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

Online Data-Driven Safety Certification for Systems Subject to Unknown Disturbances

Nicholas Rober*,1, Karan Mahesh*,2, Tyler M. Paine3,4, Max L. Greene2, Steven Lee2, Sildomar T. Monteiro2, Michael R. Benjamin3, Jonathan P. How1

***Abstract*— Deploying autonomous systems in safety critical settings necessitates methods to verify their safety properties. This is challenging because real-world systems may be subject to disturbances that affect their performance, but are unknown *a priori*. This work develops a safety-verification strategy wherein data is collected online and incorporated into a reachability analysis approach to check in real-time that the system avoids dangerous regions of the state space. Specifically, we employ an optimization-based moving horizon estimator (MHE) to characterize the disturbance affecting the system, which is incorporated into an online reachability calculation. Reachable sets are calculated using a computational graph analysis tool to predict the possible future states of the system and verify that they satisfy safety constraints. We include theoretical arguments proving our approach generates reachable sets that bound the future states of the system, as well as numerical results demonstrating how it can be used for safety verification. Finally, we present results from hardware experiments demonstrating our approach's ability to perform online reachability calculations for an unmanned surface vehicle subject to currents and actuator failures.**

I. Introduction

As autonomous systems are more frequently used in safety-critical settings (e.g., autonomous vehicles [1], medical diagnosis [2], and defense systems [3]), there is a growing need to provide statements about the safety of these systems. This is especially important as autonomy pipelines become more complicated, possibly including learned components that are sensitive to distribution shifts [4] or perturbations from nominal conditions [5]. Generating such safety assurances is challenging because applying them to real-world settings often requires dealing with uncertain, and potentially high-dimensional, systems. Moreover, in many settings, it is impossible to predict all environmental conditions or system disturbances *a priori*. In such cases, it is necessary to develop methods capable of certifying safety of a given system at runtime.

* Indicates equal contribution.
[1]Aerospace Controls Laboratory, Massachusetts Institute of Technology, Cambridge, USA. e-mail: {nrober, jhow}@mit.edu.
[2]Aurora Flight Sciences, a Boeing Company, Cambridge, USA. e-mail: {mahesh.karan, greene.max, lee.steven, monteiro.sildomar}@aurora.aero.
[3]Marine Autonomy Laboratory, Massachusetts Institute of Technology, Cambridge, USA. e-mail: {tpaine,mikerb}@mit.edu.
[4]Woods Hole Oceanographic Institution, Woods Hole, MA 02543, USA
This work was supported in part by the US Navy NIWC Atlantic Award N6523623C8011. This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Distribution Statement A. Approved for public release: distribution unlimited.

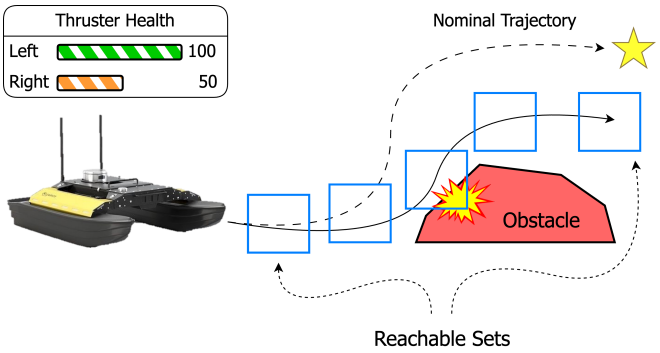

Fig. 1: Reachability analysis detects a possible collision for a system experiencing a hardware malfunction

Safety assurances can generally be obtained using either testing/data collection [6], [7], or formal analysis [8]–[12]. Testing-based approaches rely on collecting large amounts of data through simulation or deployment of a given system and evaluating if/how the system may fail. It is challenging to provide guarantees about safety with testing-based approaches because it is generally impossible to cover all possible failure cases. Alternatively, formal methods can provide guarantees about a given system, but their practical application can be limited due to computational constraints or assumptions about the system that may not reflect its actual behavior. In this paper, we incorporate online data collection into a formal reachability analysis framework to verify the safety of a system, thus striking a balance between data-driven and formal methods.

Reachability analysis, including Hamilton-Jacobi methods [13], [14], Lagrangian methods [15], probabilistic methods [16], and more recent approaches developed for systems with neural networks (NNs) [17]–[25], determines a set of possible future states, i.e., reachable sets (shown in Fig. 1), of a system given a set of possible initial states. While all reachability analysis techniques handle uncertainty in the state (reflected in the initial state set), and many consider process/measurement noise [13], [25], they typically assume knowledge of the system's behavior under bounded uncertainty. Moreover, with only a few exceptions [26]–[28], reachability analysis is often too computationally expensive for online implementation, especially when the system is high-dimensional.

In this work, we perform online reachability analysis of a system subject to disturbances that are unknown *a priori*. Reachable sets are calculated in real time by leveraging `jax_verify` [29], a computational graph analysis tool ca-

pable of efficiently providing output bounds of nonlinear functions. Disturbances acting on the system are estimated online via a moving horizon estimator (MHE) [30], and are incorporated into the reachability analysis to predict the behavior of the actual system.

Our approach is validated using numerical results and deployed on a Clearpath Robotics® Heron unmanned surface vehicle (USV), demonstrating the efficacy of the developed method on a physical system with unknown disturbances. The main contributions of this paper are as follows:

- We developed a data-driven reachability analysis technique for online safety verification of closed-loop systems subject to disturbances and modeling errors
- We prove of the validity of our reachable set over-approximations assuming limits on the rate of change of possible disturbances
- We deploy hardware experiments demonstrating real-time reachability calculations at 10 Hz for a 6DOF USV subject to actuator failures and environmental disturbances, e.g., wind and currents

II. Preliminaries

A. System Dynamics

Consider the nonlinear system

$$\begin{aligned} \mathbf{x}_{t+1} &= f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_t \\ \mathbf{y}_t &= h(\mathbf{x}_t) + \boldsymbol{\nu}_t \end{aligned} \tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^{n_x}$, $\mathbf{u}_t \in \mathbb{R}^{n_u}$, $\mathbf{y}_t \in \mathbb{R}^{n_y}$, $\mathbf{w}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{W}_t)$, $\boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R})$ and $t \in \mathbb{N}$. The dynamics function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and measurement function $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ are assumed to be known, but there is uncertainty in the system due to $\mathbf{w}_t$ because the distribution $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{W}_t)$ is time-varying and unknown, conditioned on $\boldsymbol{W}_t$ being diagonal and

$$\begin{aligned} |(\boldsymbol{\mu}_{t+1})_i - (\boldsymbol{\mu}_t)_i| &\leq \Delta\boldsymbol{\mu}_i \\ |(\boldsymbol{\sigma}_{t+1})_i - (\boldsymbol{\sigma}_t)_i| &\leq \Delta\boldsymbol{\sigma}_i, \end{aligned} \tag{2}$$

where $(\boldsymbol{\sigma}_t)_i \in \mathbb{R}$ for $i = 1, \ldots, n_x$ is the standard deviation associated with the covariance matrix $\boldsymbol{W}_t$, i.e., $(\boldsymbol{W}_t)_{ii} = (\boldsymbol{\sigma}_t)_i^2$, and $\Delta\boldsymbol{\mu}, \Delta\boldsymbol{\sigma} \in \mathbb{R}^{n_x}$ represent the possible variation in $\boldsymbol{\mu}_t$ and $\boldsymbol{\sigma}_t$, respectively, per time step. Note that the unknown nature of $\boldsymbol{\mu}_t$ and $\boldsymbol{W}_t$ (along with the potentially nonlinear dynamics) is a departure from [25], where the support of $\mathbf{w}_t$ is assumed known at each time step. Additionally, while we consider diagonal $\boldsymbol{W}_t$ for brevity of our theoretical arguments here, our approach can be extended to the non-diagonal case in future work. Finally, by introducing a general state-feedback control policy $\mathbf{u}_t = \pi(\mathbf{x}_t)$, the closed-loop dynamics are

$$\mathbf{x}_{t+1} = f_{cl}(\mathbf{x}_t; \pi) + \mathbf{w}_t. \tag{3}$$

B. Reachability Analysis

The forward reachable set at time $t+1$ of a system with closed-loop dynamics (3) is defined recursively as

$$\mathcal{R}_{t+1}(\mathcal{X}_0) = f_{cl}(\mathcal{R}_t(\mathcal{X}_0); \pi), \tag{4}$$

where $\mathcal{R}_0(\mathcal{X}_0) = \mathcal{X}_0$ is the set of possible initial states, and $f_{cl}(\mathcal{R}_t(\mathcal{X}_0); \pi)$ is shorthand for $\{f_{cl}(\mathbf{x}; \pi) \mid \mathbf{x} \in \mathcal{R}_t(\mathcal{X}_0)\}$. Going forward, we will omit the $\mathcal{X}_0$ argument unless it is needed for clarity.

The exact reachable set $\mathcal{R}_t$ is typically expensive to compute, so we instead compute reachable set over-approximations (RSOAs), i.e., $\overline{\mathcal{R}}_t \supseteq \mathcal{R}_t$, as is specified in §III. The RSOAs $\{\overline{\mathcal{R}}_t, \overline{\mathcal{R}}_{t+1}, \ldots, \overline{\mathcal{R}}_{t+\tau_r}\}$, denoted as $\overline{\mathcal{R}}_{t:t+\tau_r}$, can be used to verify safety of the system over a horizon τ_r by checking if the future states can reach an unsafe region of the state space $\mathcal{C} \subset \mathbb{R}^{n_x}$. If there is an $i \in \mathcal{T} = \{t, \ldots, t+\tau_r\}$ such that $\overline{\mathcal{R}}_i \bigcap \mathcal{C} \neq \emptyset$, the system may enter the unsafe region and safety is not guaranteed. Additionally, $\overline{\mathcal{R}}_{t:t+\tau_r}$ can be used to check if the system enters a goal region $\mathcal{G}$. If $\exists i \in \mathcal{T}$ such that $\overline{\mathcal{R}}_i \bigcap \mathcal{G} = \overline{\mathcal{R}}_i$, the system is guaranteed to reach $\mathcal{G} \subset \mathbb{R}^{n_x}$. Note that while RSOAs are capable of verifying safety and liveness as described, more conservative RSOAs make the verification conditions harder to satisfy, so tight RSOAs are preferred.

C. Computational Graphs

A computational graph (CG) $\boldsymbol{G}$ is defined as a directed acyclic graph (DAG) with nodes $\boldsymbol{V} = \{V_1, V_2, \ldots, V_{n_G}\}$ and edges $\boldsymbol{E}$ where each edge is a pair of two nodes (V_i, V_j), indicating that the output of node V_i is an input of node V_j. Each node has an associated computation function $G_i(\cdot)$ consisting of a basic computation such as ReLU or `matmul`. We use the notation $g_i^{\boldsymbol{G}} = G_i(u(V_i))$ where $g_i^{\boldsymbol{G}}$ is the output of V_i and $u(V_i)$ is the set of inputs to V_i, i.e., the outputs from nodes with edges directed toward V_i. The inputs to the graph are denoted as $\mathbf{z} \in \mathbb{R}^{n_i}$. For brevity going forward, we express $g_i^{\boldsymbol{G}}$ in terms of the graph's input, i.e., $g_i^{\boldsymbol{G}} = g_i^{\boldsymbol{G}}(\mathbf{z})$, thus avoiding the explicit use of $u(V_i)$. Without loss of generality, we assume $\boldsymbol{G}$ has a single output node V_o with $\dim(V_o) = \mathbb{R}^{n_o}$, allowing us to express the output of the graph as $g_o^{\boldsymbol{G}}(\mathbf{z}) \in \mathbb{R}^{n_o}$.

D. Computational Graph Relaxation

Computational graph relaxation is used to determine relationships between sets of inputs and outputs of a CG. More specifically, given a set of possible inputs $\mathcal{I}$ to a given CG $\boldsymbol{G}$, the goal is to determine a set of possible outputs $\mathcal{O} = \{g_o^{\boldsymbol{G}}(\mathbf{z}_0) \mid \mathbf{z}_0 \in \mathcal{I}\}$. We construct $\mathcal{I}$ as a hyper-rectangle, defined as

$$\mathcal{B}_\infty(\overline{\mathbf{z}}_0, \boldsymbol{\epsilon}) \triangleq \{\mathbf{z} \mid \|(\mathbf{z} - \overline{\mathbf{z}}_0) \oslash \boldsymbol{\epsilon}\|_\infty \leq 1\}, \tag{5}$$

where $\overline{\mathbf{z}}_0 \in \mathbb{R}^{n_i}$ is the center of the hyper-rectangle, $\boldsymbol{\epsilon} \in \mathbb{R}^{n_i}_{\geq 0}$ is a vector whose elements are the radii for the corresponding elements of $\mathbf{z}$, and $\oslash$ denotes element-wise division.

Theorem 2.1 (Linear Relaxation of CGs [10]): Given a CG $\boldsymbol{G}$ and a hyper-rectangular set of possible inputs $\mathcal{I}$, there exist two explicit functions

$$g^{\boldsymbol{G}}_{L,o}(\mathbf{z}) = \boldsymbol{\Psi}\mathbf{z} + \boldsymbol{\alpha}, \quad g^{\boldsymbol{G}}_{U,o}(\mathbf{z}) = \boldsymbol{\Phi}\mathbf{z} + \boldsymbol{\beta}$$

such that the inequality $g^{\boldsymbol{G}}_{L,o}(\mathbf{z}) \leq g^{\boldsymbol{G}}_{o}(\mathbf{z}) \leq g^{\boldsymbol{G}}_{U,o}(\mathbf{z})$ holds element-wise for all $\mathbf{z} \in \mathcal{I}$, with $\boldsymbol{\Psi}, \boldsymbol{\Phi} \in \mathbb{R}^{n_o \times n_i}$ and $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{n_o}$.

Using Theorem 2.1, a hyper-rectangular over-approximation of the output set $\mathcal{O}$ is constructed as

$$\mathcal{O} \subseteq \overline{\mathcal{O}} = \{\boldsymbol{o} \mid g^{\boldsymbol{G}}_{L,o}(\mathbf{z}) \leq \boldsymbol{o} \leq g^{\boldsymbol{G}}_{U,o}(\mathbf{z}),\ \exists \mathbf{z} \in \mathcal{I}\}.$$

E. Moving-Horizon Estimation

Like [31], [32], we combine set based methods (reachability in our case), with stochastic uncertainty representations. To capture uncertainty, we use an optimization-based estimation strategy called moving horizon estimation [30]. Assuming a system of the form (1) with a prior on the initial state and its covariance, an MHE uses measurement data from a moving window to estimate the current state of the system and its covariance. With a slight modification of the typical forward-time notation [30], the MHE optimization formulation is constructed as

$$\min_{\hat{\mathbf{x}}_{t-\tau_e:t}, \hat{\mathbf{w}}_{t-\tau_e:t}} J, \tag{6}$$

where the cost function J is

$$J = \|\hat{\mathbf{x}}_t - \overline{\mathbf{x}}_t\|^2_{\boldsymbol{Q}^{-1}_{t|t-1}} + \sum_{k=t-\tau_e}^{t} \|\mathbf{y}_k - h(\hat{\mathbf{x}}_k)\|^2_{\boldsymbol{R}^{-1}} + \|\hat{\mathbf{w}}_k\|^2_{\boldsymbol{W}_k^{-1}},$$

where $\overline{\mathbf{x}}_t$ is the state estimate prior, $\boldsymbol{Q}_{t|t-1} \in \mathbb{R}^{n_x \times n_x}$ is the state uncertainty prior, $\mathbf{y}_{t-\tau_e:t}$ represents a set of recent data measurements collected from time $t - \tau_e$ to t, and

$$\|\hat{\mathbf{x}}_t - \overline{\mathbf{x}}_t\|^2_{\boldsymbol{Q}^{-1}_{t|t-1}} \triangleq (\hat{\mathbf{x}}_t - \overline{\mathbf{x}}_t)^\top \boldsymbol{Q}^{-1}_{t|t-1} (\hat{\mathbf{x}}_t - \overline{\mathbf{x}}_t).$$

The result of (6) are values of $\hat{\mathbf{x}}_{t-\tau_e:t}$ and $\hat{\mathbf{w}}_{t-\tau_e:t}$ that optimally estimate the state and disturbance terms over the given window. Much like model predictive control [33], the typical idea behind MHE is to execute (6) and collect $\hat{\mathbf{x}}_t$ at each discrete time step. Each iteration of the process calculates a new estimate of $\hat{\mathbf{x}}_t$ with (6), and the priors of the state estimate $\overline{\mathbf{x}}_{t+1}$ and covariance $\boldsymbol{Q}_{t+1|t}$ are generated for the next time step using the update law

$$\begin{aligned} \overline{\mathbf{x}}_{t+1} &= f_{cl}(\hat{\mathbf{x}}_t; \pi) + \hat{\mathbf{w}}_t \\ \boldsymbol{Q}_{t|t} &= \left(\boldsymbol{Q}^{-1}_{t|t-1} + \boldsymbol{H}^\top \boldsymbol{R}^{-1} \boldsymbol{H}\right)^{-1} \\ \boldsymbol{Q}_{t+1|t} &= \boldsymbol{A}\boldsymbol{Q}_{t|t}\boldsymbol{A}^\top + \boldsymbol{W}_t, \end{aligned} \tag{7}$$

where $\boldsymbol{A} = \frac{\partial f_{cl}}{\partial \mathbf{x}}\big|_{\mathbf{x}=\hat{\mathbf{x}}_t}$ and $\boldsymbol{H} = \frac{\partial h}{\partial \mathbf{x}}\big|_{\mathbf{x}=\hat{\mathbf{x}}_t}$.

III. Reachability for Uncertain Systems

In this section we outline our approach to generating RSOAs for a system of the form (3), subject to *a priori* unknown disturbances. Our approach is designed to be executed online, regularly generating RSOAs over a finite time horizon at a fixed interval. The proposed data-driven reachability approach is summarized as follows:

1) First, before the deployment of the system, we construct f_{cl} as a CG.
2) At each time step during runtime, we execute an MHE iteration to obtain $\hat{\mathbf{x}}_t$ and estimates of the most recent mean and covariance values of $\mathbf{w}_t$.
3) Finally, we feed the mean and covariance estimates from the MHE into a CG relaxation to conduct reachability analysis, thus predicting the behavior of the actual system.

We first address the MHE component. From the MHE formulated in §II-E, we obtain $\hat{\mathbf{x}}_{t-\tau_e:t}$ and $\hat{\mathbf{w}}_{t-\tau_e:t}$ from (6). As described in §II-E, we collect the state estimate $\hat{\mathbf{x}}_t$ and generate the prior $\overline{\mathbf{x}}_{t+1}$, but we also use the rest of the information obtained from $\hat{\mathbf{w}}_{t-\tau_e:t}$ to calculate estimates of $\boldsymbol{\mu}_t$ and $\boldsymbol{W}_t$. Specifically, we make the approximations $\hat{\boldsymbol{\mu}}_t = \texttt{mean}(\hat{\mathbf{w}}_{t-\tau_e:t})$ and $\hat{\boldsymbol{W}}_t = \texttt{cov}(\hat{\mathbf{w}}_{t-\tau_e:t})$.

To conduct reachability analysis, we generate RSOAs using the CG analysis tool `jax_verify` [29]. Note that other CG analysis tools, such as Auto-LiRPA [10] could also be used, but `jax_verify` is fastest and thus enables online computation. As shown in Fig. 2, by specifying the control policy π and the open-loop dynamics (1) as functions in the `jax_verify` framework, we can generate a CG representation of (3) and use Theorem 2.1 to obtain bounds on $\hat{\mathbf{x}}_{t+1}$ from a set of possible $\hat{\mathbf{x}}_t$. Moreover, to account for the range of possible future disturbances given (2), we must also include terms for the disturbance and its rate of change as inputs to the CG. Thus, we introduce the CG $\boldsymbol{G_{cl}}$ with input $\mathbf{z}_t = [\tilde{\mathbf{x}}_t^\top, \tilde{\boldsymbol{\mu}}_t^\top, \mathring{\boldsymbol{\mu}}_t^\top, \mathring{\boldsymbol{\sigma}}_t^\top]^\top$ and output denoted $g^{\boldsymbol{G_{cl}}}_o(\mathbf{z}_t) = [\tilde{\mathbf{x}}_{t+1}^\top, \tilde{\boldsymbol{\mu}}_{t+1}^\top, \mathring{\boldsymbol{\mu}}_{t+1}^\top, \mathring{\boldsymbol{\sigma}}_{t+1}^\top]^\top$ where $\tilde{\mathbf{x}}_t^\top, \tilde{\boldsymbol{\mu}}_t^\top, \mathring{\boldsymbol{\mu}}_t, \mathring{\boldsymbol{\sigma}}_t \in \mathbb{R}^{n_x}$. Note that while the inputs $\tilde{\mathbf{x}}_t$ and $\tilde{\boldsymbol{\mu}}_t$ correspond to $\hat{\mathbf{x}}_t$ and $\hat{\mathbf{w}}_t$, the inputs $\mathring{\boldsymbol{\mu}}_t$ and $\mathring{\boldsymbol{\sigma}}_t$ are internal to the reachability analysis and are used to account for the time variation of $\boldsymbol{\mu}_t$ and $\boldsymbol{W}_t$ as described by (2). $\boldsymbol{G_{cl}}$ is then manually constructed with the equations

$$\tilde{\mathbf{x}}_{t+1} = f_{cl}(\tilde{\mathbf{x}}_t; \pi) + \tilde{\boldsymbol{\mu}}_t \tag{8}$$

$$\tilde{\boldsymbol{\mu}}_{t+1} = \tilde{\boldsymbol{\mu}}_t + \mathring{\boldsymbol{\mu}}_t + \gamma \mathring{\boldsymbol{\sigma}}_t \tag{9}$$

$$\mathring{\boldsymbol{\mu}}_{t+1} = \mathring{\boldsymbol{\mu}}_t \tag{10}$$

$$\mathring{\boldsymbol{\sigma}}_{t+1} = \mathring{\boldsymbol{\sigma}}_t, \tag{11}$$

where $\gamma > 0$ is a parameter used to *concretize* an uncertainty bound for a selected confidence interval, e.g., $\gamma = 3$ means we assume all samples fall within three standard deviations of the mean. Concretization is done because `jax_verify` (and many other analysis tools, such as [10]) assumes concrete bounds on the possible input states.

Having constructed $\boldsymbol{G_{cl}}$ and established how we use the MHE, we can now explicitly specify our approach, which

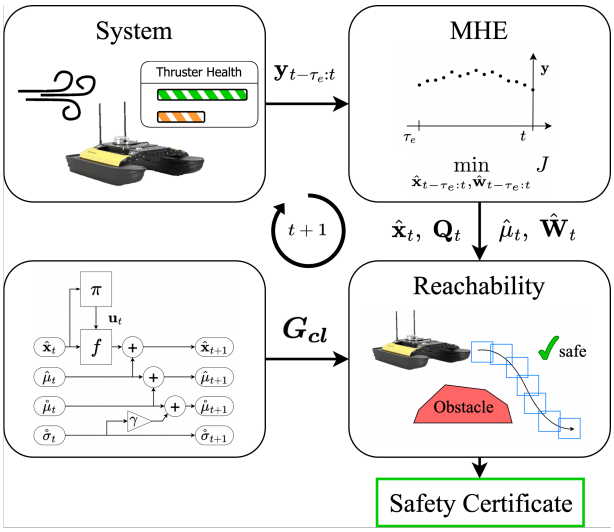

Fig. 2: Block diagram depicting our approach. Data is collected from the system and fed into the MHE, which estimates the state and disturbance terms. The outputs of the MHE are used with a CG representation of the closed-loop dynamics to conduct reachability analysis and certify safety.

is summarized pictorially in Fig. 2 and via pseudo-code in Algorithm 1. At each time step, $\mathbf{y}_{t-\tau_e:t}$ is collected from the boat and passed to the MHE, which determines optimal values for $\hat{\mathbf{x}}_{t-\tau_e:t}$ and $\hat{\mathbf{w}}_{t-\tau_e:t}$. Using the output from the MHE, we determine estimates for the state $\hat{\mathbf{x}}_t$ and its covariance $\boldsymbol{Q}_{t|t-1}$, as well as disturbance parameter estimates $\hat{\boldsymbol{\mu}}_t$ and $\hat{\boldsymbol{W}}_t$. On Lines 4 to 6, we then obtain concrete uncertainty bounds $\boldsymbol{\epsilon}$ by truncating the normal distribution according to concretization parameter γ as previously described. Next we construct the set of possible inputs for $\boldsymbol{G_{cl}}$. On Lines 6 to 8, the possible states $\mathcal{B}(\hat{\mathbf{x}}_t, \boldsymbol{\epsilon_x})$, noise values $\mathcal{B}(\hat{\boldsymbol{\mu}}_t, \boldsymbol{\epsilon_\mu})$, and reachability variables $\mathcal{B}(\mathbf{0}_6, [\Delta\boldsymbol{\mu}^\top, \Delta\boldsymbol{\sigma}^\top]^\top)$, are concatenated to get the initial set $\overline{\mathcal{R}}_t'$ necessary for reachability analysis. Next, on Lines 9 and 10 we loop over the horizon τ_r, calculating RSOAs for each time step. Notice that the RSOA is the set of possible states *and* disturbance terms, $\overline{\mathcal{R}}_t' \subset \mathbb{R}^{4n_x}$. Thus, on Line 11, we project $\overline{\mathcal{R}}_t'$ onto $\mathbb{R}^{n_x}$, thereby enabling us to check the safety condition described in §II-B on Line 13, which is the desired result. Theorem 3.1 formally states the result of our approach.

Theorem 3.1 (Safety Verification for an Uncertain System): Consider a system (3) subject to a disturbance $\mathbf{w}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{W}_t)$ truncated at γ standard deviations, where $\boldsymbol{\mu}_t$ and $\boldsymbol{W}_t$ satisfy the assumptions specified by (2) and where $\hat{\boldsymbol{\mu}}_t$ and $\hat{\boldsymbol{W}}_t$ are accurate estimates for their respective parameters at the current time step. The iterative application of Theorem 2.1 with $\boldsymbol{G_{cl}}$ defined by Eqs. (8) to (11) and where $\mathcal{I} = \mathcal{B}_\infty(\mathbf{z}_t, \boldsymbol{\epsilon})$, $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon_x}^\top, \boldsymbol{\epsilon_\mu}^\top, \Delta\boldsymbol{\mu}^\top, \gamma\Delta\boldsymbol{\sigma}^\top]^\top$ and $\mathbf{z}_t = [\hat{\mathbf{x}}_t^\top, \hat{\boldsymbol{\mu}}_t^\top, \mathbf{0}_3^\top, \mathbf{0}_3^\top]^\top$ provides bounds on all possible $\hat{\mathbf{x}}_{t:t+\tau_r}$, i.e., $\overline{\mathcal{R}}_{t:t+\tau_r}$.

Proof: First, consider the RSOA calculation for time $t+1$. Since the value of $\mathbf{w}_t$ is bound by a distribution with mean $\hat{\boldsymbol{\mu}}_t$ and truncated at γ standard deviations with

Algorithm 1 Online Safety Certification

Input: computational graph $\boldsymbol{G_{cl}}$, reachability horizon τ_r, vehicle data $\mathbf{y}_{t-\tau_e:t}$, concretization parameter γ, unsafe region $\mathcal{C}$

Output: safety certificate c over horizon τ_r

1: $c \leftarrow$ true
2: $\hat{\mathbf{x}}_{t-\tau_e:t},\ \hat{\mathbf{w}}_{t-\tau_e:t} \leftarrow \text{MHE}(\mathbf{y}_{t-\tau_e:t})$
3: $\boldsymbol{\mu}_t,\ \boldsymbol{W}_t \leftarrow \texttt{mean}(\hat{\mathbf{w}}_{t-\tau_e:t}),\ \texttt{cov}(\hat{\mathbf{w}}_{t-\tau_e:t})$
4: $\boldsymbol{\epsilon_x} \leftarrow \texttt{concretize}(\text{MHE}.\boldsymbol{Q}_t, \gamma)$
5: $\boldsymbol{\epsilon_\mu} \leftarrow \texttt{concretize}(\boldsymbol{W}_t, \gamma)$
6: $\boldsymbol{\epsilon} \leftarrow [\boldsymbol{\epsilon_x}^\top, \boldsymbol{\epsilon_\mu}^\top, \Delta\boldsymbol{\mu}^\top, \gamma\Delta\boldsymbol{\sigma}^\top]^\top$
7: $\mathbf{z}_t \leftarrow [\hat{\mathbf{x}}_t^\top, \hat{\boldsymbol{\mu}}_t^\top, \mathbf{0}_3^\top, \mathbf{0}_3^\top]^\top$
8: $\overline{\mathcal{R}}_t' \leftarrow \mathcal{B}(\mathbf{z}_t, \boldsymbol{\epsilon})$
9: **for** i in $\{t+1,\ \ldots t+\tau_r\}$ **do**
10: $\quad\overline{\mathcal{R}}_i' \leftarrow \texttt{jax\_verify}(\boldsymbol{G_{cl}},\ \overline{\mathcal{R}}_{i-1}')$
11: $\quad\overline{\mathcal{R}}_i \leftarrow \texttt{projection}(\overline{\mathcal{R}}_i')$
12: $\quad$**if** $\overline{\mathcal{R}}_i \bigcap \mathcal{C} \neq \emptyset$ **then**
13: $\quad\quad c \leftarrow$ false
14: $\quad$**end if**
15: **end for**
16: **return** c

covariance $\hat{\boldsymbol{W}}_t$, (8) describes all possible values for $\hat{\mathbf{x}}_{t+1}$ when $\tilde{\mathbf{x}}_t \in \mathcal{B}(\hat{\mathbf{x}}_t, \boldsymbol{\epsilon_x})$ and $\tilde{\boldsymbol{\mu}}_t \in \mathcal{B}(\hat{\boldsymbol{\mu}}_t, \boldsymbol{\epsilon_\mu})$. Moreover, since $\hat{\boldsymbol{\mu}}_{t+1} \in \mathcal{B}(\hat{\boldsymbol{\mu}}_t, \Delta\boldsymbol{\mu} + \gamma\Delta\boldsymbol{\sigma})$ (via (2)), (9) captures all values of $\hat{\boldsymbol{\mu}}_{t+1}$ when $\mathring{\boldsymbol{\mu}}_t \in \mathcal{B}(\mathbf{0}_3, \Delta\boldsymbol{\mu})$ and $\mathring{\boldsymbol{\sigma}}_t \in \mathcal{B}(\mathbf{0}_3, \Delta\boldsymbol{\sigma})$. Thus, the application of Theorem 2.1 with $\boldsymbol{G_{cl}}$ and $\mathcal{I}$ provides hyper-rectangular bounds on $\hat{\mathbf{x}}_{t+1}$ and $\hat{\boldsymbol{\mu}}_{t+1}$, i.e., $\overline{\mathcal{R}}_{t+1}' = \{\boldsymbol{o} \mid g_{L,o}^{\boldsymbol{G_{cl}}}(\mathbf{z}_t) \leq \boldsymbol{o} \leq g_{U,o}^{\boldsymbol{G_{cl}}}(\mathbf{z}_t),\ \exists \mathbf{z}_t \in \mathcal{I}\}$. Let $\overline{\mathcal{R}}_{t+1} = \{\mathbf{x} \mid \mathbf{x} = proj_{\mathbb{R}^{n_x}}\boldsymbol{o},\ \forall \boldsymbol{o} \in \overline{\mathcal{R}}_{t+1}'\}$ (note that because $\overline{\mathcal{R}}_{t+1}$ is hyper-rectangular, to project onto $\mathbb{R}^{n_x}$, we simply select the the first n_x components of $\overline{\mathcal{R}}_{t+1}$). Since $\overline{\mathcal{R}}_{t+1}$ contains the projection of elements of $\overline{\mathcal{R}}_{t+1}'$ onto the state-space $\mathbb{R}^{n_x}$, and $\overline{\mathcal{R}}_{t+1}'$ provides bounds on $\hat{\mathbf{x}}_{t+1}$ and $\hat{\boldsymbol{\mu}}_{t+1}$, $\overline{\mathcal{R}}_{t+1}$ is a RSOA for all possible $\hat{\mathbf{x}}_{t+1}$.

To extend this argument to time steps beyond $t+1$, recognize that $\overline{\mathcal{R}}_{t+1}'$ gives an *over*-approximation of both $\hat{\mathbf{x}}_{t+1}$ and $\hat{\boldsymbol{\mu}}_{t+1}$, allowing us to apply the same argument by considering a new initial state set $\mathcal{I}_1 = \overline{\mathcal{R}}_{t+1}'$. Thus, by taking over-approximations of over-approximations, we can construct RSOAs over an arbitrary horizon τ_r. ■

IV. Results

In this section we demonstrate the performance of our approach with results from both numerical and hardware experiments. For each set of experiments, we consider a differential-thrust USV whose body-frame velocity vector $\boldsymbol{\eta_v} = [u, v, r]^\top \in \mathbb{R}^3$ characterizes the vehicle's surge u, sway v, and yaw rate r. We use the widely accepted model of marine vehicle motion [34]:

$$\boldsymbol{M}\dot{\boldsymbol{\eta}}_{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{M}, \boldsymbol{\eta_v})\boldsymbol{\eta_v} + \boldsymbol{D}(\boldsymbol{\eta_v})\boldsymbol{\eta_v} = \boldsymbol{\tau} \tag{12}$$

where $\boldsymbol{M} \in \mathbb{R}^{3\times3}$ is the inertia matrix which includes the rigid-body and added mass terms, $\boldsymbol{C}(\boldsymbol{M}, \boldsymbol{\eta_v}) \in \mathbb{R}^{3\times3}$ is the Coriolis matrix, $\boldsymbol{D}(\boldsymbol{\eta_v}) \in \mathbb{R}^{3\times3}$ is the drag matrix, and

$\boldsymbol{\tau} \in \mathbb{R}^3$ are the forces and moments acting on the vehicle due to the control inputs. The position and orientation of the vehicle, $\boldsymbol{\eta_x} = [x, y, \psi]^\top$, where $x \in \mathbb{R}$ is the x-position, $y \in \mathbb{R}$ is the y-position, and $\psi \in \mathbb{S}^1$ is the heading angle, are subject to the dynamics

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{x}} = \begin{bmatrix} u\cos(\psi) - v\sin(\psi) \\ u\sin(\psi) + v\cos(\psi) \\ r \end{bmatrix}. \tag{13}$$

For the purpose of control design and application of our approach, we make the following discrete time approximation of the update law for the full USV state $\mathbf{x} = [\boldsymbol{\eta}_{\boldsymbol{x}}^\top, \boldsymbol{\eta}_{\boldsymbol{v}}^\top]^\top$:

$$\mathbf{x}_{t+1} = \begin{bmatrix} f_{x,d}(\mathbf{x}_t) \\ \boldsymbol{A}_{cl}\boldsymbol{\eta}_{\boldsymbol{v},t} + \boldsymbol{B}_{cl}\boldsymbol{\eta}_{des,t} \end{bmatrix}, \tag{14}$$

where $f_{x,d} : \mathbb{R}^6 \to \mathbb{R}^3$ is a discrete approximation of (13), $\boldsymbol{A}_{cl} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{B}_{cl} \in \mathbb{R}^{3\times2}$ characterize the closed-loop dynamics obtained via controller design (discussed further in §IV-A and §IV-B) and system ID techniques [35], and $\boldsymbol{\eta}_{des,t} = [u_{des,t}, r_{des,t}]^\top \in \mathbb{R}^2$ contains the desired surge $u_{des,t}$ and yaw rate $r_{des,t}$ that the closed-loop system should track. Note that $\boldsymbol{\eta}_{des,t}$ is the output of a waypoint-following algorithm, which was not compatible with the CG framework required by `jax_verify` due to the need to switch between waypoints. Therefore, to predict future values of $\boldsymbol{\eta}_{des,t}$, we simulate the system forward from the nominal state and collect the desired surges and yaw rates, which are passed to the reachability calculation.

A. Numerical Experiments

First, we use a simulated environment to evaluate the efficacy of our approach while maintaining control over the disturbances that influence the system. We employ a model reference adaptive controller (MRAC) [36], which provides a control signal to (12) such that the closed-loop system behaves according to a user-selected $\boldsymbol{A}_{cl}$ and $\boldsymbol{B}_{cl}$ by design. The details on the implementation of the closed-loop system and MRAC controller can found in the linked repository[1].

1) Symmetric Thruster Failure: Fig. 3 shows an experiment where the USV is commanded to follow the trackline between two waypoints, shown as a dotted black line. At $t = 0$, the USV is behaving as expected, travelling at $0.5\,\mathrm{m/s}$ to the right along the trackline. Simulated trajectories (orange) are propagated forward from the initial state set (black), and are contained within the RSOAs (blue) with $\gamma = 3$ over a time horizon of $2.5\,\mathrm{s}$. The RSOAs do not intersect with the unsafe regions (red) $0.5\,\mathrm{m}$ away from the trackline, so the USV is guaranteed to be safe over the time horizon. At $t = 4$, a disturbance is introduced, causing both the USV's left and right thrusters to operate at 50% effectiveness. Fig. 4 shows that until $t = 4$, the estimated disturbance in the u dynamics was near 0, but the MHE registers the loss of control effectiveness as a disturbance, which is captured by the estimates $(\hat{\boldsymbol{\mu}}_t)_1$ (line) and $(\hat{\boldsymbol{\sigma}}_t)_1$ (shaded region). Notice

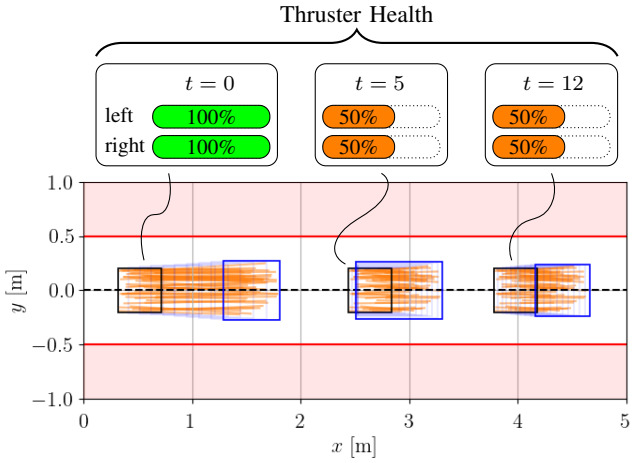

Fig. 3: Snapshots of reachable sets for a USV model over the course of a waypoint-tracking mission. Possible trajectories (orange) are sampled from the initial state set (black) and bounded by the RSOAs (blue). The RSOAs never intersect with the unsafe region (red), so the USV is guaranteed to be safe, but a 50% decrease in thruster effectiveness causes the USV to slow down. The RSOAs accurately capture the behavior of the system despite the thruster malfunction.

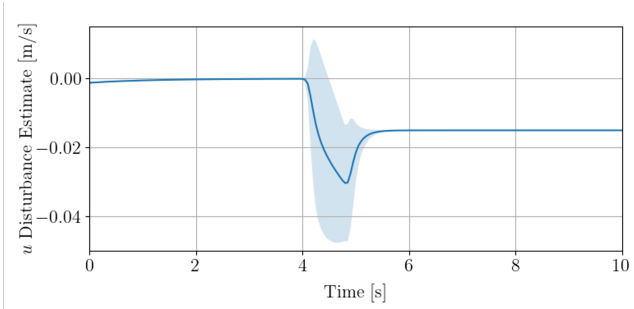

Fig. 4: Estimates $(\hat{\boldsymbol{\mu}}_t)_1$ (line) and $(\hat{\boldsymbol{\sigma}}_t)_1$ (shaded) for the experiment shown in Fig. 3. The MHE estimates the bias and covariance to characterize the disturbance (thruster malfunction) affecting the USV.

that as the system experiences a new disturbance, the RSOAs inflate in response to the increased uncertainty captured by $\hat{\boldsymbol{\sigma}}_t$. This phenomenon is reflected in the RSOAs calculated at $t = 5$, which are stretched horizontally. At $t = 12$, the disturbance estimate has stabilized and the reachable sets accurately capture the sampled trajectories under the new operating conditions.

2) Asymmetric Thruster Failure: Fig. 5 shows a scenario similar to the one discussed in §IV-A.1: the USV is commanded to follow the dotted trackline at $0.5\,\mathrm{m/s}$. In this experiment however, only the USV's right thruster experiences a malfunction. The asymmetry in the system's control effectiveness is too much for the controller to handle, causing the vehicle to steer to the right at $t = 5$. The RSOAs accurately reflect the bias in the yaw rate, correctly predicting that the USV is going to veer off course. Because the RSOAs calculated at $t = 5$ intersect with the unsafe region, safety cannot be guaranteed over the $2.5\,\mathrm{s}$ time horizon.

[1] https://github.com/mit-acl/online_mhe_reachability

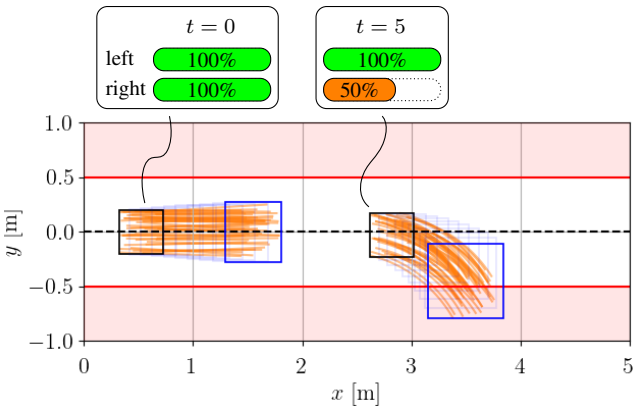

Fig. 5: Snapshots of reachable sets over the course of a waypoint-tracking mission where the USV's right thruster malfunctions. The disturbance from the thruster malfunction is incorporated into the RSOAs, allowing the analysis to predict a possible collision with the unsafe region.

Fig. 6: Clearpath Robotics Heron® USV at the MIT Sailing Pavilion.

B. Hardware Experiments

To test the ability of our approach to handle real-world disturbances, we deployed Algorithm 1 using a Clearpath Robotics® Heron USV shown in Fig. 6. The Heron USV is a $1.35m \times 0.98m \times 0.32m$ catamaran-style vehicle with parallel differential thrusters [37]. The Heron USV's sensor package includes a compass magnetometer, IMU, GPS module and antenna, WiFi antenna, and RF antenna. The Heron USV has two onboard computers for managing high-frequency operations (e.g., motor control) and lower frequency operations (e.g., waypoint following). These computers communicate with a command station equipped with an Intel® Core™ i7 CPU running at 2.7 GHz. While the command station is used to run the online computation of our approach in the following experiment, future work will move the calculation onboard the vessel. For the hardware experiment, we used system identification to estimate a linear set of matrices that represent (12) and designed a robust servomechanism LQR (RSLQR) as specified in [38].

Fig. 7 shows the results of the hardware experiment. Similarly to the experiment described in §IV-A.1, the Heron USV is commanded to follow a trackline between two waypoints. Initially, the USV behaves as expected: about 40% of its maximum power is applied to each thruster to travel at $1\,\mathrm{m/s}$. As the USV performs the mission, we induce a malfunction in both thrusters, reducing their effectiveness to 30% of the commanded thruster output. The controller attempts to overcome the reduced thruster effectiveness by increasing the commanded power to 100% ; however, given the degraded thruster health, the Heron USV can only maintain a forward velocity of $0.7\,\mathrm{m/s}$. The effect of this disturbance appears in the RSOAs shortly after it is applied: the RSOAs are compressed toward the vehicle, thus reflecting the slower forward velocity. At each instance in Fig. 7, we calculate 20 RSOAs (cyan) with $\gamma = 3$ over a time horizon of $\tau_r = 8\,\mathrm{s}$ with an average total computation time of $1.03 \pm 0.67\,\mathrm{ms}$.

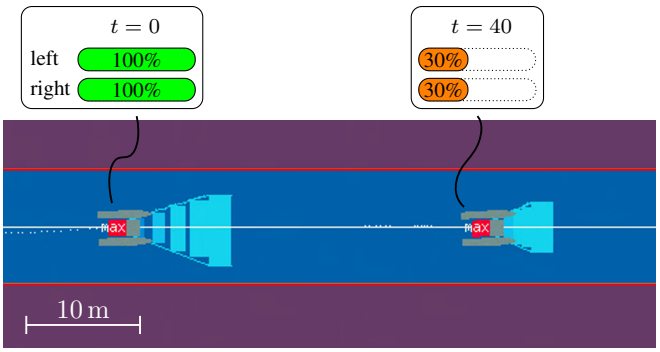

Fig. 7: Hardware experiment with symmetric thruster malfunctions. RSOAs (cyan) are smaller after the disturbance is introduced, correctly predicting the lower speed of the USV. Because the RSOAs do not intersect with the unsafe region (purple), safety is guaranteed.

V. Conclusion

This paper introduces an online safety certification method that leverages moving horizon estimation (MHE) and forward reachability analysis to predict the future states of a system despite unknown disturbances. Reachability analysis is often computationally expensive and assumes knowledge of the system's true dynamics; these trade-offs prohibit such methods from being used on real robotics systems. This method employs a computational graph analysis tool to construct linear bounds on a computational graph representation of the system's estimated dynamics, which includes the nominal system dynamics and a bias estimate from MHE.

Future work includes extending this method with online system learning techniques to improve the bias estimation throughout the reachability horizon, allowing us to handle more complex, time-varying disturbances in a less conservative way. Additionally, exploring more rigorous methods of including multi-loop control architectures, e.g., waypoint following or including the reachable sets in control decisions, would enable more complex reachable set behaviors and improve this method's safety verification accuracy.

REFERENCES

[1] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2722–2730.

[2] M. Bakator and D. Radosav, "Deep learning and medical diagnosis: A review of literature," *Multimodal Technologies and Interaction*, vol. 2, no. 3, p. 47, 2018.

[3] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao, "Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning," *IEEE Robotics and automation letters*, vol. 6, no. 2, pp. 1020–1027, 2021.

[4] E. H. Pooch, P. Ballester, and R. C. Barros, "Can we trust deep learning based diagnosis? the impact of domain shift in chest radiograph classification," in *Thoracic Image Analysis: Second International Workshop, TIA 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings 2*. Springer, 2020, pp. 74–83.

[5] A. Kurakin, I. Goodfellow, S. Bengio *et al.*, "Adversarial examples in the physical world," 2016.

[6] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[7] W. Huang, K. Wang, Y. Lv, and F. Zhu, "Autonomous vehicles testing methods review," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 163–168.

[8] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," *Advances in neural information processing systems*, vol. 31, 2018.

[9] L. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon, "Towards fast computation of certified robustness for relu networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5276–5285.

[10] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kailkhura, X. Lin, and C.-J. Hsieh, "Automatic perturbation analysis for scalable certified robustness and beyond," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1129–1141, 2020.

[11] V. Tjeng, K. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," *arXiv preprint arXiv:1711.07356*, 2017.

[12] G. Katz, D. A. Huang, D. Ibeling, K. Julian, C. Lazarus, R. Lim, P. Shah, S. Thakoor, H. Wu, A. Zeljić *et al.*, "The marabou framework for verification and analysis of deep neural networks," in *International Conference on Computer Aided Verification*. Springer, 2019, pp. 443–452.

[13] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2242–2253.

[14] L. C. Evans, "Graduate studies in mathematics," 1998.

[15] J. D. Gleason, A. P. Vinod, and M. M. Oishi, "Underapproximation of reach-avoid sets for discrete-time stochastic systems via lagrangian methods." IEEE, 2017, pp. 4283–4290.

[16] J. A. dit Sandretto, "Confidence-based contractor, propagation and potential clouds for differential equations," *Acta Cybernetica*, vol. 25, no. 1, pp. 49–68, 2021.

[17] J. A. Vincent and M. Schwager, "Reachable polyhedral marching (rpm): A safety verification algorithm for robotic systems with deep neural network components," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9029–9035.

[18] S. Dutta, X. Chen, and S. Sankaranarayanan, "Reachability analysis for neural feedback systems using regressive polynomial rule inference," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019, pp. 157–168.

[19] C. Huang, J. Fan, W. Li, X. Chen, and Q. Zhu, "Reachnn: Reachability analysis of neural-network controlled systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 5s, pp. 1–22, 2019.

[20] R. Ivanov, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Verisig: verifying safety properties of hybrid systems with neural network controllers," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019, pp. 169–178.

[21] J. Fan, C. Huang, X. Chen, W. Li, and Q. Zhu, "Reachnn*: A tool for reachability analysis of neural-network controlled systems," in *International Symposium on Automated Technology for Verification and Analysis*. Springer, 2020, pp. 537–542.

[22] W. Xiang, H.-D. Tran, X. Yang, and T. T. Johnson, "Reachable set estimation for neural network control systems: A simulation-guided approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 1821–1830, 2020.

[23] H. Hu, M. Fazlyab, M. Morari, and G. J. Pappas, "Reach-sdp: Reachability analysis of closed-loop systems with neural network controllers via semidefinite programming," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 5929–5934.

[24] C. Sidrane, A. Maleki, A. Irfan, and M. J. Kochenderfer, "Overt: An algorithm for safety verification of neural network control policies for nonlinear systems," *arXiv preprint arXiv:2108.01220*, 2021.

[25] M. Everett, G. Habibi, C. Sun, and J. P. How, "Reachability analysis of neural feedback loops," *IEEE Access*, vol. 9, pp. 163 938–163 953, 2021.

[26] T. Lew and M. Pavone, "Sampling-based reachability analysis: A random set theory approach with adversarial sampling," in *Conference on robot learning*. PMLR, 2021, pp. 2055–2070.

[27] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[28] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Fastrack: A modular framework for fast and guaranteed safe motion planning," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1517–1522.

[29] Deepmind, "jax_verify: Neural network verification in jax," 2020. [Online]. Available: https://github.com/deepmind/jax_verify

[30] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright, "Nonlinear predictive control and moving horizon estimation—an introductory overview," *Advances in control: Highlights of ECC'99*, pp. 391–449, 1999.

[31] A. Rauh, S. Wirtensohn, P. Hoher, J. Reuter, and L. Jaulin, "Reliability assessment of an unscented kalman filter by using ellipsoidal enclosure techniques," *Mathematics*, vol. 10, no. 16, p. 3011, 2022.

[32] T. A. Tran, C. Jauberthie, L. Trave-Massuyés, and Q. H. Lu, "An interval kalman filter enhanced by lowering the covariance matrix upper bound," *International Journal of Applied Mathematics and Computer Science*, vol. 31, no. 2, pp. 259–269, 2021.

[33] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[34] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011, p. 110.

[35] C. Regan, "In-flight stability analysis of the x-48b aircraft," 08 2008.

[36] E. Lavretsky and K. A. Wise, "Robust adaptive control," in *Robust and adaptive control: With aerospace applications*. Springer, 2012, pp. 317–353.

[37] M. R. Benjamin, "The heron," 2020. [Online]. Available: https://oceanai.mit.edu/pavlab/pmwiki/pmwiki.php?n=Robot.Heron

[38] E. Lavretsky and K. A. Wise, "Robust adaptive control," in *Robust and adaptive control: With aerospace applications*. Springer, 2012, pp. 58–71.