IEEE CSS

Received XX September 2021; revised ; accepted XX October 2021; Date of publication XX November 2021; date of current version XX November 2021. This paper was recommended by Associate Editor F. A. Author.

Digital Object Identifier 10.1109/OJCSYS.2021.Doi Number

Trajectory Tracking for Systems with Unknown Time-Varying Disturbances

Michael Enqi Cao[1] (Member, IEEE), Akash Harapanahalli[2] (Graduate Student Member, IEEE), Evanns Morales-Cuadrado[2] (Graduate Student Member, IEEE), Samuel Coogan[2] (Senior Member, IEEE)

[1]Unmanned Systems and Autonomy R&D Division, Sandia National Laboratories, Albuquerque, NM 87123, USA,
Previously: School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[2]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

CORRESPONDING AUTHOR: M. E. Cao (e-mail: mcao34@gatech.edu)

This work was supported in part by the National Science Foundation under awards #2333488 and #2440387.

ABSTRACT We consider trajectory tracking in nonlinear systems with partially unknown time-dependent and state-dependent dynamics. Given a reference trajectory, we leverage the mixed monotonicity property of dynamical systems to craft an embedding system, a single trajectory of which efficiently computes a forward invariant tube around the reference. We incorporate the unknown portions of the dynamics by modeling them as time-varying Gaussian processes, an existing machine learning technique, and combine them with tools from control theory via a general inclusion function which generates an associated embedding system that accounts for the first order interactions between the disturbance behavior, feedback controller, and system state. We illustrate the efficacy of this formulation through a trajectory tracking scenario in a software-in-the-loop simulation and demonstrate the viability of this method in practical implementation via physical hardware experiments on a quadrotor.

INDEX TERMS safe learning for control, time-varying systems, predictive control for nonlinear systems

I. Introduction

Safety-critical autonomous systems often require guarantees that they will only operate within an allowable safe region of their state-space. As a motivating example, in the context of urban air mobility, one key safety requirement is the ability of aerial vehicles to land safely, without damaging the vehicle or the environment around it, as illustrated in Figure 1. As these vehicles often operate in uncontrolled outdoor environments, they are subject to environmental disturbances that are not known a priori. While techniques exist for planning trajectories that are nominally safe [1], the presence of unknown disturbances in the environment means that precise tracking of these trajectories is a challenge. Thus, the goal of this work is to derive a runtime assurance mechanism that calculates a forward invariant tube around the reference trajectory which accounts for the unknown disturbance behavior. It then ensures accurate tracking of the reference trajectory by applying a feedback control strategy to preserve the forward invariance of the tube, and triggers a recalculation of the reference trajectory and invariant tube when close tracking can no longer be guaranteed.

A. Related Work

There are several general approaches to runtime assurance for safety in dynamical systems [2]. A common runtime assurance architecture is the Simplex architecture proposed by [3], where two controllers are developed for the system: one that is high-assurance and another that is high-performance. The high-performance controller is allowed to execute until some predetermined decision logic detects that the system is about to violate a safety specification, at which point control is switched to the high-assurance controller. This allows the high-performance controller to be developed without having to validate that it is safe beforehand. Alternatively, [4] introduces online active set invariance filtering, which instead minimally modifies the control action such that there exists a back-up trajectory that takes the system to a safe set, while never actually needing to execute the backup strategy. More recent approaches include tube-based

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

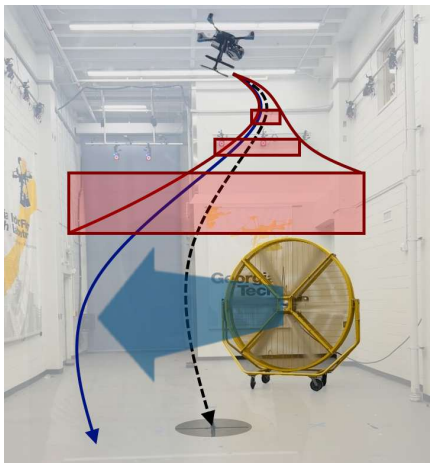


FIGURE 1: Motivating example: a multirotor attempting to follow a safe landing trajectory (black) in a windy environment. The unknown wind disturbance impedes the multirotor's ability to accurately track the reference trajectory. Thus, the objective of this work is to (i) calculate a tube (red) around the reference which encapsulates the disturbed trajectory (blue), (ii) detect when accurate tracking is infeasible, and (iii) execute a corrective action.

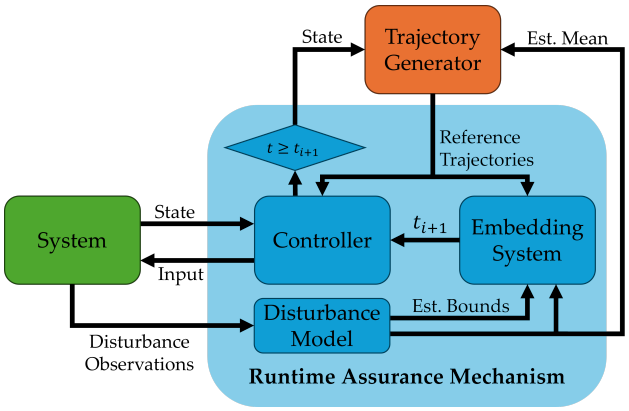


FIGURE 2: Overview of the proposed runtime assurance mechanism.

model predictive safety certification [5], wherein a model predictive controller certifies safety at runtime via ensuring the existence of a safe trajectory at all times, and [6], which guarantees safe tracking via precomputation of a tracking controller and a bound on tracking error.

Alternatively, [7] develops a runtime assurance mechanism for a distributed avionics architecture which can intervene in the event of failure. The paper [8] develops a nested control strategy consisting of an outer task-space loop and an inner joint-space loop for the trajectory tracking problem on a manipulator with uncertain kinematics and dynamics. Additionally, [9] leverages the mixed monotonicity property to produce reachable sets for stochastic systems.

Other approaches to the invariant tube synthesis problem (also known as the *funnel synthesis* problem) include [10], which develops a funnel synthesis algorithm for computing controlled invariant sets around a given nominal trajectory by solving a differential LMI, [11] which develops several strategies utilizing Sum-of-Squares programming for computing regions of finite-time invariance around solutions of polynomial differential equations, and [12] which proposes an approach to funnel synthesis that is based on falsification. Additionally, [13] develops a joint trajectory and funnel synthesis technique for discrete-time systems with locally Lipschitz nonlinearities. Finally, [14], [15] develop funnel synthesis strategies applied in real-time on aerial vehicles. In contrast to many of these works, we present a method that does not require the solving of an optimization problem.

There also exist works which incorporate tube estimations around a nominal trajectory in model predictive controllers, also known as Tube MPC. In [16], the authors outline a practical Tube MPC implementation on a ten-state quadrotor model, in [17], the authors implement Tube MPC onto an autonomous multi-vehicle racing system, and in [18], the authors develop a Dynamic Tube MPC framework for nonlinear systems. These techniques represent methods for generating trajectories that are safe with some margin of error, but generally do not incorporate learned disturbance behavior into tube calculations. In this work, we do not design nominal trajectories as in Tube MPC, and instead focus on learning the disturbance and incorporating resulting bounds into a runtime assurance mechanism. Alternatively, adaptive control [19] and disturbance-observer methods [20], [21] for trajectory tracking often provide Lyapunov-based stability guarantees, but guarantees are usually not provided for the system during its transient behavior.

With respect to the control of multirotor systems specifically, [22] develops a control law which tracks position and yaw angle up to their fourth order derivatives, [23] presents a linear-quadratic differential game approach which is used to develop an integrated estimation and control method for trajectory tracking, and [24] utilizes a geometric control approach with $\mathcal{L}_1$ adaptive augmentation. Crucially, these works are tailored toward control of multirotor systems, whereas this work, while motivated by a multirotor system, presents a method for the control of general nonlinear systems with unknown components.

B. Paper Overview

In this work, given a system with uncertain dynamics and a provided safe reference trajectory to track, we leverage mixed monotone systems theory to derive a runtime assurance mechanism that detects whether the tracking controller is unable to follow the reference trajectory within a desired tolerance. Specifically, we develop an inclusion function-based embedding system to calculate a forward invariant tube around the reference trajectory based on the current observations of the unknown disturbance behavior and the feedback control strategy. Next, we define a safe deviation

threshold which, if crossed, triggers an observation of the unknown dynamics and a recalculation of the reference trajectory and forward invariant tube, ensuring the system never deviates from the reference trajectory by more than the safe threshold. Our proposed mechanism natively accommodates nonlinear dynamics in continuous-time, and the observation collection trigger reduces the number of samples needed to ensure safety.

We empirically verify our approach using two complementary experiments. First, we conduct a case study using the PX4 flight stack in a software-in-the-loop (SITL) simulation on the Gazebo platform. This setup enables the simulation of strong wind disturbances, realistic sensor models with noise, and the use of the same flight stack and Kalman filter employed on hardware platforms. This high-fidelity simulation environment permits us to simulate wind speeds with an average magnitude of $17\,\mathrm{m/s}$, corresponding to near gale-force conditions — significantly exceeding what can be reproduced in a laboratory environment. Next, we demonstrate the real-world applicability of our method on a physical quadrotor operating in a windy environment created with industrial fans.

Our preliminary work [25] focused on developing a runtime assurance mechanism for trajectory tracking for systems with unknown time-invariant disturbance behavior. Here, we expand on this work with the following contributions:

- We leverage a generalized mixed Jacobian method for constructing an inclusion function and associated embedding system that produces forward invariant tubes around the reference trajectory for arbitrary dynamics. In contrast, [25] required the vector field to be interval-valued when evaluated over the input set, a restrictive assumption in practice.
- We incorporate time-varying disturbance behaviors via a time-varying Gaussian Process (GP) approach to bounding the disturbance behavior. Additionally, we implement this approach in a way that is compatible with our existing immrax pipeline [26] for efficient execution using JAX [27] for Python.
- We demonstrate the mechanism on both a software-in-the-loop setup mimicking physical hardware, as well as on a physical hardware setup.

The rest of this paper is structured as follows: we introduce key notation in Section II, define our problem in Section III, and provide key preliminaries in Section IV. We then derive our forward invariant tube formulation and runtime assurance mechanism in Section V, and apply it to case studies of a planar multirotor vehicle in Section VI. We conclude with a discussion in Section VII.

II. Notation

Let (x, y) denote the vector concatenation of $x, y \in \mathbb{R}^n$, i.e., $(x, y) := [x^T \; y^T]^T \in \mathbb{R}^{2n}$. Additionally, $\preceq$ denotes the componentwise vector order, i.e., $x \preceq y$ if and only if $x_i \leq y_i$ for all $i \in \{1, ..., n\}$ where vector components are indexed via subscript.

Given $x, y \in \mathbb{R}^n$ such that $x \preceq y$, we denote the interval defined by the endpoints x and y using the notation $[x, y] := \{z \in \mathbb{R}^n \mid x \preceq z \text{ and } z \preceq y\}$. Also, given $a = (x, y) \in \mathbb{R}^{2n}$ with $x \preceq y$, $[\![a]\!]$ denotes the interval formed by the first and last n components of a, i.e., $[\![a]\!] := [x, y]$. Let $\preceq_{\text{SE}}$ denote the *southeast order* on $\mathbb{R}^{2n}$ defined by $(x, x') \preceq_{\text{SE}} (y, y')$ if and only if $x \preceq y$ and $y' \preceq x'$. In particular, observe that when $x \preceq x'$ and $y \preceq y'$,

$$(x, x') \preceq_{\text{SE}} (y, y') \iff [y, y'] \subseteq [x, x']. \tag{1}$$

Additionally, denote by $\lfloor \cdot \rfloor, \lceil \cdot \rceil$ to be the lower and upper bounds on an interval set, i.e.

$$\lfloor [x, y] \rfloor = x, \lceil [x, y] \rceil = y. \tag{2}$$

Finally, $x_{[i:y]} \in \mathbb{R}^n$ denotes the vector $x \in \mathbb{R}^n$ whose ith component has been replaced with the ith component of vector $y \in \mathbb{R}^n$, i.e., $(x_{[i:y]})_j = x_j$ when $j \neq i$ and $(x_{[i:y]})_i = y_i$.

III. Problem Setup and Solution Outline

We consider the continuous-time system

$$\dot{x} = f(x, u, w) \tag{3}$$

where f is C^1 in its arguments, $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the system input, and $w \in \mathbb{R}^p$ is an unknown, time-varying, state-dependent component of the dynamics so that $w_i = g_i(t, x)$ where g_i is unknown. Denote by $\phi(t; x_0, u, w)$ the state of (3) at time t when initialized at x_0 under some input signal $u : [0, \infty) \to \mathbb{R}^m$ and some disturbance signal $w : [0, \infty) \to \mathbb{R}^p$.

Our overall objective is to design a controller that tracks an optimal trajectory in the face of an unknown disturbance w which potentially exhibits complex state- and time-dependent behavior. In particular, we wish to address scenarios for which the initial uncertainty in w is large, precluding traditional robust optimal control approaches, and for which we can accommodate online measurements of $g(t, x)$ to improve our estimate of the disturbance w. Additionally, motivated by practical hardware implementation, we seek to maximize computational efficiency while retaining tracking guarantees.

Problem Statement. *Given finite horizon cost, terminal constraints, and hard constraints on a trajectory that constitutes an optimal control problem, design a control strategy that seeks to minimize the cost while obeying terminal constraints and hard trajectory constraints, while allowing adaptation to learned knowledge of disturbance* $w = g(t, x)$.

To solve this problem, this paper proposes a model predictive assurance algorithm that incorporates GP learning of the uncertainty and on-the-fly reachability analysis to guarantee satisfaction of the hard constraints.

Our proposed solution approach is the following. At each dynamically chosen planning time t_r (whose design choice is described below):

1) Numerically solve for an optimal feedforward control input using the current best estimate of $g(t,x)$. We will use GP estimation to approximate g. Since $g(t,x)$ is fixed, this is a deterministic optimal control problem, and is therefore amenable to fast numerical computation, as shown in later sections. In solving for this optimal feedforward control, we bloat the constraints by some safe threshold Δ so that any trajectory within an Δ neighborhood of the solved trajectory is still guaranteed to satisfy both hard constraints and terminal constraints.
2) Efficiently compute a linear feedback control strategy that attempts to track the nominal state trajectory from the previous step. This linear feedback incorporates knowledge of the nonlinearities in the dynamics and the current best estimate of $g(t,x)$.
3) Use statistical estimation of the uncertainty in g to determine a reachable tube for the closed loop dynamics. Using this reachable tube, determine the next t_r, i.e. the next planning step, as the time for which the reachable tube is no longer guaranteed to stay within Δ of the nominal trajectory.
4) Repeat at the new t_r.

Trajectory generation for autonomous aerial vehicles has been addressed through a variety of formulations, including approaches based on optimal control [1], [28] and differential flatness [29], [30]. In this work, we assume such a method is available and instead focus on the problem of calculating a tube around the given reference trajectory in the presence of the unknown disturbance behavior $w = g(t,x)$, and crafting a runtime assurance mechanism to detect when the system may deviate beyond the safe threshold Δ.

Assumption 1. *There exists a mechanism to produce reference trajectories $x_r(t), u_r(t)$ which solve (3) for some nominal disturbance $w_r(t)$ that robustly satisfies the hard and terminal constraints, which, in particular, is such that remaining within Δ of x_r for all time satisfies all constraints.*

In other words, Assumption 1 states that step 1 of the proposed solution approach always produces a safe reference trajectory that solves a given optimal control problem with a buffer of Δ, such that remaining within the hyperrectangular tube defined by $[x_r(t) - \Delta, x_r(t) + \Delta]$ satisfies all constraints for all $t \geq 0$, which is achievable utilizing various approaches (e.g. [1], [28], [29], [30]).

However, because the true behavior of w is unknown, it may not be possible to track the reference trajectory within Δ. Thus, our main contribution is the formation of a runtime assurance mechanism that detects when the system may violate this Δ threshold, and triggers a recomputation of the reference trajectory, taking into account any updated knowledge of the unknown disturbance behavior. For an overview of the mechanism, see Figure 2.

To craft the desired runtime assurance mechanism, we first consider the problem of calculating a forward invariant tube around reference trajectories.

Subproblem 1. *Given a system (3), reference trajectories $x_r(t), u_r(t)$ which solve (3) for some $w_r(t)$, and linear feedback term K, produce a controlled forward invariant tube around the reference trajectory $x_r(t)$.*

Once this tube is calculated, the resulting runtime assurance mechanism must then detect when the system may violate safety, and then trigger a return to step 1, as outlined in steps 3 and 4 of the proposed solution approach. This is formalized in our second subproblem.

Subproblem 2. *Given a system (3) and reference trajectories $x_r(t), u_r(t)$ which solve (3) for some $w_r(t)$, devise an assurance mechanism that detects at what time instance the system will be unable to remain within a safe distance Δ of the reference trajectory and then triggers a recomputation, such that the system remains within Δ of x_r for all time.*

IV. Preliminaries

In this section, we provide an overview of mixed monotonicity and how it enables efficient calculation of reachable set overapproximations. We then illustrate how the introduction of time-varying GPs leads to overapproximations of reachable sets that hold with high probability, and introduce linear inclusions which enable the automated computation of embedding systems for mixed monotonicity. This formulation will be modified in subsequent sections to produce our forward invariant tube around a reference trajectory.

A. Inclusion Functions and Mixed Monotonicity

For some function $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$, a function $\mathsf{F} = [\underline{\mathsf{F}}, \overline{\mathsf{F}}] : \mathbb{IR}^n \times \mathbb{IR}^m \times \mathbb{IR}^p \to \mathbb{IR}^n$ is an *inclusion function* for f if for every $x \in [\underline{x}, \overline{x}] \in \mathbb{IR}^n$, every $u \in [\underline{u}, \overline{u}] \in \mathbb{IR}^m$, and every $w \in [\underline{w}, \overline{w}] \in \mathbb{IR}^p$,

$$f(x,u,w) \in \mathsf{F}([\underline{x},\overline{x}],[\underline{u},\overline{u}],[\underline{w},\overline{w}]) \\ = [\underline{\mathsf{F}}([\underline{x},\overline{x}],[\underline{u},\overline{u}],[\underline{w},\overline{w}]), \overline{\mathsf{F}}([\underline{x},\overline{x}],[\underline{u},\overline{u}],[\underline{w},\overline{w}])]. \tag{4}$$

Definition 1. *Given system (3) and an inclusion function F for f, the* $2n$-dimensional mixed monotone embedding system for f induced by F *is given by*

$$\dot{\underline{x}}_i = \underline{\mathsf{E}}([\underline{x},\overline{x}],[\underline{u},\overline{u}],[\underline{w},\overline{w}])_i := \underline{\mathsf{F}}([\underline{x},\overline{x}_{[i:\underline{x}]}],[\underline{u},\overline{u}],[\underline{w},\overline{w}])_i \tag{5}$$

$$\dot{\overline{x}}_i = \overline{\mathsf{E}}([\underline{x},\overline{x}],[\underline{u},\overline{u}],[\underline{w},\overline{w}])_i := \overline{\mathsf{F}}([\underline{x}_{[i:\overline{x}]},\overline{x}],[\underline{u},\overline{u}],[\underline{w},\overline{w}])_i. \tag{6}$$

When F is clear from context, we simply call this the embedding system *for (3).*

Denote the state of the embedding system defined by (5)–(6) at time t initialized at $[\underline{x}_0, \overline{x}_0]$ under some time-varying input signal $[\underline{u}, \overline{u}] : [0,\infty) \to \mathbb{IR}^m$ and time-varying disturbance signal $[\underline{w}, \overline{w}] : [0,\infty) \to \mathbb{IR}^p$ by $\Phi^{\mathsf{E}}(t; [\underline{x}_0, \overline{x}_0], [\underline{u}, \overline{u}], [\underline{w}, \overline{w}])$. A fundamental result of mixed

monotone systems theory is as follows: given

$$(\underline{x}_0, \overline{x}_0), (\underline{x}'_0, \overline{x}'_0) \in \mathbb{R}^n \times \mathbb{R}^n, \tag{7}$$
$$(\underline{u}, \overline{u}), (\underline{u}', \overline{u}') : [0, \infty) \to \mathbb{R}^m \times \mathbb{R}^m, \tag{8}$$
$$(\underline{w}, \overline{w}), (\underline{w}', \overline{w}') : [0, \infty) \to \mathbb{R}^p \times \mathbb{R}^p, \tag{9}$$

such that

$$(\underline{x}_0, \overline{x}_0) \preceq_{\text{SE}} (\underline{x}'_0, \overline{x}'_0) \ \forall_{t \geq 0}, \tag{10}$$
$$(\underline{u}(t), \overline{u}(t)) \preceq_{\text{SE}} (\underline{u}'(t), \overline{u}'(t)) \ \forall_{t \geq 0}, \tag{11}$$
$$(\underline{w}(t), \overline{w}(t)) \preceq_{\text{SE}} (\underline{w}'(t), \overline{w}'(t)) \ \forall_{t \geq 0}, \tag{12}$$

then, for all $t \geq 0$,

$$\Phi^{\mathsf{E}}(t; [\underline{x}_0, \overline{x}_0], [\underline{u}, \overline{u}], [\underline{w}, \overline{w}]) \preceq_{\text{SE}} \Phi^{\mathsf{E}}(t; [\underline{x}'_0, \overline{x}'_0], [\underline{u}', \overline{u}'], [\underline{w}', \overline{w}']). \tag{13}$$

Thus, provided that the system is initialized within $[\underline{x}_0, \overline{x}_0]$, the input signal is overapproximated by $[\underline{u}(t), \overline{u}(t)]$ for all $t \geq 0$, and the disturbance signal is overapproximated by $[\underline{w}(t), \overline{w}(t)]$ for all $t \geq 0$, then the hyperrectangle defined by $[\![\Phi^{\mathsf{E}}(t; [\underline{x}_0, \overline{x}_0], [\underline{u}, \overline{u}], [\underline{w}, \overline{w}])]\!]$ overapproximates the true reachable set of (3), *i.e.*

$$\phi(T; x_0, u, w) \subseteq [\![\Phi^{\mathsf{E}}(T; [\underline{x}_0, \overline{x}_0], [\underline{u}, \overline{u}], [\underline{w}, \overline{w}])]\!] \tag{14}$$

for all $T \geq 0$ and $x_0 \in [\underline{x}_0, \overline{x}_0]$.

Remark 1. *Mixed monotonicity is a specific embedding of the parameterized reachable set techniques proposed in [31], though we do not consider the parameterized reachability formulation in this work as techniques for the incorporation of state-dependent unknown disturbance behavior into the parameterized reachable sets have not yet been developed.*

For any system, there exists some inclusion function and associated embedding system E satisfying the necessary conditions. While there is some inherent conservatism in the fact that the embedding system produces hyperrectangular overapproximations of the reachable sets of the system, there also exist conditions to determine whether a particular embedding system will produce the tightest possible overapproximation of the reachable set from the class of all possible embedding systems [32]. Additionally, while closed form expressions of these functions can be manually derived (see, e.g., [33], [34]), we have also developed tools to automate this derivation [26], which we also demonstrate in Section VI.

B. Gaussian Processes and High Probability Reachable Sets

In this section, we recall how Gaussian processes can be used to learn function behavior from observations. We have shown previously in [35] that when the disturbance w is state-dependent but time-invariant, *i.e.*, $w_i = g_i(x)$ for some function $g_i(x)$, then it is possible to obtain bounding functions $\underline{\gamma}_i(\underline{x}, \overline{x})$ and $\overline{\gamma}_i(\underline{x}, \overline{x})$, where $\underline{\gamma}_i, \overline{\gamma}_i : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, such that for all $i \in \{1, \ldots, p\}$,

$$\underline{\gamma}_i(\underline{x}, \overline{x}) \leq g_i(x) \leq \overline{\gamma}_i(\underline{x}, \overline{x}), x \in [\underline{x}, \overline{x}] \tag{15}$$

with probability at least $1 - \eta$ for some $\eta \in (0, 1]$, via GP modeling. These functions can be inserted into the embedding system (5)–(6) to produce a new embedding system

$$\begin{aligned} \dot{\underline{x}}_i &= \underline{\mathsf{E}}([\underline{x}, \overline{x}], [\underline{u}, \overline{u}])_i \\ &:= \underline{\mathsf{F}}([\underline{x}, \overline{x}_{[i:\underline{x}]}], [\underline{u}, \overline{u}], [\underline{\gamma}_i(\underline{x}, \overline{x}_{[i:\underline{x}]}), \overline{\gamma}_i(\underline{x}, \overline{x}_{[i:\underline{x}]})])_i \end{aligned} \tag{16}$$

$$\begin{aligned} \dot{\overline{x}}_i &= \overline{\mathsf{E}}([\underline{x}, \overline{x}], [\underline{u}, \overline{u}])_i \\ &:= \overline{\mathsf{F}}([\underline{x}_{[i:\overline{x}]}, \overline{x}], [\underline{u}, \overline{u}], [\underline{\gamma}_i(\underline{x}_{[i:\overline{x}]}, \overline{x}), \overline{\gamma}_i(\underline{x}_{[i:\overline{x}]}, \overline{x})])_i \end{aligned} \tag{17}$$

which, in turn, produces hyperrectangular overapproximations of the reachable set of the system with probability at least $1 - \eta$. In this section, we extend this result to systems with time-varying disturbance behavior as defined in our problem setup, *i.e.*, with $w_i = g_i(t, x)$.

We first review the standard time-invariant model, where functions $g(x)$ are drawn from a GP with zero mean and kernel $k(\cdot, \cdot)$, and observations of $g(x)$ are perturbed by additive i.i.d. Gaussian noise $\mathcal{N}(0, \sigma^2)$ with zero mean and variance σ^2. Given a set of τ observations $\{y_j\}_{j=1}^{\tau}$ of the GP at points $\{x_j\}_{j=1}^{\tau}$, the surrogate functions of interest to approximate g_i are

$$\forall i \in \{1, \cdots, p\}, \quad \begin{cases} \overline{g}_i^{(\tau)}(x) := \mu_{\tau,i}(x) + \sqrt{\beta_\tau}\sigma_\tau(x) \\ \underline{g}_i^{(\tau)}(x) := \mu_{\tau,i}(x) - \sqrt{\beta_\tau}\sigma_\tau(x) \end{cases} \tag{18}$$

where β_τ represents a desired confidence level, $\mu_\tau(\cdot)$ is the posterior mean, and $\sigma_\tau(\cdot)$ is the posterior variance, computed according to the standard GP updates [36]:

$$\mu_\tau(x) := k_\tau(x)^T (K_\tau + \sigma^2 I)^{-1} y_\tau \tag{19}$$
$$k_\tau(x, x') := k(x, x') - k_\tau(x)^T (K_\tau + \sigma^2 I)^{-1} k_\tau(x') \tag{20}$$
$$\sigma_\tau^2(x) := k_\tau(x, x) \tag{21}$$

where $k_\tau(x) := (k(x_1, x), \cdots, k(x_\tau, x))$ and $K_\tau = [k(x_i, x_j)]$. $\underline{\gamma}, \overline{\gamma}$ are then derived by taking the corresponding min or max over $\underline{g}, \overline{g}$ as outlined in [35, Theorem 7].

To extend this formulation into the time-varying case, we adopt the framework in [37], wherein we make the following formal assumption.

Assumption 2. *Functions $g_i(t, x)$ are drawn from a zero-mean, time-varying GP with spatial kernel $k(\cdot, \cdot)$ and temporal kernel*

$$(1 - \epsilon)^{\frac{|t_1 - t_2|}{2}}. \tag{22}$$

Observations of $g_i(t, x)$ are perturbed by additive i.i.d. time-invariant Gaussian noise $\mathcal{N}(0, \sigma^2)$ with zero mean and variance σ^2.

$\epsilon \in [0, 1]$ represents the rate of change in g over time; setting $\epsilon = 0$ recovers the time-invariant model of g while $\epsilon = 1$ means that the behavior of g is entirely independent between any two observation times.

The observations $\{y_j\}_{j=1}^{\tau}$ are taken at times $\{t_j\}_{j=1}^{\tau}$, and thus, the updated posterior mean and variance equations for querying a point at time $t_{\tau+1}$ are

$$\widetilde{\mu}_{\tau+1}(x) := \widetilde{k}_\tau(x)^T (\widetilde{K}_\tau + \sigma^2 I)^{-1} y \tag{23}$$

$$k_{\tau+1}(x,x') := k(x,x') - \widetilde{k}_\tau(x)^T(\widetilde{K}_\tau + \sigma^2 I)^{-1}\widetilde{k}_\tau(x') \tag{24}$$

$$\widetilde{\sigma}^2_{\tau+1}(x) := k_{\tau+1}(x,x) \tag{25}$$

with

$$\widetilde{K}_\tau = K_\tau \odot [(1-\epsilon)^{|t_i-t_j|/2}]_{i,j=1}^{\tau} \tag{26}$$

$$\widetilde{k}_\tau(x) = k_\tau(x) \odot [(1-\epsilon)^{(t_{\tau+1}-t_i)/2}]_{i=1}^{\tau} \tag{27}$$

where $\odot$ is the Hadamard product. Thus, the surrogate functions of interest to approximate time-varying g_i are

$$\forall i \in \{1,\cdots,p\}, \quad \begin{cases} \overline{g}_i^{(\tau+1)}(x) := \widetilde{\mu}_{\tau+1,i}(x) + \sqrt{\beta_\tau}\widetilde{\sigma}_{\tau+1}(x) \\ \underline{g}_i^{(\tau+1)}(x) := \widetilde{\mu}_{\tau+1,i}(x) - \sqrt{\beta_\tau}\widetilde{\sigma}_{\tau+1}(x). \end{cases} \tag{28}$$

Then, for all $i \in \{1,\cdots,p\}$ and all $\tau \geq 1$, define for all $\underline{x} \preceq \overline{x}$,

$$\underline{\gamma}_i(t_{\tau+1},\underline{x},\overline{x}) := \min_{x \in [\underline{x}^{(\tau,-)},\overline{x}^{(\tau,+)}] \cap \mathcal{D}_\tau} \underline{g}_i^{(\tau+1)}(x) - \frac{1}{(\tau+1)^2}, \tag{29}$$

$$\overline{\gamma}_i(t_{\tau+1},\underline{x},\overline{x}) := \max_{x \in [\underline{x}^{(\tau,-)},\overline{x}^{(\tau,+)}] \cap \mathcal{D}_\tau} \overline{g}_i^{(\tau+1)}(x) + \frac{1}{(\tau+1)^2} \tag{30}$$

where $\underline{x}^{(\tau,-)}, \overline{x}^{(\tau,+)}, \mathcal{D}_\tau$ are defined as in [35, Equations 54-55].

The resulting $\underline{\gamma}_i(t,\underline{x},\overline{x})$ and $\overline{\gamma}_i(t,\underline{x},\overline{x})$, $\underline{\gamma}_i, \overline{\gamma}_i : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, are such that for all $i \in \{1,\ldots,p\}$,

$$\underline{\gamma}_i(t,\underline{x},\overline{x}) \leq g_i(t,x) \leq \overline{\gamma}_i(t,\underline{x},\overline{x}), x \in [\underline{x},\overline{x}] \tag{31}$$

for all t and all $\underline{x},\overline{x} \in \mathbb{R}^n$ with $\underline{x} \preceq \overline{x}$ with probability at least $1-\eta$. Thus, they can be directly inserted into the update dynamics for the high-probability embedding system (16)–(17) in place of the previous $\underline{\gamma},\overline{\gamma}$ to produce reachable set overapproximations of the system which hold with probability at least $1-\eta$.

Although rooted in probability theory, the derived posterior estimates can be interpreted simply as a way to reconstruct an unknown function from collected observations. In subsequent sections, we present a runtime assurance scheme that adopts this deterministic perspective and does not directly incorporate probability bounds, rather assuming the confidence parameter $\beta_\tau > 0$ is selected such that the resulting confidence bounds strictly contain the true function.

Finally, another contribution of this work is an implementation of these time-varying Gaussian processes in JAX for Python [27], a toolbox enabling the use of automatic differentiation and GPU acceleration. Our implementation is also fully compatible with `immrax` [26], our toolbox for interval analysis and bound propagation in JAX, which allows us to construct a first order mixed Jacobian inclusion for the mean function μ as discussed in the next section.

C. Open-Loop Abstraction via Linear Inclusions

Consider C^1 function $h : \mathbb{R}^a \to \mathbb{R}^b$ and given $z' \in \mathcal{Z} \subseteq \mathbb{R}^a$. A linear inclusion encompassing the error behavior of h with respect to z' is of the form $\mathcal{M}(z-z')$ where $\mathcal{M} \subseteq \mathbb{R}^{b\times a}$ is a set of matrices such that, for every $z \in \mathcal{Z}$,

$$h(z) - h(z') \in \mathcal{M}(z-z'). \tag{32}$$

There are several ways to obtain a matrix set $\mathcal{M}$ satisfying (32)—for instance, if $\overline{\text{co}}\{\frac{\partial h}{\partial x}(x)\} \subseteq \mathcal{M}$ for every $x \in \mathcal{X}$ (where $\overline{\text{co}}$ is the closed convex hull), then (32) holds for every $x \in \mathcal{X}$ by application of the mean value theorem.

In [38], exploiting the fact that z' is a priori fixed, an alternative algorithm is proposed to obtain $\mathcal{M}$, and we briefly recall this approach. Suppose we are given an interval subset $\mathcal{Z} = Z_1 \times Z_2 \times \cdots \times Z_q \subseteq \mathbb{R}^a$. Then, the following implication holds: for fixed $z' \in \mathcal{Z}$,

$$\frac{\partial h_i}{\partial z_j}(Z_1,\ldots,Z_j,z'_{j+1},\ldots,z'_q) \subseteq [\mathcal{M}]_{ij}, \ \forall i,j \tag{33}$$

$$\implies h(z) - h(z') \in [\mathcal{M}](z-z'), \ \forall z \in \mathcal{Z}. \tag{34}$$

An interval matrix $[\mathcal{M}]$ built algorithmically to satisfy (33) is called an *interval mixed Jacobian matrix*, since it mixes inputs to the Jacobian matrix of h between interval inputs and inputs fixed at the particular known z'. For example, we can apply this formulation to analyze the open-loop dynamics (3) when states, inputs, and disturbances are bounded to interval input sets $\mathcal{X} \in \mathbb{IR}^n, \mathcal{U} \in \mathbb{IR}^m, \mathcal{W} \in \mathbb{IR}^p$, by considering $f(x,u,w)$ as a mapping $\mathbb{R}^{n+m+p} \to \mathbb{R}^n$. The result is a linear differential inclusion encompassing the error dynamics of the system to a known point $(x',u',w') \in \mathcal{X} \times \mathcal{U} \times \mathcal{W}$ of the form

$$\begin{aligned} &f(x,u,w) - f(x',u',w') \\ &\in [\mathcal{M}_x](x-x') + [\mathcal{M}_u](u-u') + [\mathcal{M}_w](w-w'), \end{aligned} \tag{35}$$

where we have split the resulting interval matrix $[\mathcal{M}] \in \mathbb{IR}^{n\times(n+m+p)}$ into three interval matrices $[\mathcal{M}_x] \in \mathbb{IR}^{n\times n}$, $[\mathcal{M}_u] \in \mathbb{IR}^{n\times m}$, and $[\mathcal{M}_w] \in \mathbb{IR}^{n\times p}$.

Moreover, as will be demonstrated in the case studies, the toolbox `immrax` [26] implements the entire procedure of obtaining this inclusion automatically using automatic differentiation and interval analysis. In `immrax`, this function transformation is called `mjacM`.

As a result, given interval sets $\mathcal{X} := [\underline{x},\overline{x}]$, $\mathcal{U} := [\underline{u},\overline{u}], \mathcal{W} := [\underline{w},\overline{w}]$, and reference trajectories x_r, u_r, w_r initialized within $\mathcal{X},\mathcal{U},\mathcal{W}$, we formulate a generalized inclusion function for (3) as

$$\begin{aligned} \mathsf{F}([\underline{x},\overline{x}],[\underline{u},\overline{u}],[\underline{w},\overline{w}]) := \\ [\mathcal{M}_x]([\underline{x},\overline{x}] - x_r) + [\mathcal{M}_u]([\underline{u},\overline{u}] - u_r) \\ + [\mathcal{M}_w]([\underline{w},\overline{w}] - w_r) + f(x_r,u_r,w_r). \end{aligned} \tag{36}$$

Thus, we can craft the associated embedding system and insert our bounding functions $\underline{\gamma},\overline{\gamma}$ in place of $\underline{w},\overline{w}$ as in (16)–(17) to form an embedding system that produces a forward invariant tube around the reference trajectory which holds with at least the same probability that $\underline{\gamma},\overline{\gamma}$ hold, whose dynamics take the forms as in (64)–(65) in Appendix A.

The hyperrectangular tube given by $[\underline{x}(t),\overline{x}(t)]$ from the embedding system (64)–(65) represents a forward invariant tube around the given reference trajectories. In the next

sections, we refine the computation of this tube by leveraging the first-order interactions between a known feedback control strategy and the system state, as well as between the state-dependent disturbance behavior and system state.

V. Feedback Aware Forward Invariance

In this section, we specialize the approach introduced in Section IV-C to incorporate a known feedback control strategy and the first-order interactions between the estimated disturbance behavior and state, which tightens the controlled forward invariant tube around the reference trajectory. This tube is then inserted into a runtime assurance mechanism which guarantees the system will never deviate beyond a safe threshold of the reference, demonstrated to be computationally efficient enough to run on physical hardware platforms via the case studies in Section VI.

A. Incorporation of First-Order Interactions

Since we are attempting to track a reference trajectory, it is natural to formulate a feedback controller that counteracts any disturbance behavior which may cause the system to deviate from the reference. Suppose we consider a linear feedback controller of the form

$$u(t) = K(x(t) - x'(t)) + u'(t) \tag{37}$$

where $K \in \mathbb{R}^{m\times n}$, which, inserted into (35), results in

$$\begin{aligned} &f(x,u,w) - f(x',u',w') \in \\ &\quad [\mathcal{M}_x](x-x') + [\mathcal{M}_u](K(x-x') + u' - u') \\ &\quad + [\mathcal{M}_w](w-w'), \end{aligned} \tag{38}$$

which simplifies to

$$\begin{aligned} &f(x,u,w) - f(x',u',w') \in \\ &\quad ([\mathcal{M}_x] + [\mathcal{M}_u]K)(x-x') + [\mathcal{M}_w](w-w'). \end{aligned} \tag{39}$$

We then formulate the associated inclusion function

$$\begin{aligned} &\mathsf{F}([\underline{x},\overline{x}],[\underline{w},\overline{w}]) := \\ &\quad ([\mathcal{M}_x] + [\mathcal{M}_u]K)([\underline{x},\overline{x}] - x_r) + [\mathcal{M}_w]([\underline{w},\overline{w}] - w_r) \\ &\quad + f(x_r,u_r,w_r), \end{aligned} \tag{40}$$

which results in associated embedding system dynamics (66)–(67) in Appendix A. This embedding system no longer depends on $[\underline{u},\overline{u}]$, as it captures the known behavior of the feedback controller (37). Thus, the forward invariant tube $[\underline{x}(t),\overline{x}(t)]$ defined by dynamics (66)–(67) reflects the interactions between the feedback control strategy and system state in response to deviations from the reference trajectory caused by the disturbance behavior.

We further refine our forward invariant tube calculation by leveraging the fact that the disturbance is state-dependent; it may be possible for the disturbance behavior to have a stabilizing effect on the reference trajectory tracking in some areas of the state-space.

To capture this, presume there exists some known function $\mu(t,x)$ representing a nominal estimate of $g(t,x)$, and some function $\sigma(t,x)$ which bounds the difference between μ and g such that at a particular instance in time t, for every $x \in \mathcal{X}$, and for every $i = 1,\dots,p$,

$$w_i = g_i(t,x) = \mu_i(t,x) + d_i(t,x), \tag{41}$$

for some $|d_i(t,x)| \le \sigma_i(t,x)$. Applying the mixed Jacobian inclusion from the previous section to the known function $\mu(t,\cdot)$, we can compute an interval matrix $[\mathcal{M}_x^\mu]$ at time t such that

$$\mu(t,x) - \mu(t,x') \in [\mathcal{M}_x^\mu](x-x'). \tag{42}$$

From (41), setting $s \in \mathbb{R}^p$ by $s_i = \max_{x\in\mathcal{X}} \sigma_i(t,x)$, it follows that, setting $w' = \mu(t,x')$,

$$g(t,x) - w' \in [\mathcal{M}_x^\mu](x-x') + [-s,s] \tag{43}$$

which, due to the true behavior of w being defined by $w_i = g_i(t,x)$, can be inserted into (39) to obtain

$$\begin{aligned} &f(x,u,w) - f(x',u',w') \in \\ &([\mathcal{M}_x] + [\mathcal{M}_u]K)(x-x') + [\mathcal{M}_w]([\mathcal{M}_x^\mu](x-x') + [-s,s]), \end{aligned} \tag{44}$$

simplified to

$$\begin{aligned} &f(x,u,w) - f(x',u',w') \in \\ &([\mathcal{M}_x] + [\mathcal{M}_u]K + [\mathcal{M}_w][\mathcal{M}_x^\mu])(x-x') + [\mathcal{M}_w][-s,s]. \end{aligned} \tag{45}$$

This property then results in the associated inclusion function

$$\begin{aligned} \mathsf{F}([\underline{x},\overline{x}]) :=& ([\mathcal{M}_x] + [\mathcal{M}_u]K + [\mathcal{M}_w][\mathcal{M}_x^\mu])([\underline{x},\overline{x}] - x_r) \\ &+ [\mathcal{M}_w][-s,s] + f(x_r,u_r,\mu(t,x_r)), \end{aligned} \tag{46}$$

and associated embedding system dynamics (68)–(69) in Appendix A. Finally, we arrive at the following Theorem.

Theorem 1. *Given reference trajectories $x_r(t), u_r(t)$ that solve (3) for $w_r(t) = \mu(t,x_r(t))$, consider the feedback control strategy*

$$u(t) = K(x(t) - x_r(t)) + u_r(t),\ t \ge 0 \tag{47}$$

applied to (3) for some $K \in \mathbb{R}^{m\times n}$. If the true disturbance behavior is such that

$$w(t) = g(t,x) = \mu(t,x) + d(t,x) \tag{48}$$

for some $|d_i(t,x)| \le \sigma_i(t,x)$ for every i and all t,x, then

$$x(t) \in [\underline{x}(t),\overline{x}(t)]\ \forall t \ge 0 \tag{49}$$

where $[\underline{x}(t),\overline{x}(t)]$ is the interval tube produced by the associated embedding system dynamics (68)–(69) from initial conditions $\underline{x}(0),\overline{x}(0)$, with $x(0) = x_r(0) \in [\underline{x}(0),\overline{x}(0)]$.

Proof:
Per (45), we have that, for any $x, x_r \in [\underline{x},\overline{x}] \subseteq \mathbb{R}^n$, where u is dictated by (47) and $s = \max_{x\in[\underline{x},\overline{x}]} \sigma_i(t,x)$, it holds that

$$\begin{aligned} &f(x,u,w) - f(x_r,u_r,w_r) \in \\ &([\mathcal{M}_x] + [\mathcal{M}_u]K + [\mathcal{M}_w][\mathcal{M}_x^\mu])(x-x_r) + [\mathcal{M}_w][-s,s], \end{aligned} \tag{50}$$

or, alternatively,

$$\begin{aligned} &f(x,u,w) \in \\ &([\mathcal{M}_x] + [\mathcal{M}_u]K + [\mathcal{M}_w][\mathcal{M}_x^\mu])(x-x_r) + [\mathcal{M}_w][-s,s] \\ &+ f(x_r,u_r,w_r). \end{aligned} \tag{51}$$

Thus, the function (46) is a valid inclusion function with associated functions

$$\underline{\mathsf{F}}([\underline{x},\overline{x}]) = \lfloor([\mathcal{M}_x] + [\mathcal{M}_u]K + [\mathcal{M}_w][\mathcal{M}_x^\mu])([\underline{x},\overline{x}] - x_r) + [\mathcal{M}_w][-s,s] + f(x_r,u_r,w_r)\rfloor \tag{52}$$

$$\overline{\mathsf{F}}([\underline{x},\overline{x}]) = \lceil([\mathcal{M}_x] + [\mathcal{M}_u]K + [\mathcal{M}_w][\mathcal{M}_x^\mu])([\underline{x},\overline{x}] - x_r) + [\mathcal{M}_w][-s,s] + f(x_r,u_r,w_r)\rceil. \tag{53}$$

As a result, (49) holds via the fundamental result of mixed monotonicity (14). ■

Thus, the embedding system dynamics (68)–(69) capture the first-order interactions between the time-varying disturbance behavior and the system state, as well as the interactions between the feedback controller (37) and system state, which results in a more accurate forward invariant tube around the reference trajectory.

We have assumed that there exist known bounding functions μ, σ for the disturbance such that $w(t) = \mu(t,x) + d(t,x)$ with $|d_i(t,x)| \leq \sigma_i(t,x)$ for each i and all t, x, which results in Theorem 1 holding unconditionally and provides a solution to Subproblem 1. However, given that the disturbance behavior is deterministic but a priori unknown, a high probability estimated bound of the disturbance required for Theorem 1 can be obtained using the posterior estimates from the time-varying GP considered in the previous section. Defining $\mu(t,x) := \widetilde{\mu}_{\tau+1}(x)$ and $\sigma(t,x) := \sqrt{\beta_\tau}\widetilde{\sigma}_{\tau+1}(x)$ via (23)–(25) imposes their respective probability of correctly overapproximating the disturbance behavior on the resulting tube. In other words, if the true disturbance behavior is such that $w(t) = \mu(t,x) + d(t,x)$ for some $|d_i(t,x)| \leq \sigma_i(t,x)$ for each i with, e.g, probability at least $1-\eta$, then, given the posterior estimates from the time-varying GP, the guarantees of Theorem 1 would hold with probability at least $1-\eta$ as well. As a result, the hyperrectangular tube given by $[\underline{x}(t), \overline{x}(t)]$ from the embedding system (68)–(69) can, if desired, represent a refined *high-probability* forward invariant tube around the given reference trajectories.

In the next sections, we provide a numerical example which demonstrates the advantages of refining the hyperrectangular tube to include these interactions, and then formulate our final runtime assurance mechanism to solve Subproblem 2.

B. Numerical Example

We now provide a simple tutorial example demonstrating the effects of incorporating the first-order interactions of the observed disturbance behavior on the system. We consider the dynamics

$$\dot{x} = x + u + w \tag{54}$$

where the disturbance behavior $w = g(x)$ is dictated by the function $g(x) = -x$. We assume we have estimated disturbance bounds

$$\underline{\gamma}(t,\underline{x},\overline{x}) = \min_{x\in[\underline{x},\overline{x}]} -x - 1 \tag{55}$$

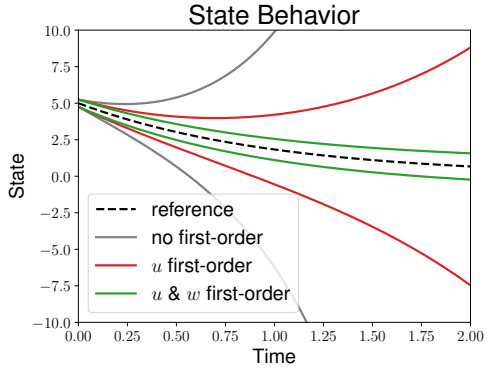


FIGURE 3: Demonstration of the advantages of including first-order interactions of the disturbance in the forward invariant tube calculation. By accounting for these interactions, the tube tightens around the reference trajectory.

$$\overline{\gamma}(t,\underline{x},\overline{x}) = \max_{x\in[\underline{x},\overline{x}]} -x + 1 \tag{56}$$

and known reference controller behavior $u_r = -x$ as well as feedback controller behavior $u = -(x - x_r) + u_r$. We initialize the system at $x = 5$ and then craft several embedding systems: one without modeling first-order disturbance interactions (i.e. (64)–(65)), one modeling only the first-order interactions between the input and state (i.e. (66)–(67)), and one modeling all first-order interactions (i.e. (68)–(69)). We initialize the embedding systems at $[x,\widehat{x}] = [4.75, 5.25]$ and simulate a 2 second trajectory. As shown in Figure 3, by accounting for the first-order interactions between the disturbance behavior and the system, the forward invariant tube remains tight around the reference trajectory, rather than quickly expanding.

C. Trajectory Tracking Runtime Assurance

We insert the forward invariant tube formulation outlined in previous sections into the runtime assurance algorithm introduced in [25]. After computing the embedding system trajectory, suppose it does not hold that

$$\bigwedge_{i=1}^{n} |\underline{x}_i(t) - x_{ri}(t)| \leq \Delta \wedge |\overline{x}_i(t) - x_{ri}(t)| \leq \Delta \tag{57}$$

for all $t \geq 0$. Then, a recomputation with new disturbance observations is triggered at the maximum time t_r such that

$$t_r \leq \min_{i\in[1,n]} \inf_{t\geq 0} (|\underline{x}_i(t) - x_{ri}(t)| \geq \Delta \vee |\overline{x}_i(t) - x_{ri}(t)| \geq \Delta) \tag{58}$$

after which a new reference trajectory is generated. Subsequently, a new feedback term K is computed via solving for a Linear Quadratic Regulator of the system linearized around the current state, along with the resulting forward invariant tube. The overall runtime assurance mechanism is outlined in Algorithm 1.

Remark 2. *In Algorithm 1, and in our case studies in section VI, t_r is set to the maximum value that satisfies* (58)

Algorithm 1 Runtime Assurance Mechanism

1: **Data:** Embedding system (68)–(69), safety threshold Δ
2: $t_r \leftarrow 0$
3: **while** In Operation **do**
4: **if** $t \geq t_r$ **then**
5: $K \leftarrow$ Solution to LQR of (3) linearized at $x(t)$
6: Generate Reference Trajectories x_r, u_r which solve (3) for $w_r = \widetilde{\mu}_t(x_r)$, $x_r(t) = x(t)$;
7: $\underline{x}, \overline{x} \leftarrow$ solution to embedding system dynamics (68)–(69), initialized from $\underline{x}(t) = \overline{x}(t) = x(t)$;
8: $t_r \leftarrow$ max value that satisfies (58);
9: Apply $u = K(x - x_r) + u_r$;

in order to reduce the amount of recomputation for the reference trajectory and forward invariant tube. If more frequent recomputation is desired, so long as t_r is set to a value greater than the current time t such that (58) is satisfied, the theoretical results of the algorithm still hold.

Given the inherent conservatism of the hyperrectangular sets, for practical application the desired safety threshold Δ and confidence bounds can be tuned to ensure that the computation completes in a timely manner, if needed. Additionally, if the computation time t_c of an iteration of Algorithm 1 is known, Remark 2 enables a buffer to be built in by setting t_r to the largest value that satisfies (58) minus t_c, and the parameters can be tuned to ensure it is still larger than t.

Assumption 3. *For brevity and concise presentation of Corollary 1 regarding Algorithm 1, we make a few simplifying assumptions for the state-based trigger t_r.*

1) *(Computation time). For simplicity, we assume that computation of the embedding dynamics and t_r in Algorithm 1 Lines (7)–(8) happens instantly to avoid the inclusion of a guard margin.*
2) *(Absence of Zeno phenomenon). We assume there exists a uniform lower bound $\underline{t}_r > 0$ where each subsequent recomputation time $t_r^{(k+1)}$ is such that $t_r^{(k+1)} - t_r^{(k)} \geq \underline{t}_r$ for all $k > 0$. There are several sufficient conditions, such as compactness of the state space, that would easily yield this bound.*

Practically, one can include a guard margin and a dwell time to explicitly account for these phenomena.

This brings us to the second theoretical result of this work.

Corollary 1. *Given reference trajectories $x_r(t), u_r(t)$ provided via Assumption 1 that solve (3) for $w_r(t) = \widetilde{\mu}_t(x_r(t))$, if the true disturbance behavior is such that*

$$w(t) = g(t,x) = \mu(t,x) + d(t,x) \tag{59}$$

for some $|d_i(t,x)| \leq \sigma_i(t,x)$ for every i and all t, x, then applying the runtime assurance mechanism outlined by Algorithm 1 guarantees that no state in the system deviates from the reference trajectory by more than Δ for all $t > 0$.

Proof:
At the instant $t = t_r$, the reference trajectory is calculated with $x_r(t) = x(t)$ and the tube is initialized with $\underline{x}(t) = \overline{x}(t) = x(t)$. Thus, it must always hold that the next $t_r > t$ given $\Delta > 0$ and condition (58).

As per Theorem 1, the tube defined by $[\underline{x}(t), \overline{x}(t)]$ is such that, so long as the true disturbance behavior is

$$w(t) = \mu(t,x) + d(t,x) \tag{60}$$

for some $|d_i(t,x)| \leq \sigma_i(t,x)$ for every i, then for all $t \geq 0$, applying the control strategy outlined by Algorithm 1 results in $x(t) \in [\underline{x}(t), \overline{x}(t)]$. Thus, combining Theorem 1 with condition (58) results in

$$x(\tau) \in [\underline{x}(\tau), \overline{x}(\tau)] \subseteq [x_r(\tau) - \Delta, x_r(\tau) + \Delta] \tag{61}$$

for all $t \leq \tau \leq t_r$. We proceed by induction to any $t \geq 0$, well-posed due to the lack of Zeno phenomenon imposed by Assumption 3. ■

Remark 3. *As before, we assume access to bounding functions μ, σ for the disturbance that satisfy the hypothesis of Corollary 1. In practice, we use the posterior estimates provided by the Gaussian process to learn these functions in (23)–(25) with a prespecified threshold β to tune the conservatism of the bounds, where a larger β results in the forward invariant tube expanding at a faster rate. Prior work [35] has shown that β_τ can be chosen so that these bounds hold for a desired probability level $1-\eta$ for a single application of Theorem 1. However, preserving a probability level across replanning with new disturbance observations, as done in Corollary 1, may require more care* (e.g., *a union bound), which we do not address here.*

Thus, Algorithm 1 solves Subproblem 2. Combined with a reference trajectory generator that satisfies Assumption 1, the overall runtime assurance mechanism solves our overall problem statement. In the next sections, we implement our forward invariant tube formulation on a planar multirotor system, first in simulation and subsequently on a software-in-the-loop setup.

VI. Case Studies

In this section, we demonstrate the forward invariant tube formulation in a trajectory tracking scenario. We first consider a software-in-the-loop setup in order to assess its computational viability for deployment as well as evaluate its performance in conditions more extreme than those achievable in a laboratory environment. We then demonstrate the assurance mechanism on a physical multirotor operating in a controlled wind environment.[1]

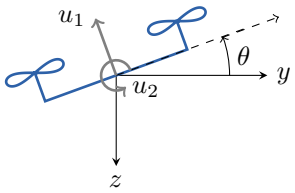


FIGURE 4: The planar multirotor model in an East-Down (Y-Z) inertial frame with horizontal position y, vertical position z, and roll angle θ. The inputs are thrust u_1 in the direction perpendicular to the line segment connecting the rotors and roll angle acceleration u_2.

A. Software-in-the-Loop Demonstration

To demonstrate the viability of our formulation on hardware, we test Algorithm 1 using a PX4 Autopilot [39] software-in-the-loop (SITL) simulation in the Gazebo simulator [40]. Because Algorithm 1 and its associated theoretical guarantees are formulated in continuous time and the SITL implementation is necessarily digital, the continuous-time tube guarantee does not directly constitute a formal sampled-data guarantee for the implemented SITL system, or the subsequent hardware experiments. However, the SITL and hardware results provide empirical evidence that the proposed approach remains computationally feasible and performs reliably under practical implementation effects, including zero-order-hold control, discrete state estimates, communication latency, and asynchronous sensor and actuator updates.

Unlike numerical simulators such as MuJoCo or PyBullet which only integrate rigid-body dynamics, PX4 SITL runs the exact same PX4 firmware used on physical flight controllers, with no modifications or simplifications. All estimation, control, motor-mixing, and safety checks execute in real time exactly as they would on embedded hardware.

Gazebo complements this by providing a hardware-realistic sensor stack (IMU, magnetometer, barometer, GPS, cameras, etc), complete with sensor noise, latency, and their own update rates, as well as actuator dynamics, aerodynamic disturbances, and asynchronous ROS 2 communication. All of this makes PX4 SITL in Gazebo fundamentally different from and substantially more realistic than physics-only simulators. Consistent with prior experience, controllers that perform reliably in this SITL environment generally transfer to hardware deployment with minimal modifications needed.

To preserve computational tractability and maintain parallel development, we assume the dynamics of a planar multirotor constrained to moving within the vertical plane, as in [25]. However, we extend the formulation to include wind disturbances in both the horizontal and vertical directions in order to capture coupled airflow effects that may be encountered in hardware testing environments (e.g., laboratory experiments employing high-power industrial fans

[1]Code, visualizations, and video demonstrations for these case studies can be found at https://github.com/evannsmc/PX4_RTA_MM_GPR/

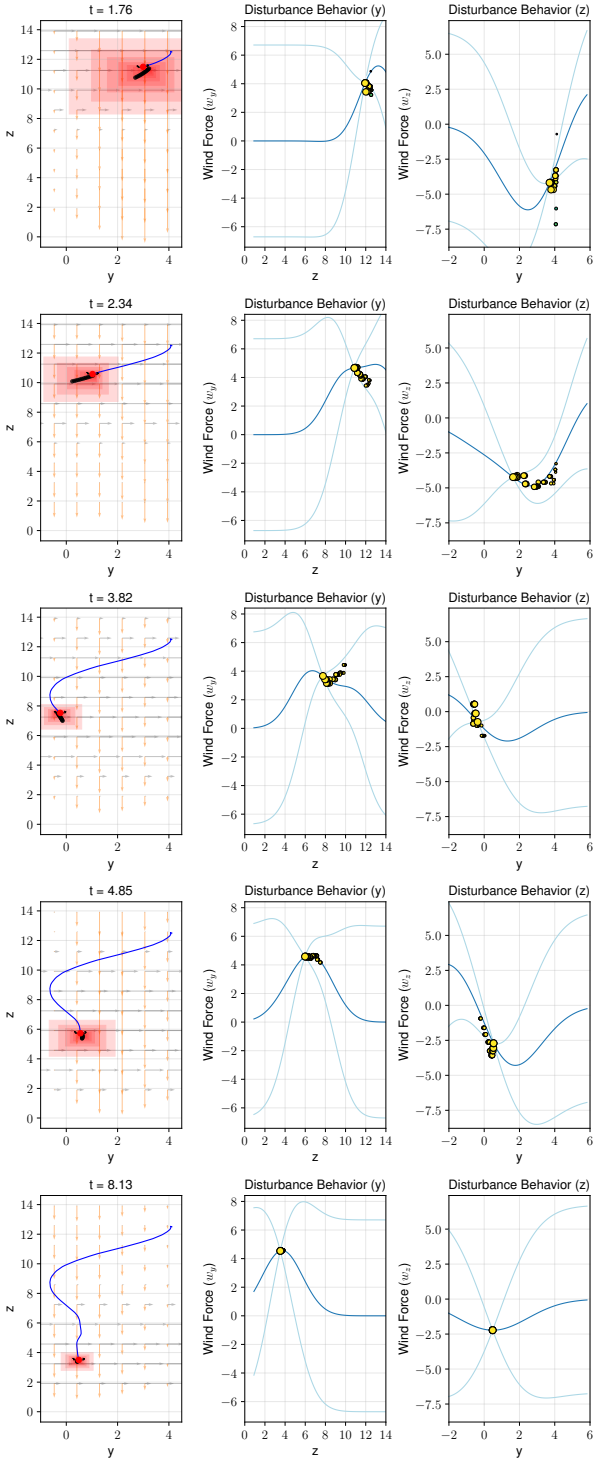


FIGURE 5: The software-in-the-loop multirotor trajectory tracking simulation. The multirotor attempts to track the reference trajectory (left, black) while the runtime assurance algorithm produces a forward invariant tube (left, rectangles) using observations (center & right, dots, largest most recent) to estimate the mean (left, orange & gray arrows, and center & right, dark blue) and confidence bounds (center & right, light blue) on the disturbance, which acts in both the y (gray) and z (orange) directions, then triggers an observation and recomputation at the appropriate times. The resulting trajectory is on the left in blue.

oriented diagonally to generate sustained wind disturbances). To estimate the disturbance behavior, we utilize a time-varying GP with a radial basis kernel ($\sigma_f = 5, l = 2, \sigma_n = 0.01, \epsilon = 0.25, \sqrt{\beta_\tau} = 3$) to model the unknown wind behavior in each axis.

Notably, a key strength of the Gazebo software-in-the-loop (SITL) framework is its ability to impose much harsher wind conditions than can be realized in laboratory-scale experimental setups. With simulated wind speeds averaging $17\,\mathrm{m/s}$, realistic physics and sensor modeling, and the same flight stack used on hardware platforms, the SITL framework enables high-fidelity evaluation of the performance of our proposed method under near gale-force conditions.

The five-dimensional state x of the planar multirotor system, illustrated in Figure 4, consists of horizontal position y, vertical position z, roll angle θ, and the derivatives $\dot{y}$ and $\dot{z}$, so that $x = \begin{bmatrix} y & \dot{y} & z & \dot{z} & \theta \end{bmatrix}^T$. The two inputs are thrust u_1 acting at the center of mass in the direction $\begin{bmatrix} -\sin\theta & \cos\theta \end{bmatrix}^T$ perpendicular to the line segment connecting the rotors, and roll angular velocity u_2. The time- and position-dependent wind forces acting on the multirotor, which are unknown a priori, are represented with $g_y(t,z)$ and $g_z(t,y)$, respectively. Utilizing an East-Down coordinate frame and non-unit mass m, the resulting dynamics of the planar quadrotor are

$$\begin{aligned} \ddot{y} &= u_1 \sin\theta + \frac{g_y(t,z)}{m} \\ \ddot{z} &= -u_1 \cos\theta + a_g - \frac{g_z(t,y)}{m} \\ \dot{\theta} &= u_2. \end{aligned} \tag{62}$$

We generate the reference trajectory by linearizing the system (62) around the equilibrium and then employing a Linear-Quadratic Regulator (LQR) feedback controller with parameters $Q = \mathrm{diag}([1000, 500, 20, 500, 1])$ and $R = \mathrm{diag}([20, 20])$ to simulate a trajectory to the origin assuming $g(t,z) = \widetilde{\mu}_t(z_r)$. The resulting state and input trajectories form the reference trajectories x_r, u_r that are used to calculate the forward invariant tube.

The true SITL system takes a full input vector $u = \begin{bmatrix} u_1, u_2, u_3, u_4 \end{bmatrix}^\top \in \mathbb{R}^4$, where u_3 and u_4 are pitch and yaw angular body rates. Thus, in order to ensure that the runtime assurance mechanism interacts with a system constrained to movement within the vertical plane, we employ a joint control strategy to calculate u_3 and u_4 separately from u_1 and u_2 to maintain a constant yaw and x position of zero. In our case, we utilize the Newton-Raphson-based control strategy [41], which is computationally efficient while providing accurate tracking in quadrotor hardware experiments [42]. Then, a second LQR controller, whose gains are computed alongside the reference trajectory, is utilized to provide stable tracking of the reference trajectory by u_1 and u_2. As shown by Figure 5, with this planar constraint, the multirotor is able to achieve a stable landing while in the presence of large wind disturbances.

TABLE 1: SITL computation-time statistics.

Computation	Average (ms)	Maximum (ms)
Joint controller	2.60 ± 0.69	6.77
Invariant-tube update	108.0 ± 5.9	134.0

We also note that when controlling body rates directly, new control inputs must be published to the quadrotor at an average of $100\,\mathrm{Hz}$ in order to achieve flight stability and reduce the risk of crash-landings in both SITL and hardware implementations. This means that the computation of Algorithm 1, the Newton-Raphson controller, and the recalculation of the LQR pipeline must take at worst $10\,\mathrm{ms}$ on average to avoid the risk of catastrophic results.

As Table 1 shows, solving (68)–(69) is relatively computationally expensive. However, this invariant-tube update is required only when $t \geq t_r$, wherein t_r was set according to (58) with $\Delta = 0.3$. This occurred 82 times during the SITL experiment, compared with 1085 control-input computations. Moreover, the update is performed asynchronously and does not block the control loop. Consequently, the joint controller requires an average of only $2.6\,\mathrm{ms}$ per update, well within the $10\,\mathrm{ms}$ control period. These computation times are enabled by our implementation in JAX and `immrax` [26], which uses just-in-time (JIT) compilation to improve the computational efficiency of both the runtime-assurance mechanism and the Newton–Raphson-based controller.

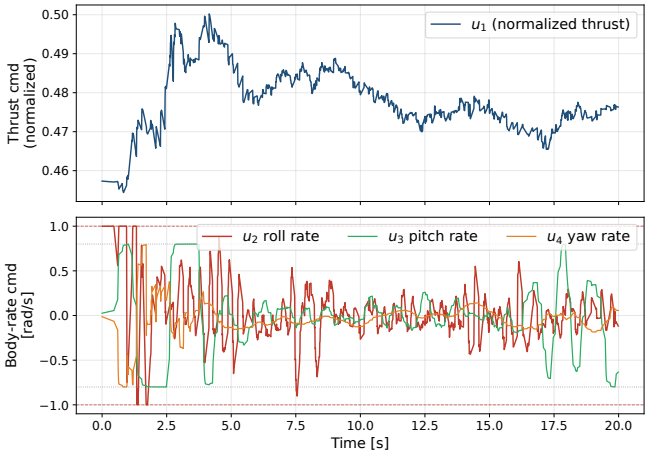


FIGURE 6: Actuation Inputs in Hardware Experiments

B. Hardware Demonstration

Having demonstrated the applicability of Algorithm 1 under near gale-force conditions in SITL, we now deploy the same pipeline on a physical quadrotor platform.[2] The hardware experiment is conducted indoors with a high-power fan oriented to produce a sustained wind disturbance, the setup

[2]Full visualizations and hardware video demonstrations can be found in the repository.

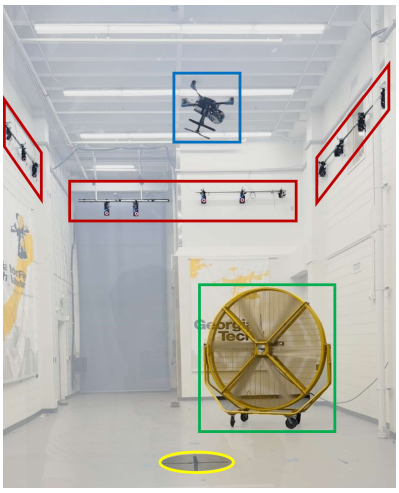


FIGURE 7: The setup for hardware experiments in the Indoor Flight Laboratory at the Georgia Institute of Technology. The quadrotor (blue), tracked by a Vicon motion capture system (red), attempts to make a safe landing on the landing pad (yellow) while in the presence of wind created by an industrial fan (green).

of which is illustrated in Figure 7. While the fan-induced airflow is milder than the $17\,\mathrm{m/s}$ simulated winds used in the SITL evaluation, it is sufficient to meaningfully perturb the vehicle throughout the mission and constitutes an unmodeled real-world disturbance of the form assumed in (62).

The mission is structured as follows: the quadrotor takes off and flies to an initial position displaced from the origin, holding a brief hover while awaiting the mission start. It then navigates back toward a point above the origin against the a priori unknown wind disturbance, sustaining a hover there for a predetermined period before executing a safe landing. Notably, the fan is positioned adjacent to the origin, so the disturbance grows in magnitude as the vehicle approaches its goal. Inputs of body rates and collective thrust over the course of one of these experiments are shown in Figure 6.

As Figure 8 shows, when wind estimation is enabled, the drone continuously collects disturbance observations and updates the TV-GP estimate of $g_y(t,z)$ and $g_z(t,y)$. We estimate wind forces via predicted acceleration residuals as

$$\hat{g}(t,x) = m(\ddot{x}_{\text{meas}}(t) - \ddot{x}_{\text{pred}}(x(t),u(t))). \tag{63}$$

These estimates are fed directly into the forward invariant tube computation (68)–(69), and enable the runtime assurance mechanism to plan reference trajectories that actively account for the wind disturbance behavior. Additionally, on average, the forward invariant tube remained within the safety threshold ($\Delta = 0.3$) of the reference trajectory for 0.2 seconds, and throughout the hardware experiments, absolute tracking error remained consistently under 0.25. The drone successfully maintains hover above the origin for a designated period, counteracting the fan disturbance throughout, and then executes a safe landing.

TABLE 2: Hardware computation-time statistics.

Computation	Average (ms)	Maximum (ms)
Joint controller	2.61 ± 0.69	6.95
Invariant-tube update	85.8 ± 22.0	144.6

Table 2 reports the computation times measured on the physical platform. The runtime-assurance mechanism is the same executable running on the same computer in both the SITL and the hardware campaigns. The joint-controller statistics are accordingly indistinguishable between Table 1 and Table 2. The invariant-tube update is faster on hardware on average, $85.8\,\mathrm{ms}$ against $108.0\,\mathrm{ms}$ in SITL, which we attribute to the SITL campaign additionally running the Gazebo physics engine and the PX4 flight stack on the same machine.

Over the 390 tube updates recorded across the hardware flights, the update occupied on average 44% of the recomputation interval it certifies, and the same fraction holds in SITL. The time between the completion of an update and the triggering of the next recomputation, which is the margin that the guard construction described following Remark 2 would have had available, averaged $109.4\,\mathrm{ms}$ on hardware and $136.0\,\mathrm{ms}$ in SITL. It remained positive on all 386 hardware and 77 SITL recomputation intervals, with minima of $40.4\,\mathrm{ms}$ and $87.0\,\mathrm{ms}$ respectively. In the single most demanding hardware cycle the update still consumed only 73% of its interval. Computation therefore never exhausted the horizon certified by the preceding tube, although we note that this is an empirical observation over the reported flights rather than an a priori bound.

In contrast, experiments in which wind estimation was disabled resulted in mission failures[3]. With no observations informing the GP, the disturbance estimate remains $\widetilde{\mu}_t \equiv 0$ throughout the flight, so the forward invariant tube (68)–(69) is computed under the false assumption that no wind acts on the system. The vehicle is therefore left with neither a trajectory that counters the wind nor a replanning cadence needed to adapt to it. Rather than navigating towards the origin, the quadrotor is observed to dive toward the ground, unable to complete the mission. This ablation study demonstrates that the TV-GP disturbance model allows the system to perform its mission in the face of a priori unknown disturbances. More generally, if wind were accounted for improperly, the guarantee in Theorem 1 would not hold, as it requires the GP bounds to contain the true disturbance and for Assumption 1 to also hold. It may therefore fail if the disturbance varies faster than assumed, exceeds the confidence bounds, or is underestimated because of errors in the acceleration-residual observations, including filtering, model mismatch, and estimator latency. The guarantee would also no longer apply if the reference generator cannot produce a feasible safe trajectory from the current state (i.e. if Assumption 1 does not hold).

[3]As can be seen in the video demonstrations in the repository.

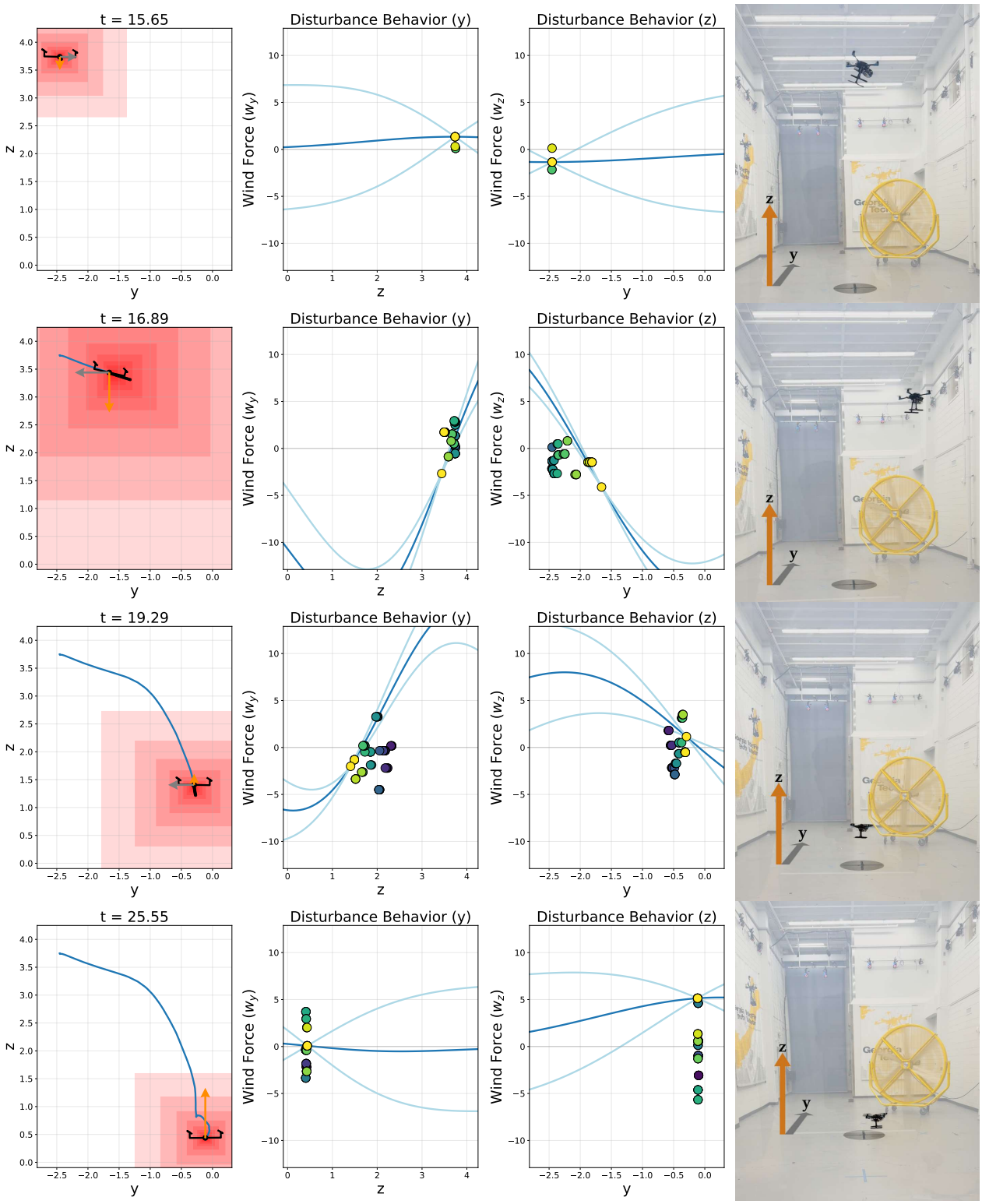


FIGURE 8: Hardware experiment with wind estimation enabled. The quadrotor tracks the reference trajectory while the runtime assurance algorithm produces a forward invariant tube (left, rectangles) using time-weighted observations (middle two columns, dots, lightest most recent) to estimate the GP mean and confidence bounds (middle two columns, dark and light blue, respectively) on the horizontal and vertical wind disturbances $g_y(t, z)$ and $g_z(t, y)$, respectively. Wind arrows at the drone in the left plots indicate the GP-estimated disturbance magnitude in each direction (z gray, y orange) at the current state, and the overall trajectory performed by the drone is plotted in blue. As shown in the video frames along the right, the drone is able to successfully hold station above the origin and then land safely.

VII. Conclusion

We present a general interval mixed-Jacobian-based formulation for efficiently computing a forward invariant tube around a reference trajectory for nonlinear systems subject to time- and state-dependent disturbance behaviors. By collecting time-weighted observations and modeling the disturbance behavior as a Gaussian Process, the formulation produces a forward invariant tube which inherits the probability of the GP. Additionally, we incorporate the first-order interactions between the feedback controller, disturbance behavior, and the system state, tightening the approximation. Taken together, the SITL and hardware experiments confirm that our JAX and `immrax`-based pipeline is both robust enough to withstand near-gale force disturbances and fast enough for real-world hardware deployment.

Appendix A
Embedding System Dynamics

We provide here the embedding system dynamics at each stage of first-order incorporation in order to better illustrate the differences between each. First, as described in Section IV-C, utilizing the Mixed Jacobian operator, the general embedding system for producing an invariant tube around a reference trajectory has the dynamics

$$\begin{aligned}\dot{\underline{x}}_i =& \underline{\mathsf{E}}([\underline{x},\overline{x}],[\underline{u},\overline{u}])_i \\ :=& \underline{\mathsf{F}}([\underline{x},\overline{x}_{[i:\underline{x}]}],[\underline{u},\overline{u}],[\underline{\gamma}_i(t,\underline{x},\overline{x}_{[i:\underline{x}]}),\overline{\gamma}_i(t,\underline{x},\overline{x}_{[i:\underline{x}]})])_i \\ =& \lfloor [\mathcal{M}_x]([\underline{x},\overline{x}_{[i:\underline{x}]}]-x_r)+[\mathcal{M}_u]([\underline{u},\overline{u}]-u_r)+[\mathcal{M}_w]([\underline{\gamma}_i(t,\underline{x},\overline{x}_{[i:\underline{x}]}),\overline{\gamma}_i(t,\underline{x},\overline{x}_{[i:\underline{x}]})]-w_r)+f(x_r,u_r,w_r)\rfloor_i \end{aligned} \tag{64}$$

$$\begin{aligned}\dot{\overline{x}}_i =& \overline{\mathsf{E}}([\underline{x},\overline{x}],[\underline{u},\overline{u}])_i \\ :=& \overline{\mathsf{F}}([\underline{x}_{[i:\overline{x}]},\overline{x}],[\underline{u},\overline{u}],[\underline{\gamma}_i(t,\underline{x}_{[i:\overline{x}]},\overline{x}),\overline{\gamma}_i(t,\underline{x}_{[i:\overline{x}]},\overline{x})])_i, \\ =& \lceil [\mathcal{M}_x]([\underline{x}_{[i:\overline{x}]},\overline{x}]-x_r)+[\mathcal{M}_u]([\underline{u},\overline{u}]-u_r)+[\mathcal{M}_w]([\underline{\gamma}_i(t,\underline{x}_{[i:\overline{x}]},\overline{x}),\overline{\gamma}_i(t,\underline{x}_{[i:\overline{x}]},\overline{x})]-w_r)+f(x_r,u_r,w_r)\rceil_i. \end{aligned} \tag{65}$$

Next, as outlined in Section V-A, incorporating a known control strategy $u = K(x - x_r) + u_r$ into the embedding system results in a controlled forward invariant tube around the reference trajectory produced by dynamics

$$\begin{aligned}\dot{\underline{x}}_i =& \underline{\mathsf{E}}([\underline{x},\overline{x}])_i \\ :=& \underline{\mathsf{F}}([\underline{x},\overline{x}_{[i:\underline{x}]}],[\underline{\gamma}_i(t,\underline{x},\overline{x}_{[i:\underline{x}]}),\overline{\gamma}_i(t,\underline{x},\overline{x}_{[i:\underline{x}]})])_i \\ =& \lfloor ([\mathcal{M}_x]+[\mathcal{M}_u]K)([\underline{x},\overline{x}_{[i:\underline{x}]}]-x_r)+[\mathcal{M}_w]([\underline{\gamma}_i(t,\underline{x},\overline{x}_{[i:\underline{x}]}),\overline{\gamma}_i(t,\underline{x},\overline{x}_{[i:\underline{x}]})]-w_r)+f(x_r,u_r,w_r)\rfloor_i \end{aligned} \tag{66}$$

$$\begin{aligned}\dot{\overline{x}}_i =& \overline{\mathsf{E}}([\underline{x},\overline{x}])_i \\ :=& \overline{\mathsf{F}}([\underline{x}_{[i:\overline{x}]},\overline{x}],[\underline{\gamma}_i(t,\underline{x}_{[i:\overline{x}]},\overline{x}),\overline{\gamma}_i(t,\underline{x}_{[i:\overline{x}]},\overline{x})])_i \\ =& \lceil ([\mathcal{M}_x]+[\mathcal{M}_u]K)([\underline{x}_{[i:\overline{x}]},\overline{x}]-x_r)+[\mathcal{M}_w]([\underline{\gamma}_i(t,\underline{x}_{[i:\overline{x}]},\overline{x}),\overline{\gamma}_i(t,\underline{x}_{[i:\overline{x}]},\overline{x})]-w_r)+f(x_r,u_r,w_r)\rceil_i. \end{aligned} \tag{67}$$

Finally, incorporating current knowledge of the unknown disturbance behavior as outlined in Section V-A, via a nominal estimate $\mu(t,x)$ and bounded difference $d(t,x) \leq \sigma(t,x)$, with $s = \max_{x\in[\underline{x},\overline{x}]}\sigma_i(t,x)$, results in a tighter controlled forward invariant tube produced by embedding system dynamics

$$\begin{aligned}\dot{\underline{x}}_i =& \underline{\mathsf{E}}([\underline{x},\overline{x}])_i \\ :=& \underline{\mathsf{F}}([\underline{x},\overline{x}_{[i:\underline{x}]}])_i \\ =& \lfloor ([\mathcal{M}_x]+[\mathcal{M}_u]K+[\mathcal{M}_w][\mathcal{M}_x^{\mu}])([\underline{x},\overline{x}_{[i:\underline{x}]}]-x_r)+[\mathcal{M}_w][-s,s]+f(x_r,u_r,w_r)\rfloor_i \end{aligned} \tag{68}$$

$$\begin{aligned}\dot{\overline{x}}_i =& \overline{\mathsf{E}}([\underline{x},\overline{x}])_i \\ :=& \overline{\mathsf{F}}([\underline{x}_{[i:\overline{x}]},\overline{x}])_i \\ =& \lceil ([\mathcal{M}_x]+[\mathcal{M}_u]K+[\mathcal{M}_w][\mathcal{M}_x^{\mu}])([\underline{x}_{[i:\overline{x}]},\overline{x}]-x_r)+[\mathcal{M}_w][-s,s]+f(x_r,u_r,w_r)\rceil_i. \end{aligned} \tag{69}$$

REFERENCES

[1] H. Heidari and M. Saska, "Trajectory planning of quadrotor systems for various objective functions," *Robotica*, vol. 39, no. 1, pp. 137–152, 2021.

[2] D. E. J. van Wijk, S. Coogan, T. G. Molnar, M. Majji, and K. L. Hobbs, "Disturbance-robust backup control barrier functions: Safety under uncertain dynamics," *IEEE Control Systems Letters*, vol. 8, pp. 2817–2822, 2024.

[3] L. Sha, "Using simplicity to control complexity," *IEEE Software*, vol. 18, no. 4, pp. 20–28, 2001.

[4] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An online approach to active set invariance," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 3592–3599, 2018.

[5] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 7130–7135, 2018.

[6] M. Chen, S. L. Herbert, H. Hu, Y. Pu, J. F. Fisac, S. Bansal, S. Han, and C. J. Tomlin, "Fastrack:a modular framework for real-time motion planning and guaranteed safe tracking," *IEEE Transactions on Automatic Control*, vol. 66, no. 12, pp. 5861–5876, 2021.

[7] S. Ghori, T. Khamvilai, E. Feron, and M. Pakmehr, "Runtime assurance for distributed avionics architecture," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, pp. 1–6, 2022.

[8] M. Tarokh, "Manipulator task space trajectory tracking with kinematics and dynamics uncertainties," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9884–9890, 2021.

[9] V. Sivaramakrishnan, R. A. Devonport, M. Arcak, and M. M. Oishi, "Forward reachability for discrete-time nonlinear stochastic systems via mixed-monotonicity and stochastic order," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 5177–5184, 2024.

[10] T. Kim, P. Elango, T. P. Reynolds, B. Açıkmeşe, and M. Mesbahi, "Optimization-based constrained funnel synthesis for systems with lipschitz nonlinearities via numerical optimal control," *IEEE Control Systems Letters*, vol. 7, pp. 2875–2880, 2023.

[11] M. M. Tobenkin, I. R. Manchester, and R. Tedrake, "Invariant funnels around trajectories using sum-of-squares programming," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 9218–9223, 2011. 18th IFAC World Congress.

[12] J. Fejlek and S. Ratschan, "Computing funnels using numerical optimization based falsifiers," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 4318–4324, 2022.

[13] T. Kim, P. Elango, and B. Acikmese, "Joint synthesis of trajectory and controlled invariant funnel for discrete-time systems with locally lipschitz nonlinearities," 2023.

[14] T. Reynolds, D. Malyuta, M. Mesbahi, B. Acikmese, and J. M. Carson, *Funnel Synthesis for the 6-DOF Powered Descent Guidance Problem*. 2021.

[15] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[16] H. Hu, X. Feng, R. Quirynen, M. E. Villanueva, and B. Houska, "Real-time tube mpc applied to a 10-state quadrotor model," in *2018 Annual American Control Conference (ACC)*, pp. 3135–3140, 2018.

[17] A. Wischnewski, T. Herrmann, F. Werner, and B. Lohmann, "A tube-mpc approach to autonomous multi-vehicle racing on high-speed ovals," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 368–378, 2023.

[18] B. T. Lopez, J.-J. E. Slotine, and J. P. How, "Dynamic tube mpc for nonlinear systems," in *2019 American Control Conference (ACC)*, pp. 1655–1662, 2019.

[19] N. Hassan and A. Saleem, "Neural network-based adaptive controller for trajectory tracking of wheeled mobile robots," *IEEE Access*, vol. 10, pp. 13582–13597, 2022.

[20] X. Yang, P. Wei, Y. Zhang, X. Liu, and L. Yang, "Disturbance observer based on biologically inspired integral sliding mode control for trajectory tracking of mobile robots," *IEEE Access*, vol. 7, pp. 48382–48391, 2019.

[21] X. Wang and W. Sun, "Trajectory tracking of autonomous vehicle: A differential flatness approach with disturbance-observer-based control," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1368–1379, 2023.

[22] E. Tal and S. Karaman, "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1203–1218, 2021.

[23] B. Jha, U. Gupta, V. Turetsky, and T. Shima, "Game theoretic approach for robust trajectory tracking by a multirotor," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 3, pp. 1435–1442, 2023.

[24] Z. Wu, S. Cheng, P. Zhao, A. Gahlawat, K. A. Ackerman, A. Lakshmanan, C. Yang, J. Yu, and N. Hovakimyan, "$\mathcal{L}_1$quad: $\mathcal{L}_1$ adaptive augmentation of geometric control for agile quadrotors with performance guarantees," 2024.

[25] M. E. Cao and S. Coogan, "Trajectory tracking runtime assurance for systems with partially unknown dynamics," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11525–11531, 2024.

[26] A. Harapanahalli, S. Jafarpour, and S. Coogan, "immrax: A parallelizable and differentiable toolbox for interval analysis and mixed monotone reachability in jax," *IFAC-PapersOnLine*, vol. 58, no. 11, pp. 75–80, 2024. 8th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2024.

[27] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, Y. Katariya, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018.

[28] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1485–1491, 2011. 18th IFAC World Congress.

[29] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *ASME international mechanical engineering congress and exposition*, Citeseer, 1995.

[30] A. Chamseddine, Y. Zhang, C. A. Rabbath, C. Join, and D. Theilliol, "Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 2832–2848, 2012.

[31] A. Harapanahalli and S. Coogan, "Parametric reachable sets via controlled dynamical embeddings," 2025.

[32] M. Abate, M. Dutreix, and S. Coogan, "Tight decomposition functions for continuous-time mixed-monotone systems with disturbances," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 139–144, 2021.

[33] M. E. Cao and S. Coogan, "Safe learning-based predictive control from efficient reachability," in *AACC American Control Conference (ACC)*, 2023.

[34] M. Abate, M. Mote, M. Dor, C. Klett, S. Phillips, K. Lang, P. Tsiotras, E. Feron, and S. Coogan, "Run time assurance for spacecraft attitude control under nondeterministic assumptions," *IEEE Transactions on Control Systems Technology*, pp. 1–12, 2023.

[35] M. E. Cao, M. Bloch, and S. Coogan, "Efficient learning of hyperrectangular invariant sets using gaussian processes," *IEEE Open Journal of Control Systems*, pp. 1–14, 2022.

[36] C. E. Rasmussen and C. K. I. Williams., *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: MIT Press, 2006.

[37] I. Bogunovic, J. Scarlett, and V. Cevher, "Time-varying gaussian process bandit optimization," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics* (A. Gretton and C. C. Robert, eds.), vol. 51 of *Proceedings of Machine Learning Research*, (Cadiz, Spain), pp. 314–323, PMLR, 09–11 May 2016.

[38] A. Harapanahalli and S. Coogan, "A linear differential inclusion for contraction analysis to known trajectories," 2024.

[39] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6235–6240, 2015.

[40] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Sendai, Japan), pp. 2149–2154, Sep 2004.

[41] Y. Wardi, C. Seatzu, J. Cortés, M. Egerstedt, S. Shivam, and I. Buckley, "Tracking control by the newton–raphson method with output prediction and controller speedup," *International Journal of Robust and Nonlinear Control*, vol. 34, no. 1, pp. 397–422, 2024.

[42] E. Morales-Cuadrado, C. Llanes, Y. Wardi, and S. Coogan, "Newton-raphson flow for aggressive quadrotor tracking control," in *2024 American Control Conference (ACC)*, pp. 3879–3884, 2024.



MICHAEL ENQI CAO (Member, IEEE) received a B.S. in Computer Engineering with a Minor in Robotics in 2018, a M.S. in Electrical Engineering in 2020, and a Ph.D. in Robotics in 2025, which he completed under the supervision of Dr. Samuel Coogan, all from the Georgia Institute of Technology, Atlanta, GA, USA. He has worked for The Aerospace Corporation, NASA's Jet Propulsion Lab, and the Johns Hopkins University Applied Physics Lab, and is currently a Postdoctoral Appointee in the Unmanned Systems and Autonomy R&D Division at Sandia National Laboratories, Albuquerque, NM, USA.

His research involves the safe control and verification of systems with partially unknown dynamics. These unknown dynamics are approximated via a combination of computationally efficient reachability tools such as mixed monotonicity which enable real-time evaluation, and robust function estimators such as Gaussian Processes which provide high-probability confidence bounds on the unknown components.



AKASH HARAPANAHALLI (Graduate Student Member, IEEE) received his B.S. in Computer Engineering in 2022 and his M.S. in Mathematics in 2025, and is currently pursuing his Ph.D. in Machine Learning under the supervision of Dr. Samuel Coogan, all from the Georgia Institute of Technology, Atlanta, GA, USA. He received the 2025-2026 Herbert P. Haley Fellowship from the Machine Learning Ph.D. program at Georgia Tech.

Akash's research is centered around safety verification of modern nonlinear control systems through reachability and invariance analysis. He has been developing the toolbox immrax, used to scalably certify and train safe performance of neural networks in safety-critical control systems.



EVANNS MORALES-CUADRADO (Graduate Student Member, IEEE) received the B.S. degree in Electrical Engineering, with minors in mathematics and physics, from The University of Texas at Arlington, Arlington, TX, USA, in 2022. He is currently pursuing the Ph.D. degree in Robotics, with a doctoral minor in Optimization Methods, at the Georgia Institute of Technology, Atlanta, GA, USA, under the supervision of Dr. Samuel Coogan.

He is also a Graduate Research Intern in AI for Autonomy at Sandia National Laboratories, Albuquerque, NM, USA. His research interests include safe autonomy, nonlinear and learning-based control, trajectory planning and generation, and the deployment of advanced control and planning methods on aerial and ground robotic systems.



SAMUEL COOGAN (Senior Member, IEEE) received the B.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, USA in 2010 and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, Berkeley, CA, USA in 2012 and 2015.

He is currently an associate professor and the Demetrius T. Paris Junior Professor at the Georgia Institute of Technology, Atlanta, GA, USA in the School of Electrical and Computer Engineering and the School of Civil and Environmental Engineering. Prior to joining Georgia Tech in 2017, he was an assistant professor at the University of California, Los Angeles from 2015 to 2017.

Prof. Coogan received a CAREER Award from the National Science Foundation in 2018, a Young Investigator Award from the Air Force Office of Scientific Research in 2019, and the Donald P Eckman Award from the American Automatic Control Council in 2020. He is a member of SIAM and a senior member of IEEE.