Safety from in-the-loop reachability for cyber-physical systems

Christian Llanes
The Georgia Institute of Technology
Atlanta, Georgia

Matthew Abate
The Georgia Institute of Technology
Atlanta, Georgia

Samuel Coogan
The Georgia Institute of Technology
Atlanta, Georgia

ABSTRACT

We demonstrate a methodology for achieving safe autonomy that relies on computing reachable sets at runtime. Given a system subject to disturbances controlled by an unverified and potentially faulty controller, this methodology computes at each time the reachable set of the system under a backup control law to ensure the system is within reach of a known a priori safe region. Control barrier functions are then used in conjunction with the reachable set to adjust potentially unsafe control actions that would otherwise move the system beyond reach of this safe set. This approach faces several computational challenges: reachable sets for the dynamics must be computed at runtime; sensitivity of the reachable set to initial conditions is required for the control barrier optimization formulation; and the presence of disturbances introduces a large number of constraints in the resulting optimization. The proposed methodology leverages the theory of mixed monotone systems to address these challenges, and the main contribution of this paper is an application of this methodology to a ten dimensional dual planar multirotor system that is implemented on embedded hardware with a controller update rate up to 100Hz.

ACM Reference Format:
Christian Llanes, Matthew Abate, and Samuel Coogan. 2021. Safety from in-the-loop reachability for cyber-physical systems. In *Computation-Aware Algorithmic Design for Cyber-Physical Systems (CAADCPS'21), May 19–21, 2021, Nashville, TN, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3457335.3461706

1 INTRODUCTION

An approach to achieving safe autonomy is to protect an unverified and potentially faulty controller by checking proposed control actions and rejecting those that lead to safety violation. In such cases, an alternative safe control action may also be required from a known safe backup strategy. In general, this approach requires reasoning about the possible evolution of a system to determine if safety will be violated. For cyber-physical systems, this is often achieved offline by computing a safe, control-invariant kernel for the dynamics. However, finding an sufficiently large kernel can be practically challenging, and an alternative approach is to instead compute forward reachable sets of the dynamics at runtime to determine possible future safety violation on-the-fly.

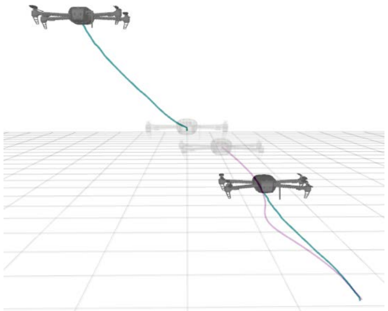

Figure 1: A depiction of trajectories from a nominally unsafe controller that leads to a collision and safe trajectories from filtering the unsafe controller with our RTA algorithm.

In [1], a runtime assurance (RTA) mechanism is proposed that leverages the theory of mixed monotone systems [3] for efficient reachable set computation. In this paper, we present an efficient instantiation of the proposed algorithm with appropriate novel modifications for real-time implementation. We apply this algorithm to a ten dimensional dual planar multirotor system implemented in a hardware-in-the-loop (HITL) experiment with an algorithm update rate up to 100Hz.

2 ALGORITHM OVERVIEW

We consider a cyber-physical system modeled as a control-affine and disturbance-affine continuous-time dynamical system with state $x \in \mathbb{R}^n$. We assume the existence of a backup feedback control law that renders invariant some backup region given as the super-zero level-set of a continuously differentiable function $h(x)$. We further assume the backup region avoids the unsafe region of the state-space but is generally conservative so that we wish to safely operate the system beyond this backup region.

Our primary assumption is that the dynamics under the backup control law are mixed monotone. A dynamical system is mixed-monotone if its vector field can be decomposed into increasing and decreasing components, in which case the decomposed dynamics are expressed in an embedding system with twice the number of states constructed via a decomposition function [3]. A single trajectory of the embedding system provides a hyperrectangular overapproximation of the reachable set for the original system.

We use the formalism of control barrier functions (CBFs) to guarantee that the hyperrectangular reachable set approximation under the backup control law over some horizon has all corners of the hyperrectangle contained in the safe backup region at some

This work is supported by the NASA University Leadership Initiative (ULI) under grant number 80NSSC20M0161.

CAADCPS'21, May 19–21, 2021, Nashville, TN, USA
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8399-8/21/05…$15.00
https://doi.org/10.1145/3457335.3461706

time $t^* \leq T_b$ for fixed backup horizon T_b. A separate CBF condition further ensures that the backup reachable set does not enter the unsafe region $\mathcal{X}_u$ along the horizon.

To construct the CBF condition ensuring that the reachable set remains within reach of the backup set, at each time $t \in [0, T_b]$ along the backup horizon, we compute the numerically stable and continuously differentiable Log-Sum-Exponential (LSE) soft-min of $h(x)$ evaluated for each of the 2^n corners of the hyperrectangular approximation of the reachable set at time t; for practical implementation, only a small sample of time instants is used. In our implementation, we selected these time points based on a novel log-based point distribution algorithm. This algorithm leverages the fact that the maximizing time varies continuously as the system evolves and thus considers more dense sample points close to the maximizing time in the previous iteration. By ensuring that the maximum over the time horizon of this LSE evaluation remains positive, we guarantee that for some time in the backup trajectory there will always be a fully contained reachable set in the backup region. We thus use this maximum evaluation as a CBF. Following standard CBF methodology [2], the overall result is a quadratic program that adjusts a nominal control input to ensure safety. Since reachable set computations are embedded in the CBF computation, the CBF optimization constraints require computing the sensitivity to initial state of the embedding system trajectory.

3 ALGORITHMIC IMPLEMENTATION

We now present an implementation of our RTA algorithm for a dual planar multirotor system, that is, two multirotors restricted to two-dimensional motion, each with three degrees of freedom. In our example, we use the Y-Z plane in the north-east-down aerospace reference frame. The overall system has ten states: four velocity states, two roll angle states, two angular velocity states, and two relative displacement states. The system is considered safe if the relative displacement remains within a ball of some nominal displacement; we consider relative displacement coordinates δ_y, δ_z. The four inputs are net thrust and angular acceleration for both multirotors, and bounded, additive disturbances affect angular and linear accelerations.

We choose a backup controller that asymptotically stabilizes δ_y and δ_z to their nominal values. We obtain a conservative, invariant safe backup region from a quadratic Lyapunov function for the linearized backup dynamics. We then consider an unsafe controller that commands both multirotors to the same Y-Z position, nominally leading to collision. The proposed RTA algorithm anticipates the unsafe nominal scenario and alters control inputs to maintain safety, as illustrated in Figure 1.

This case study is demonstrated in a HITL experiment using the Gazebo simulator as the physics engine with the PX4 autopilot software stack simulated on each multirotor for sensor and motor drivers and state estimation. The RTA control actions are computed on a Jetson Nano single-board computer running Robot Operating System (ROS) onboard each multirotor to send motor commands to PX4. We use 3DR Iris 1.5kg multirotors from a supported list of vehicles in PX4. The nominal displacement is taken to be $\delta_y = \delta_z = 1m$ in the horizontal and vertical direction, and the system is considered safe if it remains within a 0.25m ball of the nominal displacement, as illustrated in Figure 2.

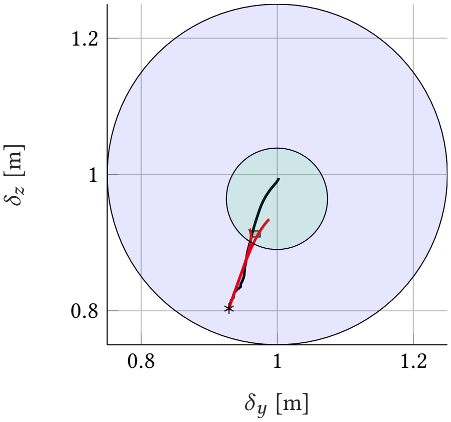

Figure 2: Trajectory and reachable set computations of dual planar multirotor system projected to the shifted relative displacement coordinates δ_y-δ_z with backup time horizon $T_b = 5s$ and simulation time $60s$. The multirotors are commanded with a nominal, unsafe controller that is adjusted and rendered safe by the RTA algorithm. The Gazebo simulation trajectory is shown in black, embedding backup trajectory in red, conservative safe backup region in green, and the safe region in blue.

Our algorithm is coded in C++ with single-precision, floating-point operations. At each controller update time, we compute the trajectory of the embedding system over a backup horizon $5s$ using forward Euler numerical integration, and we compute the sensitivity by simultaneously solving a matrix differential equation. These computations take between 4.7ms and 5.6ms. We use OSQP [4] embedded code generation to construct the convex CBF problem and generate a C static library that efficiently solves the problem in about $300\mu s$ to $900\mu s$. The optimization is a quadratic program with 132 constraints: 4 constraints on the control input magnitude, 64 constraints corresponding to the backup region, and 64 constraints corresponding to the unsafe set. The most time consuming operations are from evaluating $h(x)$ at each corner of the hyperrectangle along multiple time samples of the backup trajectory. The log-based time-sampling algorithm significantly reduces this computational burden; in this case study, satisfactory performance is obtained with six time samples along the backup trajectory. In total, we achieve a controller update rate of up to 100Hz.

REFERENCES

[1] M. Abate and S. Coogan. 2020. Enforcing Safety at Runtime for Systems with Disturbances. In *2020 59th IEEE Conference on Decision and Control (CDC)*. 2038–2043. https://doi.org/10.1109/CDC42340.2020.9304203

[2] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. 2019. Control Barrier Functions: Theory and Applications. In *2019 18th European Control Conference (ECC)*. 3420–3431. https://doi.org/10.23919/ECC.2019.8796030

[3] S. Coogan. 2020. Mixed Monotonicity for Reachability and Safety in Dynamical Systems. In *2020 59th IEEE Conference on Decision and Control (CDC)*. 5074–5085. https://doi.org/10.1109/CDC42340.2020.9304391

[4] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. 2020. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation* 12, 4 (2020), 637–672. https://doi.org/10.1007/s12532-020-00179-2