot)} \min_{w(\cdot),d(\cdot)} \min_{t\in[-T_1,0]} l(\xi_r(0;z,t,u(\cdot),w(\cdot),d(\cdot))) \\
w^*(\cdot) &= \arg\min_{w(\cdot)} \min_{d(\cdot)} \min_{t\in[-T_1,0]} l(\xi_r(0;z,t,u(\cdot),w(\cdot),d(\cdot))) \\
d^*(\cdot) &= \arg\min_{d(\cdot)} \min_{t\in[-T_1,0]} l(\xi_r(0;z,t,u(\cdot),w(\cdot),d(\cdot)))
\end{aligned} \tag{14}
$$

Proof: For all $T \geq T_c$.

$$
\begin{aligned}
V(z,T) &= \min_{t\in[-T,0]} l(\xi_r(t;z,-T,u^*(\cdot),w^*(\cdot),d^*(\cdot))) \\
&= \min\Big[\min_{t\in[-T_c,0]} l(\xi_r(t;z,-T,u^*(\cdot),w^*(\cdot),d^*(\cdot))), \\
&\qquad \min_{t\in[-T,-T_c]} l(\xi_r(t;z,-T,u^*(\cdot),w^*(\cdot),d^*(\cdot)))\Big] \\
V_\infty(z) &= \min\Big[V_\infty(\xi_r(-T_c;z,-T,u^*(\cdot),w^*(\cdot),d^*(\cdot))), \\
&\qquad \min_{t\in[-T,-T_c]} l(\xi_r(t;z,-T,u^*(\cdot),w^*(\cdot),d^*(\cdot)))\Big] \\
V_\infty(z) &\leq V_\infty(\xi_r(-T_c;z,-T,u^*(\cdot),w^*(\cdot),d^*(\cdot)))
\end{aligned} \tag{15}
$$

Since the system dynamics are time-invariant, we can pick $t_2 = -T_c$ without loss of generality, and $t_1 = -T$ to obtain the desired result. ■

Remark:

E. Dynamics of a Geometric Path

F. Solving the Optimization

HJ Reachability (1p)
Relative dynamics, setup, etc. (1p)
Capture basin computation (0.5p)

IV. Fast Path Planning using Rapidly-Exploring Random Tree

Potential methods to use (.5p)
Dealing with obstacles (.5p)

V. Results and Comparisons

demonstrate feasibility (.5)
real-time computation load (.5)
comparison to other methods (.5)

VI. Conclusions

Conclusion (0.5p)

References

[1] P. Varaiya, "On the existence of solutions to a differential game," *SIAM Journal on Control*, vol. 5, no. 1, pp. 153–162, 1967.

[2] L. C. Evans and P. E. Souganidis, "Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations," *Indiana Univ. Math. J.*, vol. 33, no. 5, pp. 773–797, 1984.

[3] E. Barron, "Differential Games with Maximum Cost," *Nonlinear analysis: Theory, methods & applications*, pp. 971–989, 1990.

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

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

[6] O. Bokanowski, N. Forcadel, and H. Zidani, "Reachability and minimal times for state constrained nonlinear problems without any controllability assumption," *SIAM Journal on Control and Optimization*, pp. 1–24, 2010.

[7] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Shankar, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *18th International Conference on Hybrid Systems: Computation and Controls*, 2015.

[8] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, "Reachability calculations for automated aerial refueling," in *IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

[9] M. Chen, Q. Hu, C. Mackin, J. Fisac, and C. J. Tomlin, "Safe platooning of unmanned aerial vehicles via reachability," in *IEEE Conference on Decision and Control*, 2015.

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

[11] H. Huang, J. Ding, W. Zhang, and C. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1451–1456.

[12] Jointed Planning and Development Office (JPDO), "Unmanned aircraft systems (UAS) comprehensive plan – a report on the nation's UAS path forward," Federal Aviation Administration, Tech. Rep., 2013.

[13] Amazon.com, Inc. (2016) Amazon prime air. [Online]. Available: http://www.amazon.com/b?node=8037720011

[14] BBC Technology. (2016) Google plans drone delivery service for 2017. [Online]. Available: http://www.bbc.com/news/technology-34704868

[15] AUVSI News. (2016) Uas aid in south carolina tornado investigation. [Online]. Available: http://www.auvsi.org/blogs/auvsi-news/2016/01/29/tornado

[16] National Aeronautics and Space Administration. (2016) Challenge is on to design sky for all. [Online]. Available: http://www.nasa.gov/feature/challenge-is-on-to-design-sky-for-all

[17] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. R. III, "Uas traffic management (utm) concept of operations to safely enable low altitude flight operations," in *AIAA Aviation Technology, Integration, and Operations Conference*, 2016.

[18] I. M. Mitchell and C. J. Tomlin, "Overapproximating reachable sets by hamilton-jacobi projections," *Journal of Scientific Computing*, vol. 19, no. 1-3, pp. 323–346, 2003.

[19] J. S. McGrew, J. P. How, L. Bush, B. Williams, and N. Roy, "Air combat strategy using approximate dynamic programming," *AIAA Guidance, Navigation, and Control Conference*, Aug 2008.

[20] J. B. Lasserre, D. Henrion, C. Prieur, and E. Trélat, "Nonlinear optimal control via occupation measures and lmi-relaxations," *SIAM Journal Control and Optimization*, vol. 47, no. 4, pp. 1643–1666, June 2008.

[21] I. M. Mitchell, "Scalable calculation of reach sets and tubes for nonlinear systems with terminal integrators: A mixed implicit explicit formulation," in *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, 2011, pp. 103–112.

[22] M. Chen and C. J. Tomlin, "Exact and efficient hamilton-jacobi reachability for decoupled systems," in *54th IEEE Conference on Decision and Control*, December 2015.

[23] I. M. Mitchell, "Comparing forward and backward reachability as tools for safety analysis," in *Proceedings of the 10th International Conference on Hybrid Systems: Computation and Control*, 2007.

[24] M. G. Crandall and P.-L. Lions, "Viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, vol. 277, no. 1, pp. 1–42, 1983.

[25] P. Bouffard, "On-board model predictive control of a quadrotor helicopter: Design, implementation, and experiments," Master's thesis, University of California, Berkeley, 2012.