Set-Invariance Methods for Spatiotemporally Constrained Systems

Adrian Wiltz and Dimos V. Dimarogonas

***Abstract*— In this paper, we provide a brief overview of our recent results on set-invariance methods for the systematic controller design for dynamic systems subject to spatiotemporal constraints. Our work includes the synthesis of value functions that characterize the system's dynamic capabilities with respect to a constraint, the synthesis of controllers that render time-varying sets forward invariant and we show how such results can be used to handle spatiotemporal logic tasks.**

I. Introduction

Spatiotemporal logic constraints are a rich class of constraint specifications that allow us to encode state and time constraints, and combine these with each other via logic operators. For example, a specification for a team of three mobile robots within this class could be as follows (see Fig. 1): Two out of the three robots shall meet at some time between 10 and 20s, and each of the robots shall move within 50s to their assigned blue box while avoiding collisions with other robots and any of the obstacles.

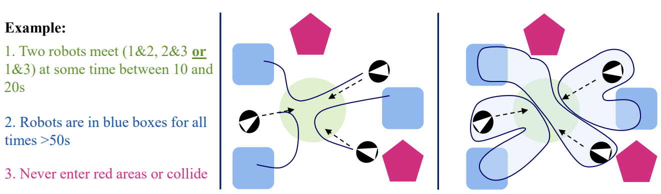

Fig. 1: Motivating example with schematic solutions sketched using trajectories and, alternatively, invariant sets.

There exist various methods to solve spatiotemporal logic problems. Most of them compute a trajectory (e.g. based on finite transition systems or RRT*) that complies with the constraint specification and is afterwards followed using a tracking controller. While yielding sometimes a type of optimality, the computation of feasible trajectories is generally time-consuming. This renders such approaches rather static and unresponsive to changes in the constraint specifications.

As an alternative approach, spatiotemporal logic constraints can be also encoded into forward invariant time-varying sets. While still requiring some planning effort, the degrees of freedom from the constraint specifications can, at least partly, be preserved and are not reduced to a single trajectory as in the first approach. This gives such approaches an increased flexibility to account for changes in the specifications online.

This work was supported by the ERC Consolidator Grant LEAFHOUND, the Swedish Research Council, and the Knut and Alice Wallenberg Foundation.

The authors are with the Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden {wiltz,dimos}@kth.se.

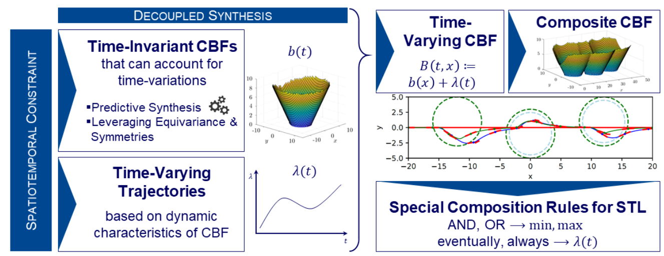

Fig. 2: Summary of the decoupled synthesis of time-varying CBFs. An example for the bicycle model is provided.

II. Background

In the sequel, we consider dynamic systems of the form

$$\dot{x} = f(x,u), \qquad x(0) = x_0 \tag{1}$$

with $x \in \mathbb{R}^n$ subject to spatiotemporal logic constraints, and input constraint $u \in \mathcal{U} \subseteq \mathbb{R}^m$; f is assumed to be Lipschitz-continuous and the solutions forward complete. An often employed formalism to express spatiotemporal constraints is Signal Temporal Logic (STL), which entails

- *state constraints* in terms of predicates, e.g. $p : h(x) \geqslant 0$;
- *temporal operators*, namely that a state constraint shall be *always* ($\mathcal{G}_{[a,b]}p$) or *eventually* ($\mathcal{F}_{[a,b]}p$) satisfied on a time interval $[a,b]$, or that a constraint p_1 shall be satisfied *until* another constraint p_2 eventually holds on a time interval $[a,b]$;
- *logic operators* as AND ($\wedge$), OR ($\vee$) and NOT ($\neg$).

For a more formal introduction of STL, we refer to [1]; an exemplary STL specification is given in Fig. 3 as ψ_0.

Inspired by [2], many works encode STL specifications into forward invariant sets, mostly using Control Barrier Functions (CBF). A CBF with respect to system (1) defined on a domain $\mathcal{D} \subseteq \mathbb{R}^n$ is a differentiable function $b : \mathcal{D} \to \mathbb{R}$ with

$$\sup_{u \in \mathcal{U}} \left\{ \frac{db}{dt}(x)\, f(x,u) \right\} \geqslant -\alpha(b(x)) \qquad \forall x \in \mathcal{D} \tag{2}$$

where α denotes an extended class $\mathcal{K}_e$ function. For encoding STL specifications, CBFs of the form

$$B(t,x) \coloneqq b(x) + \lambda(t) \tag{3}$$

are of particular interest. However, even if b itself constitutes a CBF, it is not guaranteed that the CBF property (2) is preserved when adding a time-varying function λ.

In the remainder, we briefly outline a systematic design process for constructing CBFs, before turning towards an extension of the framework in [2] that additionally allows to encode disjunctions (logic OR) based on non-smooth CBFs.

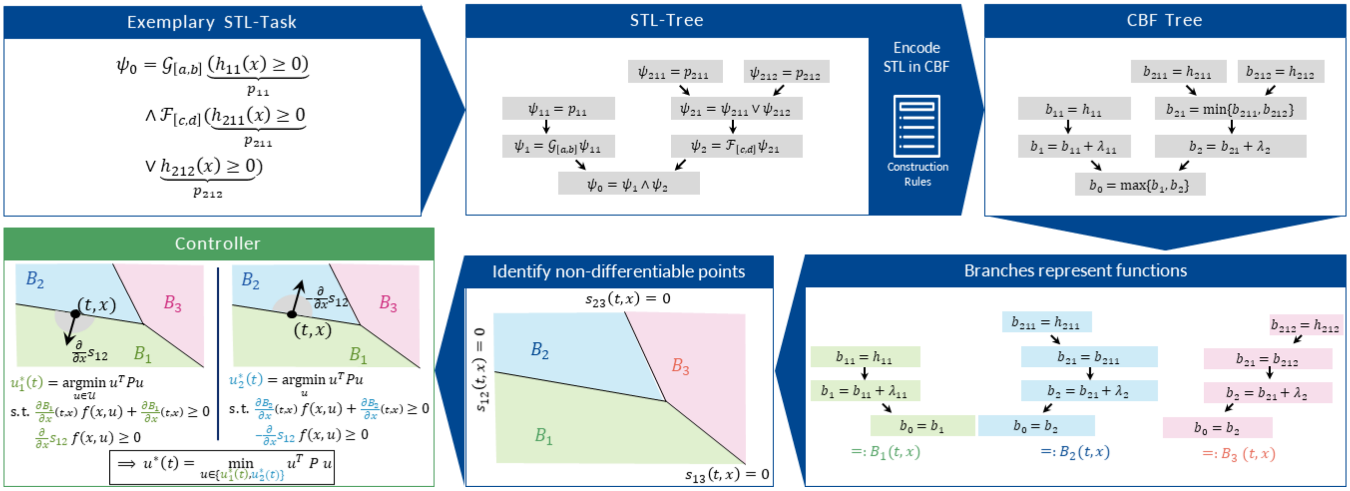

Fig. 3: Overview over the controller design: from a given STL specification to the controller.

III. Synthesis of Time-Varying CBFs

For the systematic construction of CBFs of the form (3), b must be chosen from a special class of CBFs. In particular, we require b to be a CBF that preserves CBF property (2) under the shift induced by the time-varying function λ. For this reason, we call such CBFs *shiftable CBFs*.

Definition 1 (Λ-shiftable CBF)**.** A continuously differentiable function $b : \mathbb{R}^n \to \mathbb{R}$ is called a Λ-shiftable CBF with respect to (1) for $\Lambda > 0$ if there exists an extended class $\mathcal{K}_e$ function α such that b satisfies (2) on a domain $\mathcal{C}_\Lambda \coloneqq \{x \,|\, b(x) \geqslant -\Lambda\}$.

Theorem 1. Let α be convex or concave, and let $\lambda : \mathbb{R}_{\geqslant 0} \to [0, \Lambda]$ satisfy

$$\frac{\partial \lambda}{\partial t}(t) \geqslant \alpha(-\lambda(t)).$$

Then, B is a CBF on the domain $\mathbb{R}_{\geqslant 0} \times \mathcal{C}_\Lambda$ with respect to system (1) augmented with time.

This result is a corollay of [3, Theorem 4] and allows for the decoupled synthesis of b and λ. This gives rise to an entire class of uniformly time-varying CBFs. In [4], a systematic method to synthesize shiftable CBFs is presented; a python implementation of the method is available. The method also yields an explicit characterization of the function α that takes a crucial role in the design of time-varying functions λ. By employing equivariances in the system dynamics, the computational effort for the CBF synthesis can be reduced as shown in [5]. A summary of our synthesis approach to time-varying CBFs can be found in Fig. 2. The synthesized CBFs can be then composed (under certain conditions) to larger CBFs. An example is depicted in the right half of Fig. 2 for two input-constrained bicycle models (a more and a less agile one) including simulation results for time-varying constraints. The example is taken from [4]. Special composition rules are required for STL specifications.

IV. Non-smooth CBFs for STL Including Disjunctions

The previous results on the synthesis of time-varying CBFs enable us to encode a rich fragment of STL comprising a broad variety of spatiotemporal logic constraints into CBFs. The considered specifications include, in contrast to earlier works such as [2], also disjunctions. This requires us to employ non-smooth CBFs. An overview over our proposed controller design as by [6] is given in Fig. 3. Starting with a given STL specification, we first derive its corresponding STL-tree, which is then translated to a CBF tree encoding the original STL specification. Thereby, a set of specific construction rules is employed. For instance, specifications employing the always ($\mathcal{G}$) or eventually ($\mathcal{F}$) operators are encoded into functions of the form $b(x) + \lambda(t)$. To ensure that these functions indeed constitute CBFs, we can employ the results from Theorem 1. By investigating each of the branches of the CBF-tree, we note that each of them represents a differentiable function. Only their composition via the minimum and the maximum operators renders the resulting function non-smooth. Thus, the non-differentiable points can be determined analytically by identifying those points, where the functions represented by the branches equal each other. The knowledge on the non-smooth points can be exploited to compute control inputs via Dini derivatives.

References

[1] O. Maler and D. Nickovic, “Monitoring temporal properties of continuous signals,” in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, Y. Lakhnech and S. Yovine, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 152–166.

[2] 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, 2019.

[3] A. Wiltz and D. V. Dimarogonas, “From time-invariant to uniformly time-varying control barrier functions: A constructive approach,” in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024.

[4] ——, “Predictive synthesis of control barrier functions and its application to time-varying constraints,” 2025, under review.

[5] ——, “Leveraging equivariances and symmetries in the control barrier function synthesis,” 2025, in preparation.

[6] ——, “Handling disjunctions in signal temporal logic based control through nonsmooth barrier functions,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 3237–3242.