2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Interval Signal Temporal Logic for Robust Optimal Control

Luke Baird and Samuel Coogan

Abstract— **We propose a robust optimal control strategy for linear systems subject to bounded disturbances constrained to satisfy a Signal Temporal Logic (STL) formula with uncertain predicates. We encode such constraints using Interval STL (I-STL), an extension of STL to interval signals and predicates that accommodates efficient numerical implementations for verification and synthesis using interval arithmetic methods. Given an I-STL constraint, a quadratic cost function, and a bounded hyper-rectangular disturbance set, we construct a second robust optimal control problem using an embedding system with double the state dimension and the same cost function such that a solution to this second problem is feasible for the original problem. Moreover, owing to the numerical efficiencies of I-STL and the embedding, the computational complexity of this problem is, at worst, approximately equivalent to solving a non-robust optimal STL synthesis problem with double the state dimension, and we solve this problem as a mixed-integer quadratic program. We present a case study of a miniature blimp modeled as a 12-dimensional linear system subject to disturbances and tasked with a mission specified in I-STL with multiple nested temporal operators.**

I. Introduction

Robust optimal control seeks solutions to control problems while minimizing a cost in the presence of disturbances and constraints [1] and encompasses many engineering problems. Mature methods exist for solving such problems both online and offline [1]–[3]. However, as systems and specifications become increasingly complex, it becomes challenging to represent desired or allowed behavior with an explicit set constraint on the state and control. For instance, suppose that a drone must visit a charging station every twenty minutes while completing a mission. Instead, such time-varying constraints are naturally represented using *Signal Temporal Logic* (STL).

STL is a language for encoding high-level, temporally rich behavioral specifications [4]. STL links functions of state, called predicates, with Boolean and temporal operators and is equipped with both logical semantics and quantitative semantics with a robustness metric. Synthesizing control strategies to satisfy STL constraints has been studied from multiple vantage points [5] including gradient methods [6], mixed-integer linear programs [7], [8], control barrier function approximations [9], learning [10], and non-convex smooth approximation algorithms [11]. Notably, it is known that a sound and complete encoding for linear systems with convex predicates can be achieved by a mixed-integer convex program [7], [8]. To improve computational tractability, it is possible to limit the space of valid STL formulas to a fragment of the language space [6], [9] or to introduce conservatism by considering only locally optimal solutions [11].

In a traditional robust optimal control problem, there are two primary sources of uncertainty: uncertainty in the dynamics and uncertainty in the constraints. Both types of uncertainty may be handled by constraint tightening for the worst-case realization of the uncertainty [1]. However, to accommodate uncertain STL constraints with uncertain dynamics, direct constraint tightening is not straightforward due to the time-varying and state-dependent nature of the specification. For instance, the worst-case realization at some time k may depend on the worst-case realization at time $k-1$, or two different portions of the specification may have competing and different worst-case realizations of uncertainty.

For optimal control synthesis against STL specifications, there are several approaches for accommodating uncertainty in the state [5], [12]–[14]. In [12], a probabilistic extension of STL is introduced to handle probabilistic uncertainty in the state, but it cannot handle uncertainty in the predicates themselves. In [13], [15], given a polytopic disturbance in the system dynamics, a model predictive control approach is proposed using a slack variable and the positive normal form of the STL formula. The papers propose enforcing non-negative robustness on the lower-left corner of a hyper-rectangular over-approximation of the uncertain system. However, this method scales poorly with state and disturbance dimension. Moreover, to our knowledge, no prior work accommodates uncertainty in the predicates or STL formula construction itself.

To overcome these limitations, we propose to use Interval STL (I-STL), introduced in [16], to model uncertainty in the specification and accommodate uncertainty in dynamics. I-STL is an interval-valued extension of STL that naturally handles uncertain predicates and interval-valued signals using natural inclusion functions recursively in its definition, leading to a sound over-approximation of the true worst-case STL robustness. The syntax and semantics of I-STL are a natural interval extension of STL, and thus I-STL retains the intuitive appeal of STL. Moreover, algorithms for verification and synthesis are extended from STL in a computationally efficient and sound way by using mature interval arithmetic methods [17], [18]. In particular, [16] presented a computational package for I-STL by extending `stlpy` [7] using the `npinterval` implementation of interval arithmetic [17] that was demonstrated on a verification example and a simple proof-of-concept control synthesis problem.

In this paper, we present a framework for robust optimal control synthesis of linear systems subject to bounded dis-

Luke Baird and Samuel Coogan are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30318, USA {lbaird38,sam.coogan}@gatech.edu. S. Coogan is also with the School of Civil and Environmental Engineering.

turbances and constrained to satisfy an I-STL specification. To avoid the computational explosion that occurs when computing reachable tubes, and because I-STL is evaluated over interval-valued signals, we embed the system into a new system with twice the state space dimension such that hyper-rectangular reachable sets of the original system are obtained from a single trajectory of the embedding system [19]. We then construct a new optimal control problem using the embedding system that is cast as a mixed integer quadratic program using the approaches of [16]. Owing to the numerical efficiencies of I-STL and the embedding, the computational complexity of this problem is, at worst, approximately equivalent to solving a non-robust optimal STL synthesis problem with double the state dimension. We demonstrate our approach on a case study of a miniature blimp as a 12-dimensional linear system subject to disturbances.

This paper is outlined as follows. Section II presents notation and I-STL semantics. Section III formalizes the optimal control problem that this paper addresses. Section IV provides the theory used to solve this problem while providing formal guarantees. A simulated miniature blimp case study is presented in Section V along with a discussion of the empirical computational burden of our approach. Section VII concludes the paper.

II. Preliminaries

A. Notation

Let $\mathbb{IR}^n$ denote the set of all intervals on $\mathbb{R}^n$. We denote an interval $[x] \in \mathbb{IR}^n$ with $[x] := [\underline{x}, \overline{x}]$. We denote the standard partial order on $\mathbb{R}^n$ by $\leq$, that is, for $x, y \in \mathbb{R}^n$, $x \leq y$ if and only if $x_i \leq y_i$ for all $i \in \{1, \dots, n\}$. Denote a signal as $\mathbf{x} : \mathbb{N} \to \mathbb{R}^n$ and an interval signal as $[\mathbf{x}] : \mathbb{N} \to \mathbb{IR}^n$. For two sets $X, Y \subset \mathbb{R}^n$, let $\ominus$ denote the Pontryagin difference, $X \ominus Y = \{x \in X : x + y \in X, \forall y \in Y\}$.

B. Interval Signal Temporal Logic

I-STL was introduced recently as an interval extension of signal temporal logic [16]. The syntax, logical semantics, and quantitative semantics of I-STL, reviewed below, are natural interval interpretations of their counterpart for (non-interval) STL [4]. I-STL requires a set of interval predicate functions $\mathcal{I}$, which are interval-valued functions of the state, assumed to be in $\mathbb{R}^n$, or intervals of the state, that is, $\mathcal{M} \in \mathcal{I}$ is a map $\mathcal{M} : \mathbb{IR}^n \to \mathbb{IR}$. For example, in practice, a predicate function is commonly obtained as an inclusion function [20] that gives guaranteed interval overbounds of a continuous function of state.

Definition 1 (I-STL Syntax [16])**.** *Given a set $\mathcal{I}$ of interval predicate functions, I-STL syntax is defined by*

$$\varphi \triangleq (\mathcal{M}([x]) \subseteq [0, \infty]) \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \mathcal{U}_{[t_1,t_2]} \psi \tag{1}$$

for $\mathcal{M} \in \mathcal{I}$.

The I-STL quantitative and logical semantics follow.

Definition 2 (I-STL Quantitative Semantics [16])**.** *The interval robustness $[\rho]^\varphi$ of an I-STL specification φ evaluated over an interval signal $[\mathbf{x}]$ at time step $t \in \mathbb{N}$ is calculated recursively using natural inclusion functions [18] as*

$$\begin{aligned}
[\rho]^{\Pi}([\mathbf{x}], t) &= \mathcal{M}([\mathbf{x}](t)),\ \Pi = (\mathcal{M}([\mathbf{x}]) \subseteq [0, \infty]) \\
[\rho]^{\neg\varphi}([\mathbf{x}], t) &= -[\rho]^{\varphi}([\mathbf{x}], t) \\
[\rho]^{\varphi \wedge \psi}([\mathbf{x}], t) &= [\min]\big([\rho]^{\varphi}([\mathbf{x}], t), [\rho]^{\psi}([\mathbf{x}], t)\big) \\
[\rho]^{\varphi \vee \psi}([\mathbf{x}], t) &= [\max]\big([\rho]^{\varphi}([\mathbf{x}], t), [\rho]^{\psi}([\mathbf{x}], t)\big) \\
[\rho]^{\Box_{[t_1,t_2]}\varphi}([\mathbf{x}], t) &= \min_{t' \in [t+t_1, t+t_2]} \big([\rho]^{\varphi}([\mathbf{x}], t')\big) \\
[\rho]^{\Diamond_{[t_1,t_2]}\varphi}([\mathbf{x}], t) &= \max_{t' \in [t+t_1, t+t_2]} \big([\rho]^{\varphi}([\mathbf{x}], t')\big) \\
[\rho]^{\varphi \mathcal{U}_{[t_1,t_2]} \psi}([\mathbf{x}], t) &= \\
\max_{t' \in [t+t_1, t+t_2]} &[\min]\left([\rho]^{\varphi}([\mathbf{x}], t'), \min_{t'' \in [t+t_1, t']} \big([\rho]^{\psi}([\mathbf{x}], t'')\big)\right).
\end{aligned} \tag{2}$$

If the satisfaction of an I-STL specification can be completely determined by the values of the interval signal $\mathbf{x}$ over the first N time steps, we say that the I-STL specification has a finite horizon with horizon N. See [8], [21] for details on computing horizon lengths.

Definition 3 (I-STL Three-Valued Logical Semantics [16])**.** *The truth-value of I-STL formula φ evaluated over interval signal $[\mathbf{x}]$ is denoted $\big[[\mathbf{x}] \models \varphi\big]$ and is given by*

$$\big[[\mathbf{x}] \models \varphi\big] = \begin{cases} \textsc{True} & \text{if } [\rho]^{\varphi}([\mathbf{x}], 0) \subseteq [0, \infty] \\ \textsc{False} & \text{if } [\rho]^{\varphi}([\mathbf{x}], 0) \subseteq [-\infty, 0) \\ \textsc{Undef} & \text{else.} \end{cases} \tag{3}$$

When $\mathbf{x}$ is understood to be a signal of singleton sets, we write the above as $\big[\mathbf{x} \models \varphi\big]$.

Definition 4 (Safety)**.** *We say that an interval signal $[\mathbf{x}]$ is safe with respect to I-STL formula φ if $\big[[\mathbf{x}] \models \varphi\big] = \textsc{True}$.*

III. Problem Formulation

Let

$$x_{k+1} = Ax_k + Bu_k + Gw_k \tag{4}$$

be a linear time-invariant dynamical system with, at time k, state $x_k \in \mathbb{R}^n$, control input u_k, and disturbance w_k. Let $w \in \mathcal{W} \subseteq \mathbb{IR}^q$ lie in an hyper-rectangle, and let $u \in \mathcal{U} \subseteq \mathbb{R}^m$ lie in a polytope. Define the signal $\mathbf{x}$ as the sequence of states $\{x_0, x_1, \dots\}$ with $x_k \in \mathbb{R}^n$ and signal $\mathbf{u}$ as the sequence of inputs $\{u_0, u_1, \dots\}$ with $u_k \in \mathbb{R}^m$. Thus, $\mathbf{x}(k) = x_k$ and $\mathbf{u}(k) = u_k$. Unlike STL, I-STL handles uncertainty in the predicates themselves. Therefore, we first formulate the optimal control problem with an I-STL formula to permit uncertain predicates and then use I-STL with an interval signal to handle uncertainty. For clarity, we formulate our problem using a nominal cost—the cost function as if there were no disturbances, as defined in [1, Equation 15.6].

Problem Statement. *Given a finite-horizon I-STL formula φ, initial state $\tilde{x}$, and a quadratic nominal cost function J_0 :*

$\mathbb{R}^{nN} \times \mathcal{U}^N \to \mathbb{R}$, *solve the optimal control problem*

$$\begin{aligned} &\min_{\mathbf{u}\in\mathcal{U}^N} J_0(\hat{\mathbf{x}}, \mathbf{u}) \\ &\textit{s.t. } x_{k+1} = Ax_k + Bu_k + Gw_k \\ &\hat{x}_{k+1} = A\hat{x}_k + Bu_k \\ &u_k \in \mathcal{U} \quad \big[\mathbf{x} \models \varphi\big] = \textsc{True} \\ &\mathbf{u} \in \mathcal{U}^N \quad x_0 = \tilde{x} \quad \forall w_k, \in \mathcal{W} \end{aligned} \tag{5}$$

assuming that a feasible solution exists. That is, we want to find a sequence of control inputs u_k minimizing J_0 where $\rho^\varphi(\mathbf{x}, 0) \geq 0$ for all $w_k \in \mathcal{W}$.

IV. I-STL Embedding Solution

A. Embedding and Interval STL Reformulation

To solve our proposed problem, we use mixed monotone systems theory [19] to derive a sound lower and upper bound on the system state subject to uncertain dynamics. Then, by converting the dynamics to interval dynamics, we apply I-STL [16] to construct a sound over-approximation of the true interval robustness.

Let $A \in \mathbb{R}^{n\times n}$ have entries $\{a_{ij}\}$. Then, A^+ has entries $a^+_{ij} = \max\{0, a_{ij}\}$ and A^- has entries $a^-_{ij} = \min\{0, a_{ij}\}$. We note that (4) has a *decomposition function* [19] given by

$$\begin{bmatrix}\underline{x}_{k+1}\\ \overline{x}_{k+1}\end{bmatrix} = \begin{bmatrix}A^+ & A^-\\ A^- & A^+\end{bmatrix}\begin{bmatrix}\underline{x}_k\\ \overline{x}_k\end{bmatrix} + \begin{bmatrix}B\\ B\end{bmatrix}u_k + \begin{bmatrix}G^+ & G^-\\ G^- & G^+\end{bmatrix}\begin{bmatrix}\underline{w}_k\\ \overline{w}_k\end{bmatrix} \tag{6}$$

By re-casting the optimal control problem with mixed monotonicity and I-STL, we pose a solution to problem (5) by solving the optimal control program,

$$\begin{aligned} &\min_{\mathbf{u}\in\mathcal{U}^N} J(\hat{\mathbf{x}}, \mathbf{u}) \\ &\text{s.t. (6)} \quad u_k \in \mathcal{U} \quad w_k \in [w] \quad \big[[\mathbf{x}] \models \varphi\big] = \textsc{True}, \end{aligned} \tag{7}$$

where as before, $\hat{\mathbf{x}}$ is the trajectory corresponding to the nominal system dynamics in absence of disturbances. The explicit mixed-integer encoding of I-STL is given in [7], [16].

B. Soundness of the Optimal Control Problem

Our main theoretical contribution follows from the combination of the over-approximation from I-STL and the over-approximation from the embedding system dynamics.

Theorem 1 (Soundness of optimal control problem)**.** *Let* $\mathbf{u}$ *be a solution to* (7)*. Then,* $\mathbf{u}$ *is feasible for* (5)*.*

Proof. Let $[\mathbf{x}]$ be the interval state in the embedding dynamics (6) and let $\mathbf{x}$ be the state in the original dynamics (4). It suffices to show that if $\big[[\mathbf{x}] \models \varphi\big] = \textsc{True}$, then $\big[\mathbf{x} \models \varphi\big] = \textsc{True}$. Assume $\big[[\mathbf{x}] \models \varphi\big] = \textsc{True}$. We know that $\mathbf{x} \in [\mathbf{x}]$ as the dynamics are mixed monotone with respect to the decomposition function $d(x, w, \hat{x}, \hat{w}) = A^+x + G^+w + A^-\hat{x} + G^-\hat{w}$ [19, Example 9]. We do not include u as there is no uncertainty associated with it—it is a singleton. Furthermore, by [16, Theorem 1], we know that $\rho^\varphi(\mathbf{x}, 0) \subseteq \rho^\varphi([\mathbf{x}], 0)$. Therefore, $\underline{\rho}^\varphi(\mathbf{x}, 0) \geq \underline{\rho}^\varphi([\mathbf{x}], 0)$ and thus $\big[\mathbf{x} \models \varphi\big] = \textsc{True}$. ∎

V. Case Study - Miniature Autonomous Blimp

In this section, we present a case study of a a miniature blimp modeled as a 12- dimensional linear system subject to disturbances that is tasked with a mission specified in I-STL with multiple nested temporal operators. We use the computational package for I-STL presented in [16] and compare the computation time of our method to three restricted cases that removes sources of uncertainty: the first case removes the uncertainty in the I-STL predicates, the second case removes the disturbances in the dynamics, and the third case removes both sources of uncertainty. This third case reduces to the `stlpy` control synthesis method [7]. All experiments are performed on a computer with a Intel Xeon Gold 6230 CPU running Kubuntu 22.04[1].

A. Blimp Model and I-STL Specification

The model for the blimp is derived from 6-DOF rigid-body kinematics [22], with numerical parameters found in the papers [23], [24], linearized about the hover position and discretized in time. The blimp has an undermounted gondola with four lateral and six vertical fans, enabling holonomic control. The four inputs available are thus f_x, f_y, f_z, and τ_z. The resulting model is

$$x_{k+1} = e^{A\Delta t}x_k + \int_0^{\Delta t} e^{A\tau}Bd\tau u_k + G\Delta t w_k \tag{8}$$

where

$$A = \begin{bmatrix}A_1 & A_2\\ I_{6\times6} & 0_{6\times6}\end{bmatrix},$$

$$A_1 = \begin{bmatrix} -.073 & 0 & 0 & 0 & .007 & 0\\ 0 & -.073 & 0 & -.007 & 0 & 0\\ 0 & 0 & -.268 & 0 & 0 & 0\\ 0 & -.208 & 0 & -.189 & 0 & 0\\ .208 & 0 & 0 & 0 & -.189 & 0\\ 0 & 0 & 0 & 0 & 0 & -.168 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&-20.4&0\\ 0&0&0&0&0&-20.4\\ 0&0&0&0&0&0 \end{bmatrix},$$

$$B = \begin{bmatrix} 5.83 & 0 & 0 & 0\\ 0 & 5.83 & 0 & 0\\ 0 & 0 & 5.57 & 0\\ 0 & 20.62 & 0 & 0\\ 20.62 & 0 & 0 & 0\\ 0 & 0 & 0 & 171.8 \end{bmatrix}, \qquad G = \begin{bmatrix}I_{6\times6}\\ 0_{6\times6}\end{bmatrix},$$

with $\Delta t = 0.25s$, $x_k \in \mathbb{R}^{12}$, $u_k \in [-0.6, 0.6]^4 \subset \mathbb{R}^4$, and $w_k \in [-.0002, .0002]^6 \subset \mathbb{R}^6$ at each time step k. The first six elements of the state x are linear and angular velocity states, and the last six states are the world-frame position and orientation as Euler angles. The disturbance lies in an interval in the velocity states.

Consider a scenario where we wish to service a target location with the blimp while periodically visiting charging stations and then returning to the starting location. Specifically, starting in region $\mathcal{B}$ we require that the blimp reaches $\mathcal{A}$ within 6 seconds and remains there for 1.5 seconds. However, the blimp may not spend more than 3 seconds in region $\mathcal{B}$ to make way for, *e.g.*, other actors to visit the region.

[1]The code used to generate figures in this paper may be found at https://github.com/gtfactslab/Baird_CDC2024/

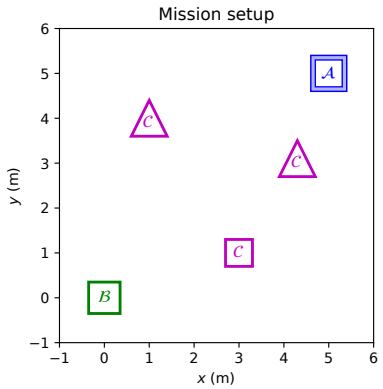

Fig. 1. Top-down view of the blimp mission setup. The blimp must reach the uncertain region $\mathcal{A}$ within 6 seconds, remain there for 1.5 seconds but not more than 3 seconds, and then return to region $\mathcal{B}$ within 20 seconds of the mission start time. It must visit a charging station $\mathcal{C}$ within the time intervals $[0, 6]$, $[7, 12]$, and $[13, 20]$.

Additionally, the blimp must visit a charging station every 6 seconds.

Let $p \in \mathbb{R}^2$ be the horizontal planar position coordinates of the blimp. Define region $\mathcal{A}$ as a square at $(5, 5)$ with an uncertain but bounded width in the interval $[0.5, 0.7]m$. Choose $\mathcal{B}$ as a $0.7m$-wide square at $(0, 0)$. The charging stations $\mathcal{C}$ are located at $(1, 4)$, $(4.3, 3)$, and $(3, 1)$, drawn as a triangle, rectangle, and triangle, respectively. This is illustrated in Figure 1. This mission specification is encoded with the following STL formula:

$$\begin{aligned}\varphi \models &\Diamond_{[0,\frac{6}{\Delta t}]}\Box_{[0,\frac{1.5}{\Delta t}]}(p \in \mathcal{A}) \wedge (p \notin \mathcal{A} | \Diamond_{[0,\frac{3}{\Delta t}]} p \notin \mathcal{A}) \\ &\Diamond_{[\frac{19}{\Delta t},\frac{20}{\Delta t}]}(p \in \mathcal{B}) \wedge \Diamond_{[0,\frac{6}{\Delta t}]}(p \in \mathcal{C}) \\ &\wedge \Diamond_{[\frac{7}{\Delta t},\frac{12}{\Delta t}]}(p \in \mathcal{C}) \wedge \Diamond_{[\frac{13}{\Delta t},\frac{20}{\Delta t}]}(p \in \mathcal{C}).\end{aligned} \tag{9}$$

We choose $\sum_k \hat{x}_k^\top Q \hat{x}_k + u_k^\top R u_k$, with $Q = \texttt{diag}(I_{3\times 3}, 0_{9\times 9})$ to minimize the velocity states and $R = I_{4\times 4}$, where $\hat{x}$ is the nominal system state.

B. Stability in the Embedding Space via Feedback Control and Transformation

To achieve a feasible solution to our problem, we need to ensure that the uncertainty bounds do not grow exponentially, otherwise they will become too large over a 20 second horizon. The linear system contains double integrator chains from velocity to position, and thus is only marginally stable. Borrowing from the tube model predictive control literature [2], we first define the nominal system dynamics, $\hat{x}_{k+1} = A\hat{x}_k + B\hat{u}_k$. Then, we introduce a feedback matrix K to stabilize the system, requiring that $u_k = K(x_k - \hat{x}_k) + \hat{u}_k$ where the input to the nominal system $\hat{u}_k$ is a feed-forward term. Next, we apply this input to the error dynamics equation with $e_k := x_k - \hat{x}_k$. Define $A_K := A + BK$. Solving for the error dynamics gives $e_{k+1} = A_K e_k + G w_k$.

However, due to the interval over-approximations and interactions, the embedding system is unstable. To achieve stability, we additionally apply a transformation matrix T derived from the eigen decomposition before applying our feedback matrix [25]. Thus, the reachable set of the embedding system converges to a bounded set, reminiscent of linear tube MPC [2].

Define $\xi := Tx$, $\varepsilon := Te$, and $A_{T,K} = TA_KT^{-1}$. The nominal dynamics become $\hat{\xi}_{k+1} = A_{T,K}\hat{\xi}_k + TB\hat{u}_k$ and the error dynamics become

$$\varepsilon_{k+1} = A_{T,K}\varepsilon_k + TGw_k. \tag{10}$$

The I-STL formula robustness $[\rho]^\varphi$ is a function of the original coordinates x. Although it is possible to create a new formula in ξ coordinates φ_ξ by applying T^{-1} to each predicate in φ, we instead add $x := T^{-1}\eta$ as a constraint in our optimization program and use φ for computational reasons: because the predicates define regions in the horizontal plane, all but two coordinates of linear predicates of the form $\{x : [\alpha]^\top x \le [\beta]\}$ are zero, and thus require fewer binary variables.

Now, we finally construct the interval version of our optimization problem,

$$\begin{aligned}&\min_{\hat{u}_0,\ldots,\hat{u}_{N-1}} \sum_k \hat{x}_k^\top Q \hat{x}_k + u_k^\top R u_k \\ &\text{s.t. } u_k = \hat{u}_k + K(\xi_k - \hat{\xi}_k) \\ &\hat{\xi}_{k+1} = A_{T,K}\hat{\xi}_k + TB\hat{u}_k \\ &\begin{bmatrix}\underline{\varepsilon}_{k+1} \\ \overline{\varepsilon}_{k+1}\end{bmatrix} = \begin{bmatrix}A_{T,K}^+ & A_{T,K}^- \\ A_{T,K}^- & A_{T,K}^+\end{bmatrix}\begin{bmatrix}\underline{\varepsilon}_k \\ \overline{\varepsilon}_k\end{bmatrix} \\ &\qquad + \begin{bmatrix}(TG)^+ & (TG)^- \\ (TG)^- & (TG)^+\end{bmatrix}\begin{bmatrix}\underline{w} \\ \overline{w}\end{bmatrix} \\ &[\xi] = \hat{\xi} + [\varepsilon] \qquad [x] = T^{-1}[\xi] \\ &\big[[\mathbf{x}] \models \varphi\big] = \textsc{True} \qquad u_k \in \mathcal{U}\end{aligned} \tag{11}$$

where $N = 20/\Delta t = 80$. We choose an initial condition $\hat{x}_0 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ -1.5\ 0\ 0\ 0]^\top$ to place the blimp at rest $1.5m$ off the ground, noting that the blimp is in a North-East-Down frame [23].

STL constraints may be soundly written as mixed-integer linear constraints [7], [8]. Building off the original `stlpy` implementation [7], the I-STL formulation is applied to convert the I-STL safety constraints to mixed-integer linear constraints [16]. We use the Gurobi 11.0.1 solver[2].

C. Results of the I-STL Control Synthesis

The implementation of our program directly transcribes (11) as an optimization problem in Gurobi but leads to both extra continuous and binary variables. However, Gurobi performs a presolve step that reliably removes a large number of redundant variables. The optimization program in Gurobi originally contains 187 308 continuous variables and 2 862 binary variables, but after Gurobi performs a presolve step there are 16 033 continuous and 2 862 binary variables remaining. The program takes $126.1s$ to solve. The resulting trajectory has a robustness for φ in the interval $[\rho]^\varphi([\mathbf{x}], 0) = [0.0, 0.098]$, while the nominal trajectory $\hat{x}$ has interval robustness $[\rho]^\varphi(\hat{\mathbf{x}}, 0) = [0.098, 0.098]$. Note that

[2]https://www.gurobi.com/

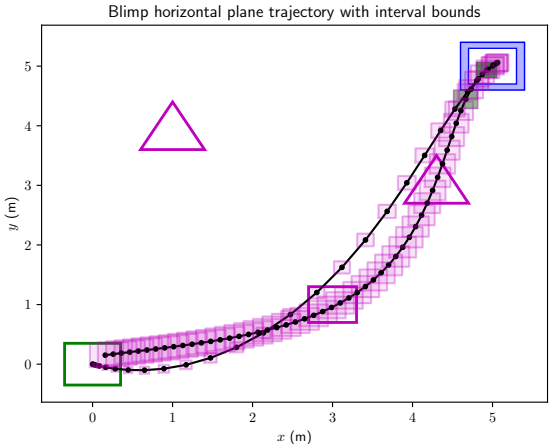

Fig. 2. Solution to the robust optimal control problem. The horizontal planar trajectory of the blimp obeying the specification φ is plotted. The magenta rectangles represent the uncertainty intervals at each time step. The blimp starts at $(0, 0)$ and is tasked with satisfying (11).

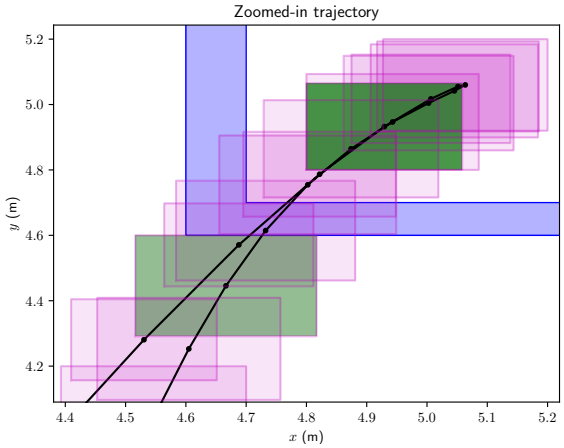

Fig. 3. Close-up view of the trajectory near $\mathcal{A}$. The upper and lower green regions are the first box inside $\mathcal{A}$ and the first box outside $\mathcal{A}$, respectively, for any realization of $\mathcal{A}$. Depending on the realization of the disturbance and specification, the blimp is within $\mathcal{A}$ for between 1.75 and 3.0 seconds, depending on the realization of $\mathcal{A}$. The first green box corresponds to time $6.0s$, and the second green box corresponds to time $8.5s$. Thus, this plot demonstrates that the trajectory satisfies the first part of φ.

the interval robustness comes from the uncertain predicates of $\mathcal{A}$ for the nominal trajectory, while the overall interval robustness represents the worst-case robustness depending on the uncertain predicate and disturbance. Figure 2 plots the horizontal planar trajectory along with the uncertainty from the mixed-monotone embedding system. Figure 3 is an expanded view of region $\mathcal{A}$, showing that the I-STL formula component regarding $\mathcal{A}$ is satisfied. Specifically, we note that the blimp is inside $\mathcal{A}$ for at least 1.75 seconds, but not more than 3.0 seconds for the respective worst-case disturbance and realization of $\mathcal{A}$. The time from when the blimp is guaranteed to lie inside $\mathcal{A}$ to when it is guaranteed to leave $\mathcal{A}$ is 2.5 seconds.

Although the program takes a long time to solve due to the exponential complexity of mixed-integer programs, the control solution is computed offline. Furthermore, we have a 12-dimensional model with a 24-dimensional embedding system considering a total of 80 time steps with a complicated I-STL formula φ containing nested temporal operators and uncertain predicates, leading to a relatively complex mixed-integer quadratic program.

D. Comparison to Control without I-STL or Disturbances

In this section, we offer a comparison regarding conservatism and implementation details to competing robust linear optimal control methods. Specifically, we explore the conservatism introduced by I-STL coupled with the conservatism introduced by mixed-monotonicity with polytopic set computations.

We note that it is not immediately obvious how to transform I-STL constraints into a traditional linear robust optimal control setting. One standard approach is, given a sequence of constraint sets $\mathcal{X} \subseteq \mathbb{R}^{n \times N}$ enforcing an STL specification, one can write $x_k \in \mathcal{X}_k$ at each time step along the horizon. For each time step k, one can compute a disturbance set S_k [1]. Then, the tightened constraints become $\hat{\mathcal{X}}_k = \mathcal{X}_k \ominus S_k$. For invariance conditions such as "always avoid an obstacle," it is apparent that a constraint tightening approach is to simply dilate the obstacle by a margin dependent on k. However, a time-dependent constraint with an uncertain I-STL predicate such as the one proposed for the target set $\mathcal{A}$ requires a careful analysis of the worst-case realization of the uncertain predicate.

Similar to [13], one method is to take each predicate π_j and consider the sets $\Pi = \{\alpha^\top x \leq \beta\}$, representing the satisfying set of a predicate. One can compute the exact disturbance polytope S_k by propagating the disturbance through the dynamics Minkowski summing with S_{k-1}. Then, one can tighten the constraint by computing $\Pi \ominus S_k$. However, the number of vertices grows to an intractable number, especially for integer programming. Therefore, one can instead tighten the constraints by computing $\Pi \ominus \overline{S}_k$, where $\overline{S}_k$ is a hyperrectangle over-approximation of S_k.

This approach struggles to deal with the portion of φ that require competing conservative estimates: eventually reaching $\mathcal{A}$ would shrink $\mathcal{A}$ in the worst case, while spending no more than 3 seconds in $\mathcal{A}$ would dilate $\mathcal{A}$ in the worst case. The paper [13] constructs a secondary signal equal to the negation of the first signal. Then, the same constraint tightening can be applied to both predicates, but one predicate monitors the original signal while the second monitors the secondary signal. However, this can artificially introduce model infeasibility. One can alleviate this by introducing a slack variable [13], but our method does not require this extra ingredient. Furthermore, by leveraging interval arithmetic afforded by `npinterval`, the implementation of this does not require the construction of a secondary signal. The proper tightening of the time-varying constraints are handled natively with natural inclusion functions [17], [18].

Although we mitigate the vertex explosion problem via mixed-monotonicity, there are other methods for handling this problem in the literature. For instance, one could use zonotopic [26] approximations and then take bounding boxes as needed to keep the program feasible [27], [28].

TABLE I

COMPUTE TIME AND NUMBER OF VARIABLES AFTER PRESOLVE

	Cont. Var.	Bin. Var.	Nom. Cost	Mean Time
All uncertainty	16 033	2 862	24.13	241.6s
No interval $\mathcal{A}$	16 033	2 862	16.79	225.6s
No w	15 887	2 862	15.03	183.8s
No int. $\mathcal{A}$ or w	1 426	2 439	11.89	112.7s

To verify safety, consider the error dynamics equation in the transformed coordinate space (10). Instead of performing a mixed monotone step, we can instead compute polytopic reachable sets given $w_k \in [\underline{w}, \overline{w}]$. Then, we can compute the true lower bound on the robustness by solving a separate mixed-integer linear program, enumerating the vertices on the disturbance sets generated by mixed-monotonicity. We report that the true $\underline{\rho}$ from this computation is 0.0.

We compare the computation time of our method for four scenarios in Table I. For each scenario, we uniformly sample an initial condition from a $0.2m$-wide box located at the origin. Then, we report the mean time over 10 solutions to mitigate variability in numerical solver times from Gurobi. The costs are reported for $\hat{x}_0 = 0$ except for the z position at $-1.5m$. First, we have our results from above. Then, we remove the uncertainty in the predicates defining $\mathcal{A}$. Next, we remove the disturbance but keep the uncertainty in the predicate. Finally, we reduce the problem to no disturbance and no uncertainty, which is equivalent to an STL control synthesis problem from `stlpy` [7]. We note that the last scenario does not use embedding dynamics and thus is a significantly smaller optimization problem, as shown in Table I. Furthermore, we note empirically that the average time for the no uncertainty case is roughly twice the average time of the predicate and dynamics uncertainty case.

VI. CONCLUSION

In this paper we demonstrated a sound method of solving a linear robust optimal control problem with interval disturbances and an I-STL constraint containing uncertain predicates. By applying I-STL and mixed monotone system theory, we soundly over-bound the dynamics and the interval robustness. Our program is implemented as a mixed-integer quadratic program and we demonstrated it with a 12-dimensional model of a miniature blimp. In this paper, we focused on the single shot optimal problem, but this could be used in a model predictive control scheme similar to [8], [13]. Future work could investigate extending this method for a model predictive control or runtime assurance framework and could explore alternative reachability methods including ellipsoidal or zonotopic methods.

REFERENCES

[1] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[2] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, LLC, 2022.

[3] S. Chen, V. Preciado, M. Manfred, and N. Matni, "Robust model predictive control with polytopic model uncertainty through system level synthesis," *Automatica*, vol. 162, p. 111431, 2024.

[4] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Springer, 2004, pp. 152–166.

[5] C. Belta and S. Sadraddini, "Formal methods for control synthesis: An optimization perspective," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 115–140, 2019.

[6] K. Leung, N. Aréchiga, and M. Pavone, "Back-propagation through signal temporal logic specifications: Infusing logical structure into gradient-based methods," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2020, pp. 432–449.

[7] V. Kurtz and H. Lin, "Mixed-integer programming for signal temporal logic with fewer binary variables," *IEEE Control Systems Letters*, vol. 6, pp. 2635–2640, 2022.

[8] L. Baird and S. Coogan, "Runtime assurance from signal temporal logic safety specifications," in *American Control Conference (ACC)*, 2023, pp. 3535–3540.

[9] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE control systems letters*, vol. 3, no. 1, pp. 96–101, 2018.

[10] K. Wabersich and M. Zeilinger, "Nonlinear learning-based model predictive control supporting state and input dependent model uncertainty estimates," *international journal of robust and nonlinear control*, vol. 31, no. 18, pp. 8897–8915, 2021.

[11] Y. Gilpin, V. Kurtz, and H. Lin, "A smooth robustness measure of signal temporal logic for symbolic control," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 241–246, 2020.

[12] D. Sadigh and A. Kapoor, "Safe control under uncertainty with probabilistic signal temporal logic," in *Proceedings of Robotics: Science and Systems XII*, 2016.

[13] S. Sadraddini and C. Belta, "Robust temporal logic model predictive control," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 772–779.

[14] S. Farahani, V. Raman, and R. Murray, "Robust model predictive control for signal temporal logic synthesis," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 323–328, 2015.

[15] S. Sadraddini and C. Belta, "Formal synthesis of control strategies for positive monotone systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp. 480–495, 2018.

[16] L. Baird, A. Harapanahalli, and S. Coogan, "Interval signal temporal logic from natural inclusion functions," *IEEE Control Systems Letters*, 2023.

[17] A. Harapanahalli, S. Jafarpour, and S. Coogan, "A toolbox for fast interval arithmetic in numpy with an application to formal verification of neural network controlled systems," in *ICML 2023 Workshop on Formal Verification of Machine Learning*, 2023.

[18] L. Jaulin, M. Kieffer, D. Olivier, and E. Walter, *Applied Interval analysis*. Springer, 2001.

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

[20] S. Walcher, "On cooperative systems with respect to arbitrary orderings," *Journal of Mathematical Analysis and Applications*, vol. 263, no. 2, pp. 543–554, 2001.

[21] A. Dokhanchi, B. Hoxha, and G. Fainekos, "On-line monitoring for temporal logic robustness," in *International Conference on Runtime Verification*. Springer, 2014, pp. 231–246.

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

[23] Q. Tao, J. Wang, Z. Xu, T. X. Lin, Y. Yuan, and F. Zhang, "Swing-reducing flight control system for an underactuated indoor miniature autonomous blimp," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 1895–1904, 2021.

[24] Q. Tao, J. Tan, J. Cha, Y. Yuan, and F. Zhang, "Modeling and control of swing oscillation of underactuated indoor miniature autonomous blimps," *Unmanned Systems*, vol. 9, no. 01, pp. 73–86, 2021.

[25] M. Abate and S. Coogan, "Improving the fidelity of mixed-monotone reachable set approximations via state transformations," in *American Control Conference*, 2021, pp. 4674–4679.

[26] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *International workshop on hybrid systems: Computation and control*. Springer, 2005, pp. 291–305.

[27] S. Bak and P. S. Duggirala, "Hylaa: A tool for computing simulation-equivalent reachability for linear systems," in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, 2017, pp. 173–178.

[28] M. Althoff and D. Grebenyuk, "Implementation of interval arithmetic in CORA 2016," in *Proc. of the 3rd International Workshop on Applied Verification for Continuous and Hybrid Systems*, 2016, pp. 91–105.