Control Synthesis using Signal Temporal Logic Specifications with Integral and Derivative Predicates

Ali Tevfik Buyukkocak, Derya Aksaray, and Yasin Yazıcıoğlu

Abstract—In many applications, the integrals and derivatives of signals carry valuable information (e.g., cumulative success over a time window, the rate of change) regarding the behavior of the underlying system. In this paper, we extend the expressiveness of Signal Temporal Logic (STL) by introducing predicates that can define rich properties related to the integral and derivative of a signal. For control synthesis, the new predicates are encoded into mixed-integer linear inequalities and are used in the formulation of a mixed-integer linear program to find a trajectory that satisfies an STL specification. We discuss the benefits of using the new predicates and illustrate them in a case study showing the influence of the new predicates on the trajectories of an autonomous robot.

I. INTRODUCTION

Temporal logics such as Metric Temporal Logic (MTL) [1] and Signal Temporal Logic (STL) [2], [3] are expressive specification languages that can define properties of dense-time real-valued signals with explicit spatial and time parameters. Different than the existing temporal logics with an automaton representation, MTL and STL contain predicates in the form of inequalities and are endowed with a metric called robustness degree that can quantify how good a signal satisfies a specification [3]. Robustness degree not only gives a yes/no answer but also provides a real value that quantifies the degree of satisfaction. Such a metric also enables to formulate an optimization problem that solves for a trajectory satisfying the temporal logic specifications (e.g., [4]–[7]).

Standard STL capabilities are enhanced by several studies (e.g., [8]–[12]). For example, since conventional robustness degree focuses on critical time instants, authors of [8] and [9] define new measures to differentiate the satisfaction of predicates achieved at multiple time instants from the instantaneous satisfaction of them. With a similar motivation, a temporal operator is defined in [10] that explicitly specifies how long a predicate must be satisfied. However, existing approaches do not accommodate predicates expressing cumulative success or local behavior over a time interval. To this end, this paper introduces integral and derivative predicates for STL. While the derivative predicate enables to define properties for the rate of change of the signal, the integral predicate allows the definition of properties such as average or cumulative progress at desired time intervals.

This paper is closely related to [13] which defines predicates to compare signal values at different time instants to find local extrema of the signal. Also in [14]–[16], authors propose the notions of cumulative and average robustness metrics calculated via the signal values at different time steps. With these methods, while the system can satisfy a conventional predicate as long and robust as possible by maximizing the new metrics, it does not result in the satisfaction of cumulative properties such as getting a specific amount of reward/value within a given time window. To this end, we propose to define new integral and derivative predicates that can take into account the signal values at different time steps to define cumulative and relative specifications within desired time windows. By the proposed predicates, the local and global signal characteristics can be controlled extensively without losing the existing STL capabilities.

II. PRELIMINARIES

A. Notation

In this paper, $\mathbb{R}_{\geq 0}$ refers to the set of nonnegative real numbers, and $\mathbb{R}^n$ denotes the set of n-dimensional real-valued vectors. We represent $l_1$ and $l_2$ norms by the operators of $|·|$ and $∥·∥$. Right and left time derivatives of a function are denoted as $\frac{d(·)}{dt^+}$ and $\frac{d(·)}{dt^-}$, respectively.

B. Signal Temporal Logic

Rich time series can compactly be expressed by the Signal Temporal Logic [2]. In this paper, we use the following STL fragment:

$$\phi ::= \mu | \neg \phi | \phi_1 \land \phi_2 | F_{[t_1,t_2]} \phi,$$  \hspace{1cm} (1)

where $t_1, t_2 \in \mathbb{R}_{\geq 0}$ are time bounds with $t_2 \geq t_1$; $F_{[t_1,t_2]} \neg$ are finally (i.e., eventually), negation, and conjunction (i.e., and) operators, respectively; $\phi$ is an STL formula, and $\mu$ is a predicate in the inequality form such as $\mu = g(x) \geq c$ with a constant $c \in \mathbb{R}$, a signal $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$, and a function $g : \mathbb{R}^n \rightarrow \mathbb{R}$. Remaining useful operators are generated from the others as follows:

$$G_{[t_1,t_2]} \phi = \neg F_{[t_1,t_2]} \neg \phi \text{ is globally (i.e., always) operator, and } \phi_1 \lor \phi_2 = \neg (\neg \phi_1 \land \neg \phi_2) \text{ is disjunction (i.e., or) operator. We can also define an implication operator as } \phi_1 \Rightarrow \phi_2 = \neg \phi_1 \lor \phi_2.$$

Let $x_t$ denote the value of $x$ at time $t$ where $x$ represents the run (or trajectory) of the system. Satisfaction of an STL formula by the part of the signal starting from $t$, i.e., $(x,t)$, is determined as follows:

$$(x,t) \vdash \mu \iff g(x_t) \geq c,$$
$$(x,t) \vdash \neg \mu \iff \neg ((x,t) \models \mu),$$
$$(x,t) \vdash \phi_1 \lor \phi_2 \iff (x,t) \models \phi_1 \text{ and } (x,t) \models \phi_2,$$
$$(x,t) \vdash \phi_1 \land \phi_2 \iff (x,t) \models \phi_1 \text{ or } (x,t) \models \phi_2,$$

$$(x,t) \vdash F_{[t_1,t_2]} \phi \iff \forall \; t' \in [t + t_1, t + t_2], (x,t') \models \phi,$$
$$(x,t) \vdash G_{[t_1,t_2]} \phi \iff \exists \; t' \in [t + t_1, t + t_2], (x,t') \models \phi.$$  \hspace{1cm} (2)
While \((x, t) \models F_{[t_1, t_2]} \phi\) implies that \(\phi\) must hold at least in one time instant between \([t + t_1, t + t_2]\), \((x, t) \models \ldots\) can depend on part of the past, present, or future of the signal. The new predicates are not the same.

The satisfaction of specifications is indicated as either True or False for the most of the temporal logics. STL, on the other hand, is endowed with a metric called robustness degree, \(r(x, \phi, t) \in \mathbb{R}\), a real-valued function that is used to quantify the satisfaction of an STL formula \(\phi\) with respect to a signal \((x, t)\). While positive robustness degree indicates the satisfaction of \(\phi\), negative one represents a violation. In general, zero robustness degree is considered inconclusive, but we consider this case as satisfaction in this paper. The robustness degree metric can be formally and recursively defined as follows [3]:

\[
\begin{align*}
\ r(x, g(x) \geq c, t) &= g(x_t) - c, \\
\ r(x, -(g(x) \geq c), t) &= -r(x, g(x) \geq c, t), \\
\ r(x, \phi_1 \land \phi_2, t) &= \min\{r(x, \phi_1, t), r(x, \phi_2, t)\}, \\
\ r(x, F_{[t_1, t_2]} \phi, t) &= \max\{\ r(x, \phi, t'), \ \ t' \in [t+t_1, t+t_2] \}, \\
\ r(x, G_{[t_1, t_2]} \phi, t) &= \min\{\ r(x, \phi, t'), \ \ t' \in [t+t_1, t+t_2] \}.
\end{align*}
\]

Robustness degree has limitations due to its main focus on the critical time instances and neglecting the remaining parts of the signal. For instance, suppose that eventually the value of \(x\) needs to be at least 1 within \([0, 100]\), i.e., \(\phi = F_{[0,100]} \ x \geq 1\). Consider two trajectories \(x\) and \(x'\) shown in Fig. 1. Since, the degree of satisfaction is evaluated over any critical instant at which the predicate \(x_t \geq 1\) is satisfied for \(t \in [0, 100]\), although the signal \(x\) satisfies the predicate for a longer time, robustness degree of both signals would be equal, i.e., \(r(x, \phi, 0) = r(x', \phi, 0)\). Therefore, the conventional robustness degree cannot differentiate between such cases and cannot provide comprehensive information about the signals \(x\) and \(x'\). To address this issue, [8]–[10] define new measures that can capture the duration of predicate satisfaction. While existing metrics can track how long a predicate \(x(t) \geq \delta\) is satisfied within an interval \([0, T]\), they are not able to address a notion of cumulative success such as \(\int_0^T x(t) \, dt \geq \delta\) within an interval \([0, T]\).

### III. Problem Statement

#### A. Motivation

While the duration of satisfaction of a conventional predicate may be important for certain applications; it may also be desired to meet some cumulative success criteria. Authors in [14]–[16], propose discrete-time cumulative and average robustness degrees calculated by the summation of the robustness degrees of the same predicate at different time steps. This enables to have a predicate satisfied as long and robust as possible. In [13], a freezing operator is used to store the signal values at different time instants in memory and to compare them inside the predicates which enables to find local extrema and examine the oscillatory behavior.

In other words, the satisfaction of a predicate now depends on more than one signal values at different time instants. Similarly, in the case of progressive events, one may desire to evaluate the contributions of the past, future, or both signal values to determine a satisfaction at the current time. Therefore, the success may depend on the accumulation of signal values from different time instants. We can define such success criteria inside the same predicate, and use the integral of a signal over a given bounded time interval to assess satisfaction.

**Example 1.** Consider a continuous signal function that represents the speed, \(\|v\|\). Let trajectory of \(\|v\|\) be the same with the signal \(x\) in Fig. 1. Assume that at \(t = 20\), system is desired to travel more than \(11\) m within the last \(10\) s. In other words, integration of \(\|v\|\) over \([10, 20]\) must be at least \(11\). One may check the satisfaction of this specification at \(t = 20\) by inspecting the red curve in Fig. 1 obtained by the integration of \(\|v\|\) (i.e., \(x\)) over the window of the previous \(10\) s at each time instant.

The interval of interest over which the integrated signal may not depend only on the past values. One may also specify thresholds on the integrals defined partially or fully over the future signal values. In Fig. 1, sample intervals defined in the past, future, or both are partially shown with red, magenta, and orange, respectively. For each time instant, the window of \(10\) s is shifted throughout the time axis to generate the integral curves. While at \(t = 50\) total distance traveled between \([45, 55]\) (depending on both past and future values) is more than \(11\) m, at \(t = 80\) this time the distance traveled during \([80, 90]\) (depending on only future values) is less than the given threshold.

In the next section, we introduce STL predicates whose satisfaction can depend on part of the past, present, or future of the signal. The new predicates are not the same.
as using the standard predicates with temporal operators. For example, consider some specifications defined between $t = 80$ and $t = 90$ over $x$ represented in Fig. 1. Trajectory of $(x, 80)$ does not satisfy the predicate $x \geq 5$ because $x_{80} < 5$. Moreover, the trajectory $(x, 0)$ does not satisfy an STL formula $F_{[80, 90]} x \geq 5$ because there is no $t \in [80, 90]$ such that $x_t \geq 5$. Now, consider another predicate including an integral over time whose satisfaction does not depend only on the current time step but determined by the signal values within the bounds of the integral. The same signal $(x, 0)$ satisfies such a predicate $\int_{80}^{90} x_{\tau} \geq 5$ as shown in Fig. 1 with magenta. Furthermore, temporal operators can also be used with such a predicate to create more complex specifications by shifting the integral interval.

One may try to represent such specifications by simply considering the integral or derivative of the original signal, and using it with conventional predicates. For example, again consider the magenta signal in Fig. 1. Let the specification and using it with conventional predicates. For example, again consider the magenta signal in Fig. 1. Let the specification

\[
\phi := \mu | \mu_{a,b}^t | \mu_{a,-}^t | \neg \phi \land \phi_2 \big| \big| F_{[a,b]} \phi,
\]

where the satisfactions of $\mu_{a,b}^t$ and $\mu_{a,-}^t$ are determined for any signal $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ as follows:

\[
\begin{align*}
(x, t) &\models \mu_{a,b}^t \iff \int_{t+a}^{t+b} g(x_{\tau}) d\tau \geq c, \\
(x, t) &\models \mu_{a}^t \iff \frac{d g(x_{t})}{d t} \geq c, \\
(x, t) &\models \mu_{a,-}^t \iff \frac{d g(x_{t})}{d t} \geq c, \\
(x, t) &\models \neg \mu_{a,b}^t \iff \neg (x, t) \models \mu_{a,b}, \\
(x, t) &\models \neg \mu_{a,-}^t \iff \neg (x, t) \models \mu_{a,-}.
\end{align*}
\]

Note that the signal $x$ is undefined for $t < 0$, therefore we assume $t + a \geq 0$ throughout the paper. By preserving the quantitative semantics for the common operators defined in (3), we can quantify the satisfaction of the new predicates similarly as:

\[
\begin{align*}
\tau(x, \mu_{a,b}^t, t) &\equiv \int_{t+a}^{t+b} g(x_{\tau}) d\tau - c, \\
\tau(x, \mu_{a}^t, t) &\equiv \frac{d g(x_{t})}{d t} - c, \\
\tau(x, \mu_{a,-}^t, t) &\equiv \frac{d g(x_{t})}{d t} - c, \\
\tau(x, \neg \mu_{a,b}^t, t) &\equiv - \tau(x, \mu_{a,b}, t), \\
\tau(x, \neg \mu_{a,-}^t, t) &\equiv - \tau(x, \mu_{a,-}^t, t).
\end{align*}
\]

An integral predicate can have negative time bounds with respect to the bounds of the outer temporal operators (e.g., eventually and globally). We can also use the integral predicate alone by using nonnegative time bounds or defining it with the negative time bounds at future time steps. The horizon definition of STL [17] can be extended for the integral predicate and nested use of it with the temporal
operators as follows:
\[ \mu_{[a,b]} = \int_a^b g(x, \tau) d\tau \geq c \]
\[ \Rightarrow \text{hrz}(\mu_{[a,b]}) = \max(|a|, b, b-a), \quad (9) \]
\[ \phi = F_{[t_1, t_2]}[\mu_{[a,b]}] \quad \text{or} \quad \phi = G_{[t_1, t_2]}[\mu_{[a,b]}] \]
\[ \Rightarrow \text{hrz}(\phi) = \max(t_2-t_1, t_2+b), \]
where \( t_1, t_2 \in \mathbb{R}_{\geq 0} \) and \( a, b \in \mathbb{R} \) are time bounds with \( t_2 \geq t_1, b > a \), and a constraint of \( t_1 + a \geq 0 \) for the sake of nonnegative global time.

**Remark 1.** For discrete-time signals, we define the integral and derivative predicates at time \( t \) assuming a discrete-time signal function \( g(x) \) as:
\[ \mu_{[a,b]} = \sum_{k'=k+a/\delta t}^{k+b/\delta t-1} g(x_{k'=\delta t}) \delta t \geq c, \]
\[ (10) \]
\[ \mu_{\phi} = g(x_{k+b\delta t}) - g(x_{k}\delta t) \geq c \delta t \]
\[ \mu_{\phi} = g(x_{k}\delta t) - g(x_{(k-1)\delta t}) \geq c \delta t, \]
where \( k \in \mathbb{Z} \) is the step number, and \( \delta t \) is the time step such that \( t = k\delta t \).

Modified robustness metrics [14–16] can also be used to quantify cumulative properties. However, instead of modifying the semantics of existing temporal operators, we introduce the integral predicate as a new operator with its own qualitative and quantitative semantics that can easily be used with the standard STL syntax and semantics. For example, consider a signal \( x = \{1, 1, 1, 1, 1, 1, 2, \epsilon\} \), where \( \epsilon \) is a small arbitrary number since \( \mu_{\phi}^{[a,b]} \) does not consider the last signal value in discrete-time signals. Suppose that the sum of any two consecutive signal values need to eventually be at least 3. The proposed integral predicate enables to define this as \( \phi = F_{[0,4]}[\mu_{[0,2]}] \geq 3 \) where \( \mu_{[0,2]} = \int_0^2 x \delta t, \) for which the robustness degree of \( x \) with respect to \( \phi \) is \( r(x, \phi) = 0 \). Thus, \( x \) barely satisfies the specification \( \phi \). Now, consider the cumulative robustness degree, \( \rho^+() \) [14] that is the closest measure to our proposed idea. One can express the task in \( \phi \) by using the cumulative robustness degree as \( \phi' = F_{[0,4]}(\rho^+(x, F_{[0,1]} x \geq 0) \geq 3) \) for the same signal, which has \( \rho^+(x, \phi') = -4 \) implying a violation while \( x \) actually satisfies \( \phi \).

**C. Optimal Control Problem**

We consider a discrete time system \( x^+ = f(x,u) \) evolving over a continuous state space with state and input vectors of \( x \in \mathbb{R}^n \) and \( u \in \mathbb{R}^m \), respectively. Accordingly, the signal we evaluate becomes the finite state trajectory, \( x = [x_{0\delta t}, x_{1\delta t}, \cdots, x_{H\delta t}] \in \mathbb{R}^{n \times (H+1)} \) where \( H \) is the length of the mission horizon.

The state trajectory, \( x \), is desired to achieve an STL specification, \( \Phi \). Note that the mission horizon has to be longer than the specification horizon, i.e., \( H > \text{hrz}(\Phi)/\delta t \) to determine the satisfaction or violation of \( \Phi \).

**Problem 1.** Given a system with discrete-time dynamics, \( x^+ = f(x,u) \), find the optimal control policy over the horizon \( u^* = [u_{0\delta t}^*, u_{1\delta t}^*, \cdots, u_{(H-1)\delta t}^*] \in \mathbb{R}^{m \times H} \) to achieve a global specification \( \Phi \):
\[ u^* = \arg\min \sum_{k=0}^{H-1} J(x_{k\delta t}, u_k) \]
\[ s.t. \quad x^+ = f(x,u), \quad x_0 = x_0, \quad r(x, \Phi, 0) \geq 0 \]

where \( J(x_{k\delta t}, u_k) \) is a running cost as a function of state vector and control inputs, \( J : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R} \), \( x_0 \) is the initial state vector, and \( r(x, \Phi, 0) \geq 0 \) enforces the satisfaction of the temporal logic constraints by requiring the robustness degree to be nonnegative.

**IV. Solution Approach**

The solution of (11) requires a nonnegative robustness degree ((3) and (8)) to enforce satisfaction. However, it is computed via recursive definitions of computationally expensive and non-smooth min and max functions. Therefore, using it as a constraint in the optimization or feasibility problems makes the problem non-trivial. Alternatively, satisfaction of \( \Phi \) can be encoded as a set of constraints with binary variables in the form of \( z^\Phi[k] \in \{0,1\} \) [4, 5]. For each predicate in the form of inequality, a couple of big M constraints can be written depending on a binary variable as
\[ \mu = g(x) \geq c \begin{cases} g(x) - c \geq M(z^\Phi - 1), \\ g(x) - c \leq Mz^\Phi, \end{cases} \]
\[ (12) \]
where \( M \in \mathbb{R}^+ \) is a sufficiently large number. Similarly, the satisfaction of an integral predicate can also be encoded as
\[ \phi = \mu_{[a,b]} \begin{cases} \sum_{k'=k+a/\delta t}^{k+b/\delta t-1} g(x_{k'=\delta t}) \delta t - c \geq M(\Phi^k - 1), \\ \sum_{k'=k+a/\delta t}^{k+b/\delta t-1} g(x_{k'=\delta t}) \delta t - c \leq Mz^\Phi, \end{cases} \]
\[ (13) \]

The derivative predicate is encoded in a similar fashion as follows:
\[ \mu_{\phi} = \frac{dg(x)}{dt^+} \geq c \begin{cases} g(x_{(k+1)\delta t}) - g(x_{k\delta t}) - c \delta t \geq M(z^\Phi - 1), \\ g(x_{(k+1)\delta t}) - g(x_{k\delta t}) - c \delta t \leq Mz^\Phi - k], \end{cases} \]
\[ \mu_{\phi} = \frac{dg(x)}{dt^-} \geq c \begin{cases} g(x_{k\delta t}) - g(x_{(k-1)\delta t}) - c \delta t \geq M(z^\Phi - 1), \\ g(x_{k\delta t}) - g(x_{(k-1)\delta t}) - c \delta t \leq Mz^\Phi - k], \end{cases} \]
\[ (14) \]

Starting with the predicates, remaining binary constraints corresponding to an STL formula can be built into each other. We follow the recipe described in [5] to encode the connections between temporal operators and new predicates (13), (14). As a result, the satisfaction of the general formula, \( \Phi \), which is previously implied by \( r(x, \Phi, k) \geq 0, \) is now rendered to \( z^\Phi_k = 1 \). Since we handle the specifications on the trajectory of the state vector, \( x_k \), inside the temporal logic constraints, in the optimization, we penalize only control inputs, \( u_k \). Therefore, we define the running cost as the \( l_1 \)-norm of the control input, i.e., \( J(u_k) = |u_k| \).
Nonlinear systems can be linearized around known equilibrium points. Furthermore, the mission constraints can be encoded as linear inequalities with binary variables as in (12), (13), (14). Accordingly, under linear dynamics and predicates, the constrained optimization problem in (11) can be posed as a mixed-integer linear program.

Problem 2.

\[
\mathbf{u}^* := \arg \min_{\mathbf{u}} \sum_{k=0}^{N-1} |u_k| \quad \text{s.t.} \quad x_{k+1} = Ax_k + Bu_k, \quad x_0 = x_0, \quad z_0 = 1, \quad (15)
\]

where \( A \in \mathbb{R}^{n \times n} \) and \( B \in \mathbb{R}^{n \times m} \) are the relevant system and input matrices, respectively.

Note that solving Problem 2 enforces the satisfaction of the STL specification (due to the existence of constraint \( z_0 = 1 \)). It can be solved via several off-the-shelf tools such as MATLAB’s built-in integer programming solver, intlinprog, and Gurobi [19]. This problem can also be formulated for maximal satisfaction of the STL specification by relaxing each predicate with a slack variable to be minimized with the negation-free STL specifications [8], [9].

V. CASE STUDY

To illustrate the functionality of the new predicates, here we define a complementary example.

Example 2. Consider an autonomous robot with discrete-time double integrator dynamics

\[
x_{k+1} = \begin{bmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.5 \delta t^2 & 0 \\ 0 & 0.5 \delta t^2 \\ 0 & 0 & \delta t \\ 0 & 0 & 0 & \delta t \end{bmatrix} u_k, \quad (16)
\]

with the state vector \( x = [x, v_x, v_y, y]^T \) where \( v_x, v_y, y \in \mathbb{R} \) are the velocities in \( x, y \) directions, respectively; and the input vector \( u = [u_x, u_y]^T \) where \( u_x, u_y \in \mathbb{R} \) are the specific forces in given directions. The goal of the mission is to monitor the natural habitat in predefined areas by observing as much as possible without disturbing the members of it. Mission specifications are given as follows: i) Continuously service regions A and B for at least 3 s and 6 s, respectively, and while in B, travel inside it for at least 2 m in both directions. ii) Whenever in A or B, do not disturb the members of the natural habit by limiting the acceleration to 0.25 \( m/s^2 \) in each direction. iii) Overall the robot cannot exceed an acceleration of 0.5 \( m/s^2 \) in any direction. iv) Since the habitats of region C are adversarial, whenever you are in C keep the horizontal velocity at least 1 \( m/s \) to leave the C immediately (its shortest edge lies in the horizontal direction). With a total mission time of 20 s, we can express these specifications as follows:

\[
\Phi = P_{[0,17]}(G_{[0,3]}R_A) \land P_{[0,14]}(G_{[0,6]}R_B \land \mu_{[0,6]}^{1,1} \land \mu_{[0,6]}^{1,2}) \\
\land G_{[1,20]}((R_A \lor R_B) \Rightarrow (\mu_{[0,6]}^{4,1} \land \mu_{[0,6]}^{4,2})) \\
\land G_{[0,19]}(\mu_{[0,6]}^{5,1} \land \mu_{[0,6]}^{5,4}) \\
\land G_{[0,20]}(R_C \Rightarrow |v_x| \geq 1), \quad (17)
\]

where the integral predicates are defined to constrain the total traveled distance as \( \mu_{[0,6]}^{1,1} = \int_0^6 |v_x| d\tau \geq 2 \) and \( \mu_{[0,6]}^{1,2} = \int_0^6 |v_y| d\tau \geq 2 \); the derivative predicates limit the acceleration either temporally or throughout the mission as \( \mu_{[0,6]}^{4,1} = |v_x| \leq 0.25, \mu_{[0,6]}^{4,2} = |v_y| \leq 0.25, \mu_{[0,6]}^{5,1} = |v_x| \leq 0.5, \) and \( \mu_{[0,6]}^{5,4} = |v_y| \leq 0.5 \). The reason for specifying left-derivatives for the predicates inside the A and B is to avoid an aggressive entrance into them. The regions in two-dimensional \( x - y \) plane can be represented with linear predicates as follows:

\[
R_A = x \geq 1.5 \land x \leq 2 \land y \geq 4.75 \land y \leq 5.25, \\
R_B = x \geq 4 \land x \leq 5 \land y \geq 1 \land y \leq 3, \\
R_C = x \geq 2 \land x \leq 4 \land y \geq 1 \land y \leq 5. \quad (18)
\]

To bound the total traveled distance, we define integral predicates (Def. (1)) that will keep the distance in each particular direction at least 2 \( m \) by integrating the speeds in that direction. Moreover, we also constrained the maximum acceleration in any direction using derivative predicates defined in Def. 2. Although one can limit the control inputs to that in this particular example, where the inputs are the acceleration, it is not generally the case for other systems. It is also worth to mention that the expressions with the absolute value operators in (17) can be represented by multiple linear inequalities.

A. Simulations

In the case study, YALMIP [20] is used to model the problem in (15), which is solved via Gurobi [19] in MATLAB R2019b. A laptop computer with 1.8 GHz, Intel Core i5 processor is used to run the simulations with \( \delta t = 1 \) s.

First, the system trajectory is computed according to the original specification (17). Then, for comparison, the system trajectories are computed to satisfy specifications 1) without the integral and derivative predicates defined in the green areas, 2) without the integral predicates, \( \mu_{[0,6]}^{1,1} \) and \( \mu_{[0,6]}^{1,2} \), and 3) without the derivative predicates, \( \mu_{[0,6]}^{4,1} \) and \( \mu_{[0,6]}^{4,2} \). The generated trajectories are illustrated in Fig. 2. It is evident from the figures that with the integral predicates, the robot does not only service B but also it is required to travel inside the region more to monitor the natural habitat better. Moreover, the derivative predicates constrain the acceleration of the robot successfully both inside and outside of the regions. For instance, compared to no integral and derivative predicates case, given in cyan in Fig. 2a, when region-based derivative constraint is applied, red in Fig. 2c, robot cannot accelerate enough to enter region C and go around instead. This is the reason why the only trajectory that enters the C is the cyan one by increasing its horizontal velocity to 1 \( m/s \) inside it. Finally, when there is no derivative predicate applied, the trajectory in magenta also avoids C since it has to slow down inside B to travel the minimum required distance. This makes the magenta trajectory a more aggressive one compared to the blue trajectory with both integral and derivative predicates inside B. The comparison
of these four scenarios in terms of the control effort and computation time are also presented in Table I.

Fig. 2: Comparisons of the blue trajectory that achieves the original specification (17) with the ones achieving the same specification except \( a \) the integral and derivative predicates (cyan) \( b \) the derivative predicates (magenta) \( c \) the integral predicates (red). All trajectories start from \( (0.5, 0.5) \) in.

<table>
<thead>
<tr>
<th>Table I: Results of the simulation scenarios in Fig. 2.</th>
<th>Int.&amp;Der. Pred. (blue)</th>
<th>Only Int. Predicates (magenta)</th>
<th>Only Der. Predicates (red)</th>
<th>None of Them (cyan)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Cost</td>
<td>6.5363</td>
<td>4.8253</td>
<td>4.0749</td>
<td>3.9930</td>
</tr>
<tr>
<td>Soln. Time</td>
<td>4.53</td>
<td>15.43</td>
<td>1.13</td>
<td>2.29</td>
</tr>
</tbody>
</table>

VI. CONCLUSION AND FUTURE WORK

In this study, we introduce integral and derivative predicates for Signal Temporal Logic. These new predicates can be used to define specifications regarding the cumulative effects and the rate of change in a signal. We show that the integral and derivative predicates can be encoded as mixed-integer linear constraints. Optimal trajectories satisfying the complex temporal logic specifications are then obtained by solving a mixed-integer linear program. A case study on the control of an autonomous robot in two-dimensional continuous space is included. We show how the new predicates can be used in the design of the mission specifications in a richer and more expressive way compared to the conventional STL. As a future direction, we plan to extend this work to multi-agent systems that are coupled through the specifications of their objectives and constraints (e.g., [21]–[23]).

REFERENCES


