2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Trajectory Tracking Runtime Assurance for Systems with Partially Unknown Dynamics

Michael E. Cao and Samuel Coogan

Abstract— **We consider the problem of tracking a reference trajectory for dynamical systems subject to a priori unknown state-dependent disturbance behavior. We propose a formulation that embeds the uncertain system into a higher dimensional deterministic system that accounts for worst case disturbances. Our main insight is that a single controlled trajectory of this embedding system corresponds to a controlled forward invariant interval tube around the reference trajectory. By taking observations of the system, we then propose to estimate the state-dependent uncertainty with Gaussian Process regression, which improves the accuracy of the forward invariant tube as data is collected. Given a safety objective, we also provide conditions on when an additional observation of the unknown disturbance behavior needs to be collected to maintain safety. We demonstrate our formulation on a case study of a planar multirotor attempting a safe landing in an unknown wind field.**

I. Introduction

Safety-critical autonomous systems often require guarantees that they will only operate within an allowable safe region of their statespace. As a motivating example considered throughout this paper, in the context of urban air mobility, one key safety requirement is the ability of aerial vehicles to land without damaging the vehicle or the environment around it. As these vehicles are operating 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], ensuring that the system is able to track these trajectories during runtime in the presence of unknown disturbances is challenging. The goal of this work is to design a runtime-assurance framework that ensures safety in the presence of unknown disturbances. Our approach is to compute forward invariant tubes for the system given the limits of the control input and the current knowledge of the disturbance behavior, and override a nominal tracking control law if the system is at risk of violating its safety restrictions.

In many cases, uncertainty in the dynamics can be learned via observations. For example, the position-dependent wind field that an aerial vehicle encounters might be unknown a priori but can be learned through observations of the wind at different points. For such systems, in [2], [3], we derive high probability bounds on the unknown disturbance behavior by modeling the disturbance as a state-dependent Gaussian Process (GP). In turn, these bounds enable us to calculate overapproximations of the reachable sets of the system that hold with high probability. For this, we leverage the *mixed monotonicity* property of dynamical systems to embed the dynamics in a higher dimensional system [4], [5]. A key advantage of this overapproximation technique is its computational efficiency: it reduces the reachable set computation to the evaluation of a single trajectory of an *embedding system* with twice the number of states as the original system. This computation is able to be updated in real-time, even for systems of moderately high dimension [6], and we leverage this property to develop an approach that dynamically detects when a new observation of the disturbance behavior needs to be collected to keep the system within an acceptable bound of the reference trajectory. We then update the reference trajectory and forward invariant tube with the new observation during runtime to preserve safety.

There are several methods for ensuring safety at runtime, which is commonly referred to as runtime assurance. The most common runtime assurance architecture is the Simplex architecture proposed by [7], 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 run 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, [8] 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. Another approach is introduced in [9], which develops a formal language to implement runtime assurance. An illustrative example of practical runtime assurance for unmanned aerial systems is implemented in [10], though that method is implemented at the waypoint and trajectory selection level.

With regards to assurance for systems with partially unknown dynamics, [11] develops a Twin Neural Lyapunov Function which is then used to build a runtime monitor. However, like many neural network applications, a large amount of training data is needed to ensure safety, and the formulation assumes no prior knowledge of the system's dynamics. Alternatively, [12] develops a runtime assurance mechanism for distributed avionics architecture which can

M. E. Cao and S. Coogan are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, 30332, USA {mcao34, sam.coogan}@gatech.edu. S. Coogan is also with the School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, 30332, USA. This work was supported in part by the National Science Foundation under awards #1749357 and #1924978, the Ford Motor Company under the Ford-GT Alliance Program, and the NASA University Leadership Initiative under grant #80NSSC20M0161. This article solely reflects the opinions and conclusions of its authors.

intervene in the event of failure. [13] 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.

Other approaches to the invariant tube synthesis problem (also known as the *funnel synthesis* problem) include [14], which develops a funnel synthesis algorithm for computing controlled invariant sets around a given nominal trajectory by solving a differential LMI, [15] which develops several strategies utilizing Sum-of-Squares programming for computing regions of finite-time invariance around solutions of polynomial differential equations, and [16] which proposes an approach to funnel synthesis that is based on falsification. Additionally, [17] develops a joint trajectory and funnel synthesis technique for discrete-time systems with locally Lipschitz nonlinearities. Finally, [18], [19] develop funnel synthesis strategies applied in real-time on aerial vehicles.

In this work, we leverage the mixed monotonicity property to derive a runtime assurance mechanism that can detect whether the controller is unable to follow the reference trajectory within a desired tolerance. Specifically, we apply the formulation proposed by [20] to calculate a forward-invariant tube around the reference trajectory based on the current observations of the unknown disturbance behavior and the limits of the controller, and we then design a controller which guarantees this forward invariance. 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 that the system never deviates from the reference trajectory by more than the safe threshold. We demonstrate this formulation on a case study of a planar multirotor vehicle making a safe landing in a wind field.

In contrast to the current literature, our proposed formulation natively accommodates nonlinear dynamics without the need to solve for LMIs, which are relatively expensive computationally. Additionally, we solve the problem fully in continuous-time and do not need to choose time discretization points. Finally, we propose a method for sampling to improve knowledge of the unknown dynamics that minimizes the number of samples needed to ensure safety.

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

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 hyperrectangle 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 hyperrectangle formed by the first and last n components of a, i.e., $[\![a]\!] := [x, y]$. Finally, 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 $y \preceq y'$,

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

III. Problem Setup

We consider the continuous-time, nonlinear, Lipschitz-continuous system

$$\dot{x} = f(x, u, w) \tag{2}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in U = [\underline{u}, \overline{u}] \subset \mathbb{R}^m$ is the system input constrained to an interval, and $w \in \mathbb{R}^p$ is an unknown, state-dependent component of the dynamics so that $w_i = g_i(x)$ where g_i is unknown.

We make the following assumptions on (2).

Assumption 1. *The Isaacs minimax condition is satisfied: given any interval sets $U = [\underline{u}, \overline{u}] \subset \mathbb{R}^m$ and $W = [\underline{w}, \overline{w}] \subset \mathbb{R}^p$, for all $q \in \mathbb{R}^n$,*

$$\min_{u \in U} \max_{w \in W} \langle q, f(x, u, w) \rangle = \max_{w \in W} \min_{u \in U} \langle q, f(x, u, w) \rangle \tag{3}$$

This is a mild assumption requiring that, for obtaining an optimal control strategy, it does not matter whether the input is chosen before or after the disturbance is realized at each time instant.

Assumption 2. *For all $(x, w) \in \mathbb{R}^n \times \mathbb{R}^p$, $f(x, U, w)$ is an interval set.*

This assumption is more restrictive, but as we will showcase in Section V, it is sometimes possible to apply a transformation to the state-space dynamics of a system such that the transformed dynamics fulfill this assumption. Additionally, when the assumption does not hold, it is always possible to under-approximate $f(x, U, w)$ with interval sets.

Thus, we define our problem as follows.

Problem 1. *Given a system* (2) *which fulfills Assumptions 1 and 2, as well as reference trajectories $x_r(t), \dot{x}_r(t), u_r(t)$, which solve* (2) *for some $w_r(t)$, devise an assurance mechanism that ensures the system remains within a safe distance ε of the reference trajectory for all time.*

Multiple formulations exist to produce safe trajectories for, *e.g.*, autonomous aerial vehicles, leveraging techniques from optimal control [1], [21], and differential flatness [22], [23], for example. In this work, we are specifically interested in the problem of ensuring close tracking of a given reference trajectory in the presence of unknown disturbance behavior at runtime. Thus, we presume that any of these techniques are readily available for generating the reference trajectory.

IV. High Probability Forward Invariance

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 Gaussian Processes (GPs) leads to overapproximations of reachable sets that hold with high probability. Finally, we modify the formulation to enable the calculation of a forward invariant tube with respect to a reference trajectory in service of solving Problem 1.

A. Mixed Monotonicity

The system (2) is *mixed monotone with respect to a decomposition function* δ if it satisfies the following:

1) For all x, u, w, $\delta(x, u, w, x, u, w) = f(x, u, w)$;
2) For all $\underline{x}, \overline{x}, u, \widehat{u}, w, \widehat{w}$ and all $i, j \in \{1, \cdots, n\}$, $i \neq j$, $\frac{\partial \delta_i}{\partial \underline{x}_j}(\underline{x}, u, w, \overline{x}, \widehat{u}, \widehat{w}) \geq 0$;
3) For all $\underline{x}, \overline{x}, u, \widehat{u}, w, \widehat{w}$ and all $i, j \in \{1, \cdots, n\}$, $\frac{\partial \delta_i}{\partial \overline{x}_j}(\underline{x}, u, w, \overline{x}, \widehat{u}, \widehat{w}) \leq 0$;
4) For all $i \in \{1, \cdots, n\}$ and all $k \in \{1, \cdots, p\}$, $\frac{\partial \delta_i}{\partial w_k}(\underline{x}, u, w, \overline{x}, \widehat{u}, \widehat{w}) \geq 0$ and $\frac{\partial \delta_i}{\partial \widehat{w}_k}(\underline{x}, u, w, \overline{x}, \widehat{u}, \widehat{w}) \leq 0$ for all $\underline{x}, \overline{x}, u, \widehat{u}, w, \widehat{w}$.
5) For all $i \in \{1, \cdots, n\}$ and all $k \in \{1, \cdots, m\}$, $\frac{\partial \delta_i}{\partial u_k}(\underline{x}, u, w, \overline{x}, \widehat{u}, \widehat{w}) \geq 0$ and $\frac{\partial \delta_i}{\partial \widehat{u}_k}(\underline{x}, u, w, \overline{x}, \widehat{u}, \widehat{w}) \leq 0$ for all $\underline{x}, \overline{x}, u, \widehat{u}, w, \widehat{w}$.

For any system, there exists some decomposition function δ satisfying the above conditions [24], although one may not be readily available in closed form. In general, obtaining a decomposition function is problem specific, but automated tools exist for computing certain classes of decomposition functions [4]. We demonstrate construction of a decomposition function in the case study of Section V. For more examples of decomposition functions and practical applications of mixed monotonicity, see [25], [26].

Given mixed monotone system (2) and a corresponding decomposition function, we then construct the *embedding system* with state $(\underline{x}, \overline{x}) \in \mathbb{R}^n \times \mathbb{R}^n$, input $(u, \widehat{u}) \in \mathbb{R}^m \times \mathbb{R}^m$, and disturbance $(w, \widehat{w}) \in \mathbb{R}^p \times \mathbb{R}^p$ defined by the dynamics

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\overline{x}} \end{bmatrix} = \varepsilon(\underline{x}, u, w, \overline{x}, \widehat{u}, \widehat{w}) := \begin{bmatrix} \delta(\underline{x}, u, w, \overline{x}, \widehat{u}, \widehat{w}) \\ \delta(\overline{x}, \widehat{u}, \widehat{w}, \underline{x}, u, w) \end{bmatrix}. \quad (4)$$

Denote the state of (2) at time t when initialized at x_0 under some input signal $u : [0, \infty) \to \mathbb{R}^m$ and some disturbance signal $w : [0, \infty) \to \mathbb{R}^p$ by $\phi(t; x_0, u, w)$, and denote the state of (4) at time t initialized at $(\underline{x}_0, \overline{x}_0)$ under some input signal $(u, \widehat{u}) : [0, \infty) \to \mathbb{R}^m \times \mathbb{R}^m$, and disturbance signal $(w, \widehat{w}) : [0, \infty) \to \mathbb{R}^p \times \mathbb{R}^p$ by $\Phi^\varepsilon(t; (\underline{x}_0, \overline{x}_0), (u, \widehat{u}), (w, \widehat{w}))$. The fundamental result of mixed monotone systems theory is that (4) is a monotone control system as defined in [27] with respect to the southeast order on state, input, and disturbance; that is, given $a, a' \in \mathbb{R}^n \times \mathbb{R}^n$, $b, b' : [0, \infty) \to \mathbb{R}^m \times \mathbb{R}^m$ and $c, c' : [0, \infty) \to \mathbb{R}^p \times \mathbb{R}^p$ such that $a \preceq_{\text{SE}} a'$, $b \preceq_{\text{SE}} b'$, and $c(t) \preceq_{\text{SE}} c'(t)$ for all $t \geq 0$, then for all $t \geq 0$,

$$\Phi^\varepsilon(t; a, b, c) \preceq_{\text{SE}} \Phi^\varepsilon(t; a', b', c'). \quad (5)$$

In other words, provided that the system is initialized within $[\underline{x}_0, \overline{x}_0]$, the input signal is overapproximated by $[u, \widehat{u}]$, and the disturbance signal is overapproximated by $[w, \widehat{w}]$, then the hyperrectangle defined by $[\![\Phi^\varepsilon(t; (\underline{x}_0, \overline{x}_0), (u, \widehat{u}), (w, \widehat{w}))]\!]$ overapproximates the true reachable set of (2), *i.e.*

$$\phi(T; x_0, u, w) \subseteq [\![\Phi^\varepsilon(T; (\underline{x}_0, \overline{x}_0), (u, \widehat{u}), (w, \widehat{w}))]\!] \quad (6)$$

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

B. Gaussian Processes and High Probability Reachable Sets

If there exist known bounding functions $\underline{\gamma}_i(\underline{x}, \overline{x})$ and $\overline{\gamma}_i(\underline{x}, \overline{x})$, $\underline{\gamma}_i, \overline{\gamma}_i : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, for all $i \in \{1, \ldots, p\}$ such that

$$\underline{\gamma}_i(\underline{x}, \overline{x}) \leq g_i(x) \leq \overline{\gamma}_i(\underline{x}, \overline{x}), x \in [\underline{x}, \overline{x}], \quad (7)$$

for all $\underline{x}, \overline{x} \in \mathbb{R}^n$ with $\underline{x} \preceq \overline{x}$, where $g(x)$ represents the a priori unknown function dictating the behavior of w, then these functions can be inserted into the previously described embedding system to produce valid reachable set overapproximations. This is achieved by taking the embedding system (4) and inserting $\underline{\gamma}(\underline{x}, \overline{x}), \overline{\gamma}(\underline{x}, \overline{x})$ in place of $w, \widehat{w}$ to produce a new embedding system as

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\overline{x}} \end{bmatrix} = e(\underline{x}, u, \overline{x}, \widehat{u}) := \begin{bmatrix} \delta(\underline{x}, u, \underline{\gamma}(\underline{x}, \overline{x}), \overline{x}, \widehat{u}, \overline{\gamma}(\underline{x}, \overline{x})) \\ \delta(\overline{x}, \widehat{u}, \overline{\gamma}(\underline{x}, \overline{x}), \underline{x}, u, \underline{\gamma}(\underline{x}, \overline{x})) \end{bmatrix}. \quad (8)$$

We have shown previously in [3] that modeling the unknown functions $g_i(x)$ as GPs enables us to formulate bounding functions that fulfill (7) with probability $1 - \eta, \eta \in (0, 1]$. Given observations $\{y_j\}_{j=1}^t$ of the GP at corresponding points $\{x_j\}_{j=1}^t$, the surrogate functions of interest to approximate g_i are

$$\forall i \in \{1, \cdots, p\}, \quad \begin{cases} \overline{g}_i^{(t)}(x) := \mu_t(x) + \sqrt{\beta_t}\sigma_t(x) \\ \underline{g}_i^{(t)}(x) := \mu_t(x) - \sqrt{\beta_t}\sigma_t(x) \end{cases} \quad (9)$$

where β_t is

$$\beta_t := 2 \log \left(\frac{pt^2\pi^2}{3\eta} \right) + 2n \log \left(t^2 nbr \sqrt{\log \left(\frac{2pna}{\eta} \right)} \right), \quad (10)$$

$\mu_t(\cdot)$ is the posterior mean, and $\sigma_t(\cdot)$ is the posterior variance, computed according to the standard GP updates [28]:

$$\mu_t(x) := k_t(x)^T (K_t + \sigma^2 I)^{-1} y \quad (11)$$

$$k_t(x, x') := k(x, x') - k_t(x)^T (K_t + \sigma^2 I)^{-1} k_t(x') \quad (12)$$

$$\sigma_t^2(x) := k_t(x, x) \quad (13)$$

where $k_t(x) := (k(x_1, x), \cdots, k(x_t, x))$ and $K_t = [k_t(x_i, x_j)]$. Then, for all $i \in \{1, \cdots, p\}$ and all $t \geq 1$, define for all $\underline{x} \preceq \overline{x}$,

$$\underline{\gamma}_i^{(t)}(\underline{x}, \overline{x}) := \min_{x \in [\underline{x}^{(t,-)}, \overline{x}^{(t,+)}] \cap \mathcal{D}_t} \underline{g}_i^{(t)}(x) - \frac{1}{t^2}, \quad (14)$$

$$\overline{\gamma}_i^{(t)}(\underline{x}, \overline{x}) := \max_{x \in [\underline{x}^{(t,-)}, \overline{x}^{(t,+)}] \cap \mathcal{D}_t} \overline{g}_i^{(t)}(x) + \frac{1}{t^2}. \quad (15)$$

These functions fulfill (7) with probability at least $1 - \eta$, and thus the reachable set overapproximations calculated by the embedding system (8) using these functions hold with probability at least $1 - \eta$.

C. Forward Invariant Tube

We now apply the formulation proposed in [20] to generate our forward invariant tube. For any system, a valid decomposition function takes the form

$$\delta_i(\underline{x}, u, w, \overline{x}, \widehat{u}, \widehat{w}) = \min_{a\in[\underline{x},\overline{x}],a_i=\underline{x}_i} \min_{b\in[u,\widehat{u}]} \min_{c\in[w,\widehat{w}]} f_i(a,b,c) \tag{16}$$

$$\delta_i(\overline{x}, \widehat{u}, \widehat{w}, \underline{x}, u, w) = \max_{a\in[\underline{x},\overline{x}],a_i=\underline{x}_i} \max_{b\in[u,\widehat{u}]} \max_{c\in[w,\widehat{w}]} f_i(a,b,c) \tag{17}$$

for all $i \in [1, n]$. However, solving for these values at runtime is generally difficult; thus the main challenge lies in deriving a closed-form version of δ that solves (16)-(17). A function δ that has this property is considered to be *tight*.

For the particular problem setting of tracking a reference trajectory, the control input u and disturbance input w are strictly competitive forces. In other words, any deviation from $w_r(t)$ taken by the disturbance input w must be countered with deviation from $u_r(t)$ by the control input u. To model this competition, we note that a function δ that is tight (i.e. fulfills (16)–(17)) also has the properties

$$\delta_i(\underline{x}, \widehat{u}, w, \overline{x}, u, \widehat{w}) = \min_{a\in[\underline{x},\overline{x}],a_i=\underline{x}_i} \max_{b\in[u,\widehat{u}]} \min_{c\in[w,\widehat{w}]} f_i(a,b,c) \tag{18}$$

$$\delta_i(\overline{x}, u, \widehat{w}, \underline{x}, \widehat{u}, w) = \max_{a\in[\underline{x},\overline{x}],a_i=\underline{x}_i} \min_{b\in[u,\widehat{u}]} \max_{c\in[w,\widehat{w}]} f_i(a,b,c) \tag{19}$$

for all $i \in [1, n]$ via property 5 of the requirements of a valid decomposition function δ.

Since our application is essentially interested in the worst-case scenario, we can use $\underline{u}$ and $\overline{u}$ moving forward, as these represent the full capability of our controller. We also recall that the reference trajectory $x_r(t)$ solves (2) for some $u_r(t)$ and $w_r(t)$; such a trajectory can be attained by defining $w_r(t) = \mu_t(x_r(t))$, which lies within the high-probability bounds attained previously. Finally, as our objective is to track the reference trajectory, we want to know when we are capable of doing so exactly. Thus, as is proposed in [20], we define the embedding system dynamics as

$$\dot{\underline{x}}_i = \begin{cases} d_i(\underline{x}, \overline{x}), & x_i(t) < x_{ri}(t) \\ \min\{d_i(\underline{x}, \overline{x}), \dot{x}_{ri}(t)\}, & x_i(t) \geq x_{ri}(t) \end{cases} \tag{20}$$

$$\dot{\overline{x}}_i = \begin{cases} d_i(\overline{x}, \underline{x}), & x_i(t) > x_{ri}(t) \\ \max\{d_i(\overline{x}, \underline{x}), \dot{x}_{ri}(t)\}, & x_i(t) \leq x_{ri}(t) \end{cases} \tag{21}$$

where

$$d_i(\underline{x}, \overline{x}) = \delta_i(\underline{x}, \overline{u}, \underline{\gamma}(\underline{x}, \overline{x}), \overline{x}, \underline{u}, \overline{\gamma}(\underline{x}, \overline{x})), \tag{22}$$

$$d_i(\overline{x}, \underline{x}) = \delta_i(\overline{x}, \underline{u}, \overline{\gamma}(\underline{x}, \overline{x}), \underline{x}, \overline{u}, \underline{\gamma}(\underline{x}, \overline{x})). \tag{23}$$

The resulting hyperrectangular tube formed by $[\underline{x}(t), \overline{x}(t)]$ is a controlled invariant tube, that is, a tube that the system is able to remain within given the limits of the controller. Again, as we are using bounds on the disturbance that hold with probability at least $1 - \eta$, this property holds with the same probability.

We now characterize a class of safety assurance control policies. At runtime, given current state $x(t)$, any control strategy that satisfies

$$u(x(t)) \in \operatorname*{argmin}_{u\in[\underline{u},\overline{u}]} \max_{w\in[\underline{\gamma},\overline{\gamma}]} \langle p(x(t)), f(x(t), u, w)\rangle \tag{24}$$

where

$$p_i(x(t)) = \begin{cases} 1, & x_i(t) \geq \overline{x}_i(t) \\ -1, & x_i(t) \leq \underline{x}_i(t) \\ 0, & \text{otherwise} \end{cases}$$

guarantees forward invariance of $[\underline{x}(t), \overline{x}(t)]$ with probability at least $1 - \eta$.

Moreover, if at some time it does not hold that

$$\bigwedge_{i=1}^{n} |\underline{x}_i(t) - x_{ri}(t)| \leq \varepsilon \wedge |\overline{x}_i(t) - x_{ri}(t)| \leq \varepsilon \tag{25}$$

for all $t \geq 0$, then an observation of the unknown disturbance behavior $g(x)$ is triggered at the time t_r where

$$t_r = \min_{t\geq 0} |\underline{x}_i(t) - x_{ri}(t)| \geq \varepsilon \vee |\overline{x}_i(t) - x_{ri}(t)| \geq \varepsilon, \tag{26}$$

after which a new reference trajectory is generated and the forward invariant tube is recalculated. The overall runtime assurance mechanism is outlined in Algorithm 1.

Algorithm 1 Runtime Assurance Mechanism

```
Data: Embedding system (20)-(21), bounding functions γ, γ̄, safety threshold ε
t_r ← 0
while In Operation do
    if t ≥ t_r then
        Generate Reference Trajectories x_r, ẋ_r, u_r which solve (2) for w_r ∈ [γ, γ̄], x_r(t) = x(t);
        x, x̄ ← solutions to (20), (21), x(t) = x̄(t) = x(t);
        t_r ← (26);
    Apply u satisfying (24);
```

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

Theorem 1. *Given reference trajectories $x_r(t), \dot{x}_r(t), u_r(t)$ that solve 2 for $w_r(t) = \mu_t(x_r(t))$, applying the runtime assurance mechanism outlined by Algorithm 1 guarantees that no state in the system deviates from its reference trajectory by more than ε 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 $\varepsilon > 0$ and condition (26). Consequently, we can say that $t \leq t_r$ for all $t > 0$ at runtime.

We then note that, per condition (26), it must hold that

$$[\underline{x}(t), \overline{x}(t)] \subseteq [x_r(t) - \varepsilon, x_r(t) + \varepsilon] \forall_{t\leq t_r}. \tag{27}$$

Since $t \leq t_r$ for all $t > 0$ at runtime, then for all time $t > 0$,

$$[\underline{x}(t), \overline{x}(t)] \subseteq [x_r(t) - \varepsilon, x_r(t) + \varepsilon]. \tag{28}$$

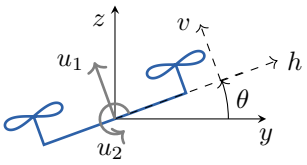

Fig. 1: The planar multirotor model has horizontal position y, vertical position z, and roll angle θ. The inputs are thrust u_1 in the direction perpendicular to the line segment connecting the rotors and roll angle acceleration u_2. The modified five dimensional state x consists of y, z, θ, and the velocity in the horizontal and vertical directions of the frame of the multirotor h and v, so that $x = \begin{bmatrix} y & h & z & v & \theta \end{bmatrix}^T$, which fulfills Assumption 2.

Finally, per [20, Theorem 1], the hyperrectangular tube defined by $[\underline{x}(t), \overline{x}(t)]$ is forward invariant given a control strategy that satisfies (24). ■

Thus, the outlined formulation solves Problem 1.

V. Aerial Vehicle Case Study

We now apply the forward invariant tube and runtime assurance mechanism to an example of a multirotor system that is constrained to moving within the vertical plane. The five-dimensional state x of the planar multirotor system 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.

We assume the system is subject to input constraints $[\underline{u}, \overline{u}] = [(-5, -5), (15, 5)]$ and gravitational acceleration $a_g = 9.81 m/s^2$, as well as an unknown force due to wind. We assume this force affects acceleration in the horizontal direction and is a function of altitude z. The resulting dynamics with normalized mass and moment of inertia are

$$\begin{aligned} \ddot{y} &= -u_1 \sin\theta + g(z) \\ \ddot{z} &= u_1 \cos\theta - a_g \\ \dot{\theta} &= u_2 \end{aligned} \tag{29}$$

where $g(z)$ constitutes the unknown wind force in the horizontal direction.

Generally, a set of system dynamics where the same input appears in multiple state update equations (as in (29)) does not fulfill Assumption 2. Thus, we introduce state variables v and h to denote the vertical and horizontal velocity of the multirotor *in its own frame*. In effect, this is applying a rotation based on θ to the original velocity dynamics $\dot{y}$ and $\dot{z}$. The transformed state-space dynamics take the form

$$\begin{aligned} \dot{y} &= h\cos\theta - v\sin\theta \\ \dot{h} &= -a_g \sin\theta + g(z)\cos\theta \\ \dot{z} &= h\sin\theta + v\cos\theta \\ \dot{v} &= u_1 - a_g\cos\theta - g(z)\sin\theta \\ \dot{\theta} &= u_2 \end{aligned} \tag{30}$$

which now fulfill Assumption 2. These dynamics are illustrated in Figure 1. The resulting decomposition function is

$$\delta(\underline{x}, u, w, \overline{x}, \widehat{u}, \widehat{w}) = \begin{bmatrix} d^y & d^h & d^z & d^v & u_2 \end{bmatrix}^T \tag{31}$$

$$\begin{aligned} d^y &= d^{b_1b_2}\left(\begin{bmatrix} \underline{h} \\ d^{\cos}(\underline{\theta}, \overline{\theta}) \end{bmatrix}, \begin{bmatrix} \overline{h} \\ d^{\cos}(\overline{\theta}, \underline{\theta}) \end{bmatrix} \right) \\ &\quad - d^{b_1b_2}\left(\begin{bmatrix} \overline{v} \\ d^{\sin}(\overline{\theta}, \underline{\theta}) \end{bmatrix}, \begin{bmatrix} \underline{v} \\ d^{\sin}(\underline{\theta}, \overline{\theta}) \end{bmatrix} \right) \\ d^h &= -a_g d^{\sin}(\overline{\theta}, \underline{\theta}) + d^{b_1b_2}\left(\begin{bmatrix} w \\ d^{\cos}(\underline{\theta}, \overline{\theta}) \end{bmatrix}, \begin{bmatrix} \widehat{w} \\ d^{\cos}(\overline{\theta}, \underline{\theta}) \end{bmatrix} \right) \\ d^z &= d^{b_1b_2}\left(\begin{bmatrix} \underline{h} \\ d^{\sin}(\underline{\theta}, \overline{\theta}) \end{bmatrix}, \begin{bmatrix} \overline{h} \\ d^{\sin}(\overline{\theta}, \underline{\theta}) \end{bmatrix} \right) \\ &\quad + d^{b_1b_2}\left(\begin{bmatrix} \underline{v} \\ d^{\cos}(\underline{\theta}, \overline{\theta}) \end{bmatrix}, \begin{bmatrix} \overline{v} \\ d^{\cos}(\overline{\theta}, \underline{\theta}) \end{bmatrix} \right) \\ d^v &= u_1 - a_g d^{\cos}(\overline{\theta}, \underline{\theta}) - d^{b_1b_2}\left(\begin{bmatrix} \widehat{w} \\ d^{\sin}(\overline{\theta}, \underline{\theta}) \end{bmatrix}, \begin{bmatrix} w \\ d^{\sin}(\underline{\theta}, \overline{\theta}) \end{bmatrix} \right) \end{aligned}$$

where, for $b, \widehat{b} \in \mathbb{R}^2$,

$$d^{b_1b_2}(b, \widehat{b}) = \begin{cases} \min\{b_1b_2, \widehat{b}_1b_2, b_1\widehat{b}_2, \widehat{b}_1\widehat{b}_2\}, & \text{if } b \preceq \widehat{b} \\ \max\{b_1b_2, \widehat{b}_1b_2, b_1\widehat{b}_2, \widehat{b}_1\widehat{b}_2\}, & \text{if } \widehat{b} \preceq b, \end{cases}$$

and $d^{\sin}, d^{\cos}$ take the forms

$$d^{\sin}(x, \widehat{x}) := \tag{32}$$
$$\begin{cases} \sin(x), & \text{if } (\cos(x), \cos(\widehat{x})) \succeq 0 \\ & \text{and } |x - \widehat{x}| \le \pi \\ \sin(\widehat{x}), & \text{if } (\cos(x), \cos(\widehat{x})) \preceq 0 \\ & \text{and } |x - \widehat{x}| \le \pi \\ \text{sign}(x - \widehat{x}), & \text{if } |x - \widehat{x}| \ge 2\pi \\ \text{sign}(x - \widehat{x}), & \text{if } \cos(x) \le 0 \le \cos(\widehat{x}) \\ & \text{and } |x - \widehat{x}| \le 2\pi \\ \text{sign}(x - \widehat{x}), & \text{if } \cos(x)\cos(\widehat{x}) \ge 0 \\ & \text{and } \pi \le |x - \widehat{x}| \le 2\pi \\ \min\{\sin(x), \sin(\widehat{x})\}, & \text{if } x \le \widehat{x} \\ & \text{and } \cos(x) \ge 0 \ge \cos(\widehat{x}) \\ & \text{and } |x - \widehat{x}| \le 2\pi \\ \max\{\sin(x), \sin(\widehat{x})\}, & \text{if } x \ge \widehat{x} \\ & \text{and } \cos(x) \ge 0 \ge \cos(\widehat{x}) \\ & \text{and } |x - \widehat{x}| \le 2\pi, \end{cases}$$

$$d^{\cos}(x, \widehat{x}) := d^{\sin}\left(x + \frac{\pi}{2}, \widehat{x} + \frac{\pi}{2}\right). \tag{33}$$

To generate a reference trajectory, we linearize the system (30) around the equilibrium and then employ a Linear-Quadratic Regulator (LQR) feedback controller with parameters $Q = \text{diag}([1000, 500, 20, 500, 1])$ and $R = \text{diag}([20, 20])$ to simulate a trajectory to the origin assuming g behaves according to the current mean of the Gaussian process estimation of the disturbance using a radial basis kernel. The resulting state and input trajectories form the reference trajectories $x_r, \dot{x}_r, u_r$ that we use in calculating the forward invariant tube.

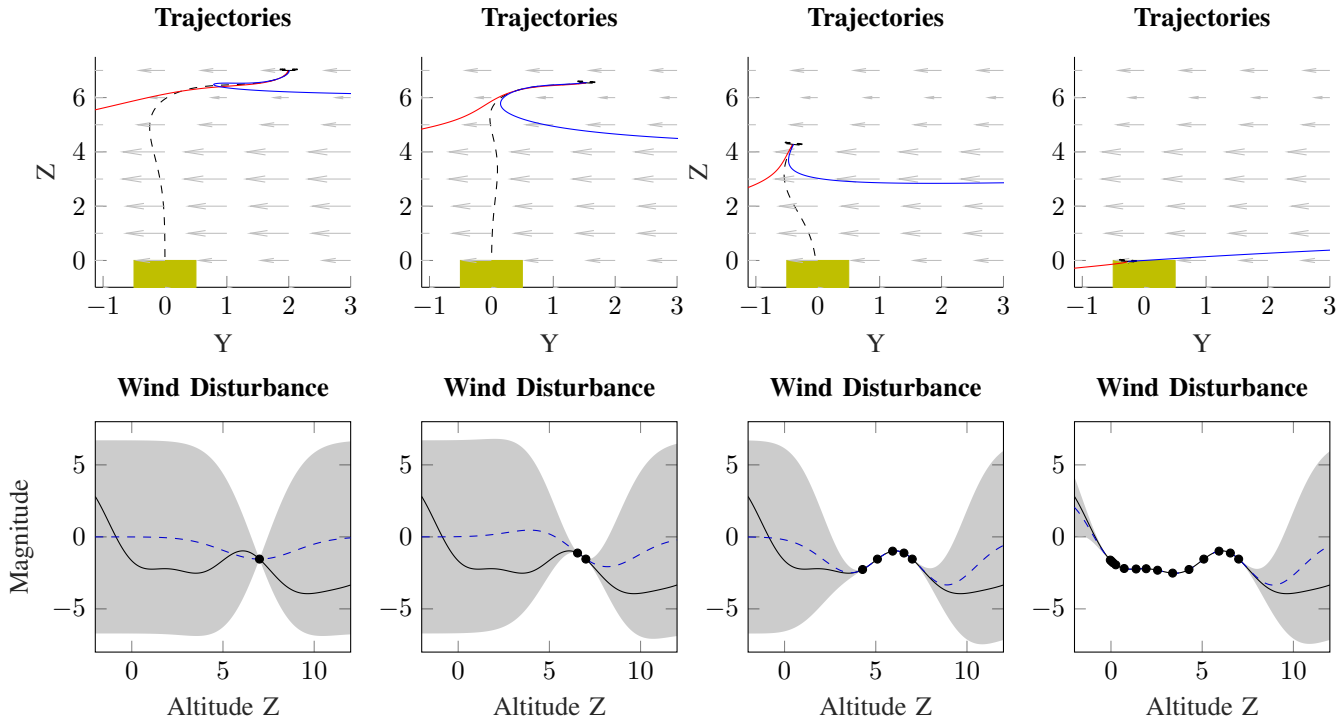

Fig. 2: The planar multirotor landing simulation. The multirotor attempts to make a safe landing by following the calculated reference trajectory (top, black dashed), which is a solution to (2) assuming the wind behaves according to the current estimated mean (bottom, blue dashed) based on the available observations (bottom, points) of the true wind behavior (bottom, black, and top, arrows). Based on these observations and the limits of the propellers, it is possible to calculate a forward invariant tube (top, red and blue) around the reference trajectory. This forward invariant tube assumes the worst-case wind behavior which is calculated by the current confidence bounds (bottom, shaded) on the wind behavior. At the point in the trajectory where the invariant tube deviates from the reference by a certain threshold, a recalculation of the reference trajectory and invariant tube is triggered. As the multirotor descends and collects observations of the wind disturbance behavior, the reference trajectory and invariant tube are continuously updated until the multirotor makes a safe landing.

We then designate a safe landing hyperrectangle $\mathcal{X}_{\text{land}} := [(-0.5, -0.1, -1, -0.1, -\pi/6), (-0.5, 0.1, 0, 0.1, \pi/6)]$ and then calculate the forward invariant tube of the system using the formulation outlined in Section IV and the current estimated bounds on g.

We calculate the applied control action as $u(x(t)) = [u_1(v(t)), u_2(\theta(t))]^T$ where, for $i = \{1, 2\}$,

$$u_i(x(t)) = u_{ri}(t) + \begin{cases} \left(\frac{x(t)-x_r(t)}{\overline{x}(t)-x_r(t)}\right)^3 (u_{ri}(t) - \underline{u}_i), & x(t) \geq x_r(t) \\ \left(\frac{x_r(t)-x(t)}{x_r(t)-\underline{x}(t)}\right)^3 (\overline{u}_i - u_{ri}(t)), & x(t) \leq x_r(t) \end{cases} \tag{34}$$

which fulfills the condition (24). If there does not exist a time t such that $[\underline{x}(t), \overline{x}(t)] \subseteq \mathcal{X}_{\text{land}}$, then we collect a new observation of the disturbance when any edge of the tube deviates from the reference trajectory more than the safety threshold $\varepsilon = 0.3$, and then recalculate the reference trajectory and forward invariant tube. The simulation of the system was executed in Simulink on a personal computer, the results of which are outlined in Figure 2.

As shown, the quadcopter is initially unable to guarantee a safe landing; the forward invariant tube quickly expands past the allowed threshold, as it must account for the uncertainty of the disturbance behavior when the reference trajectory enters regions with few observations of the disturbance. As the quadcopter descends, collecting observations and updating the reference trajectory and forward invariant tube, it is eventually able to achieve a safe landing. The calculation of the reference trajectories and forward invariant tube takes between 0.08–0.2 seconds, and a recalculation is triggered every 0.75–2.0 seconds, showcasing the real-time capabilities of the formulation.

VI. Conclusion

In this work, we have presented a formulation that leverages the mixed monotonicity property of dynamical systems that is able to calculate a forward invariant tube around a desired reference trajectory. We use this formulation to detect when the controller will be unable to adhere to the reference trajectory within a desired threshold, and trigger an observation of the unknown behavior and recalculation of the reference trajectory and forward invariant tube. This formulation guarantees that the system will remain within the desired threshold of safety for all time. Finally, we showcased a case study of a planar multirotor wherein the system is able to achieve a safe landing.

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] M. E. Cao, M. Bloch, and S. Coogan, "Estimating high probability reachable sets using gaussian processes," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 3881–3886, 2021.

[3] 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.

[4] S. Coogan, "Mixed monotonicity for reachability and safety in dynamical systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 5074–5085, 2020.

[5] G. Enciso, H. Smith, and E. Sontag, "Nonmonotone systems decomposable into monotone systems with negative feedback," *Journal of Differential Equations*, vol. 224, no. 1, pp. 205–227, 2006.

[6] M. Abate, M. Mote, E. Feron, and S. Coogan, "Verification and runtime assurance for dynamical systems with uncertainty," in *Hybrid Systems: Computation and Control*, 2021.

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

[8] 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.

[9] I. Lee, S. Kannan, M. Kim, O. Sokolsky, and M. Viswanathan, "Runtime assurance based on formal specifications," *Departmental Papers (CIS)*, p. 294, 1999.

[10] J. D. Schierman, M. D. DeVore, N. D. Richards, and M. A. Clark, "Runtime assurance for autonomous aerospace systems," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 12, pp. 2205–2217, 2020.

[11] Z. Xiong, J. Eappen, A. H. Qureshi, and S. Jagannathan, "Model-free neural lyapunov control for safe robot navigation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5572–5579, 2022.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

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

[19] 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.

[20] V. Sinyakov and A. Girard, "Abstraction of continuous-time systems based on feedback controllers and mixed monotonicity," *IEEE Transactions on Automatic Control*, pp. 1–15, 2022.

[21] 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.

[22] 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.

[23] 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.

[24] M. Abate and S. Coogan, "Computing robustly forward invariant sets for mixed-monotone systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 4553–4559, 2020.

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

[26] 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.

[27] D. Angeli and E. D. Sontag, "Monotone control systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 10, pp. 1684–1698, 2003.

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