Control Synthesis for Partial Differential Equations from Spatio-Temporal Specifications

Francisco Penedo[1], Harold Park[2] and Calin Belta[3]

***Abstract*— In this paper, we introduce a new boundary control synthesis problem with temporal logic specifications for a wide range of linear partial differential equations. We leverage the finite element method (FEM) to reduce the problem to a control problem for discrete-time linear systems. The specifications are formalized using an extension of signal temporal logic (STL), called Spatial-STL (S-STL). A conservative procedure to reformulate the specification into a regular STL formula as part of the FEM reduction is presented. A mixed-integer linear encoding is then used to synthesize the control inputs from a given allowed set. We illustrate the algorithm on one- and two-dimensional heat and wave propagation equations.**

I. Introduction

Partial differential equations (PDEs) model nearly all of the physical systems and processes of interest to scientists and engineers. Some well-known examples include the Navier-Stokes equation for fluid mechanics, the Maxwell equations for electromagnetics, the Schrödinger equation for quantum mechanics, the heat equation and the wave equation. The analysis of these and other PDEs has had a tremendous impact on society by enabling our understanding of thermal, electrical, fluidic and mechanical processes.

While a mature field, the study of PDEs is often approached through simulations in which approximate models obtained through spatio-temporal discretization techniques, such as the Finite Element Method (FEM) [1], are solved numerically. These methods, however, do not allow for rigorous guarantees that a system satisfies a complex specification. Other approaches generalizing classic Ordinary Differential Equations (ODEs) tools such as Lyapunov analysis and backstepping [2] do provide some guarantees and can be used to obtain boundary control strategies for a wide variety of PDEs. However, they are restricted to classical control objectives such as stabilization, cost optimization and trajectory tracking [3].

The formal statement of specifications and the development of analysis techniques that can guarantee their satisfaction by design has been the main focus of the formal methods field. During the past decades, many specification languages have been defined, such as Linear Temporal Logic (LTL) [4] and Signal Temporal Logic (STL) [5]. Originally, these logics have been applied to the study of finite systems, although more recently abstraction procedures have been developed to reduce problems with ODEs to finite models (for example through state space discretization [6] or mixed-integer linear program (MILP) encodings [7]). However, these techniques cannot be immediately applied to the analysis of PDEs due to the lack of spatial information in the formal language. This issue was first addressed in [8] in the context of spatially distributed systems, which can be viewed as PDEs in a discretized spatial domain. In their work, the authors view the system state as an image and define a formal language with explicit spatial information called Linear Superposition Logic (LSSL) based on quadtrees, a tree-based abstraction of the system. In [9], a variant of LSSL with quantitative semantics, Tree Spatial Superposition Logic (TSSL), is used for (steady-state) pattern synthesis in reaction diffusion systems, while in [10], a new formal language combining TSSL and STL, Spatial Temporal Logic (SpaTeL), allows the synthesis of dynamical patterns. The specification of patterns in these logics is difficult, however, and machine learning techniques are needed in order to obtain logic formulas corresponding to a set of examples of the desired pattern, which is not always available or desirable for some applications where the system behavior must be specified a priori. More recently, a new spatio-temporal logic called STREL was introduced in [11] to specify the behavior of mobile and spatially distributed Cyber-Physical Systems.

In this work, we extend STL to allow specifications over the solutions of PDEs in such a way that both temporal and spatial properties can be formally stated in a user-friendly manner. We call the resulting language Spatial-STL or S-STL. Then, we formulate and solve the problem of synthesizing a boundary control input for a PDE such that a property defined in S-STL is satisfied. To the best of our knowledge, the boundary control synthesis of PDEs from temporal constraints has not been explored before.

Our approach uses the FEM to approximate the trajectory of the PDE by converting the PDE, as is standard with the FEM [1], to a system of ODEs. The FEM is a well-established numerical method to obtain approximate numerical solutions to PDEs, and is particularly well-suited to problems with complicated geometries where exact solutions cannot be found. The state space of the resulting ODE system represents the field values of the PDE at discrete locations, called nodes, obtained after discretizing the domain. In the process of approximating the PDE trajectory by the FEM model and then discretizing it both in space (considering only the values at the nodes) and time, we define a conservative reformulation of the specification that

This work was partially supported by the National Science Foundation under grants CNS-1446607 and CMMI-1400167.

[1]F. Penedo is with the Division of Systems Engineering, Boston University `franp@bu.edu`

[2]H. Park is with the Department of Mechanical Engineering, Boston University `parkhs@bu.edu`

[3]C. Belta is with the Division of Systems Engineering, Boston University `cbelta@bu.edu`

follows the same approximation and discretization steps. In order to account for the approximation and discretization errors, we introduce correction terms in the formula in such a way that all trajectories satisfying the corrected formula also satisfy the original one. This procedure is similar to the syntactical re-writing rules for MTL proposed in [12] in order to automatically infer properties satisfied in a derived system model (via simplification or implementation, for example) from properties satisfied in the original model. Finally, we encode the resulting control problem into a MILP following previously established methods. If the control inputs are given, it is trivial to adapt the method to solve a verification problem instead.

Our method requires the PDE to be linear, which usually constrains the physical system to linear material behavior. However, this assumption is not overly restrictive as most engineering systems are designed to operate within the linear material response regime. We also assume linear boundary conditions with respect to the control input.

II. Preliminaries

A. Finite Element Method

In this section we provide a summary of the Finite Element Method (FEM) [1]. For simplicity, we present the method applied to a heat equation over a one-dimensional domain, although the same technique can be applied to other PDEs.

Let $\Omega = (0, L) \subset \mathbb{R}$ be an open interval representing the interior of a one-dimensional rod of length L; $\rho, c, \kappa > 0$ constants denoting density, capacity and conductivity of the rod's material respectively; $g = (g_0, g_L) \in \mathbb{R}^2$ the (time-independent) boundary conditions at each end of the rod; and $u_0 : \Omega \to \mathbb{R}$ an initial value for the temperature on the rod. The evolution of the temperature at each point in the rod can be described by a function $u : \bar{\Omega} \times [0, T] \to \mathbb{R}$, where $T > 0$ denotes the final time and can be infinity and $\bar{\Omega}$ is the closure of Ω (which we call the spatial domain), such that the following initial boundary value problem (IBVP) is satisfied:

$$\Sigma^H(u_0, g) \quad \begin{cases} \rho c \dfrac{\partial u}{\partial t} - \kappa \dfrac{\partial^2 u}{\partial x^2} = 0, \text{on } \Omega \times (0, T)\,, \\ u(0, t) = g_0, \forall t \in (0, T)\,, \\ u(L, t) = g_L, \forall t \in (0, T)\,, \\ u(x, 0) = u_0(x), \forall x \in \Omega\,. \end{cases} \tag{1}$$

An important aspect of the FEM is the creation of a weak formulation of the governing PDEs in (1). Its purpose is to reduce the second order spatial derivative of u in (1) to first derivatives so that low order (in this case, linear) polynomials can be used to approximate the field value u. Let D be the set of sufficiently smooth real-valued functions on $\bar{\Omega} \times (0, T)$ such that all $u \in D$ satisfy $u(0, t) = g_0, u(L, t) = g_L, \forall t \in (0, T)$, and V a similar set of time independent functions such that all $w \in V$ satisfy $w(0) = w(L) = 0$. The problem is to find $u \in D$ such that for all $w \in V$,

$$\begin{aligned} \int_\Omega \frac{\partial w}{\partial x} \kappa \frac{\partial u}{\partial x} d\Omega + \int_\Omega w \rho c \frac{\partial u}{\partial t} d\Omega = 0\,, \\ \int_\Omega w \rho c u(\cdot, 0) d\Omega = \int_\Omega w \rho c u_0 d\Omega\,. \end{aligned} \tag{2}$$

We now obtain an approximate solution to the weak formulation by considering (2) with u and w in subspaces of D and V. Let $\{x_i\}_{i=0}^{n+1}$, where $x_0 = 0, x_{n+1} = L, x_i \in \Omega, i = 1, ..., n$, be a partition of $\bar{\Omega}$. Let $d_i(t), i = 0, ..., n+1$ represent the temperature of the rod at node x_i, with $d_0(t) = g_0, d_{n+1} = g_L$ and let $d = (d_1, ..., d_n)' \in \mathbb{R}^n$. We define the following linear node shape function matrices for $i = 0, ..., n+1$:

$$N_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & i > 0, x \in [x_{i-1}, x_i]\,, \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & i < n+1, x \in [x_i, x_{i+1}]\,, \end{cases} \tag{3}$$

which results in the following linear interpolation of the field variable u in terms of its nodal values d:

$$u^d(x, t) = \sum_{i=0}^{n+1} N_i(x) d_i(t)\,. \tag{4}$$

Consider the subspaces $D^h \subset D$ and $V^h \subset V$ of linear interpolations defined above and time-invariant interpolations respectively. It can be shown that $u^d(x, t)$ is a solution of the weak formulation over the sets D^h and V^h, where d evolves according to the following linear system:

$$\Sigma^H_{FEM}(u_0, g) \quad \begin{cases} M\dot{d} + Kd = F(g)\,, \\ d_i(0) = u_0(x_i), \quad i = 1, ..., n\,. \end{cases} \tag{5}$$

In the above, M, K and F are the mass, stiffness and external force matrices respectively, whose specific form depend on the partition and the parameters of the PDE. In general, M can be considered diagonal and K is a banded matrix, in this specific PDE having bandwidth 3. In general, we will denote a PDE by $\Sigma(...)$ and we will call $\Sigma_{FEM}(...)$ the FEM system corresponding to a PDE system $\Sigma(...)$.

B. Signal Temporal Logic

A detailed definition of STL can be found in [5], so we just provide here necessary notation and examples. An STL formula ϕ is constructed from predicates over signals $s : \mathbb{R}_{\geq 0} \to \mathbb{R}$ of the form $\mu \equiv \mu(s) > 0$, Boolean operators $\neg$ (negation), $\wedge$ (conjunction) and $\vee$ (disjunction), and temporal operators $\mathbf{U}_{[a,b]}$ (until between a and b), $\mathbf{F}_{[a,b]}$ (eventually between a and b, equivalent to $\top \mathbf{U}_{[a,b]}$) and $\mathbf{G}_{[a,b]}$ (always between a and b, equivalent to $\neg \mathbf{F}_{[a,b]} \neg$). We write $s[t] \models \phi$ if signal s satisfies ϕ at time t and $s \models \phi$ if $s[0] \models \phi$. We also consider quantitative semantics given by the robustness function $r(\phi, s, t)$, which gives a measure of distance to satisfaction of a signal to a formula and can be computed recursively using $\min$ and $\max$ operators. The relationship between the two semantics is given by $s[t] \models \phi \iff r(\phi, s, t) > 0$.

As an example, consider the temperature at the end of a rod to be $s(t) = 2t$, and the specification $\phi = \mathbf{G}_{[1,2]} s > 0.5$.

We have $s \models \phi$, since $s(t) > 0.5, \forall t \in [1, 2]$ and robustness $r(\phi, s, 0) = 0.5$ since $\min_{t\in[1,2]}(s(t) - 0.5) = 0.5$.

III. Signal Temporal Logic for PDEs

In order to define specifications over the trajectories of PDEs, we extend regular STL using the following set of predicates: let Λ be a set of predicates over a set $\bar{\Omega}$, where each predicate $\lambda \in \Lambda$ is defined as a tuple $(Q_\lambda, X_\lambda, \mu_\lambda, D_\lambda)$, and represented using the syntax $Q_\lambda x \in X_\lambda : D_\lambda u(x) - \mu_\lambda(x) > 0$, where:

- $Q_\lambda \in \{\forall, \exists\}$ is the spatial quantifier,
- $X_\lambda \subseteq \bar{\Omega}$ is the spatial domain of the predicate and we require it to be a closed set,
- $\mu_\lambda : X_\lambda \to \mathbb{R}$ is a continuous and differentiable function representing the reference profile, and
- $D_\lambda \in \{\frac{d^i}{dx^i}\}_{i=0,1,\dots}$ is a differential operator, with $\frac{d^0}{dx^0} = id$ the identity.

We consider STL formulas with the usual syntax and semantics over the set of predicates Λ, which we call Spatial-STL (S-STL). The satisfaction of a predicate with respect to a continuous-time signal $u : \bar{\Omega} \times [0, T] \to \mathbb{R}$ at time $t \in \mathbb{R}$ is defined as $u[t] \models \lambda \iff D_\lambda u(x, t) - \mu_\lambda(x) > 0$ for all $x \in X_\lambda$ if $Q_\lambda = \forall$ or for some $x \in X_\lambda$ otherwise. We define the quantitative semantics as before with $r(\lambda, u, t) = \min_{x\in X_\lambda}(D_\lambda u(x, t) - \mu_\lambda(x))$ if $Q_\lambda = \forall$ and $r(\lambda, u, t) = \max_{x\in X_\lambda}(D_\lambda u(x, t) - \mu_\lambda(x))$ otherwise. For convenience, we define syntactic sugar for predicates with the opposite inequality using the following equivalence: $\forall x \in X : Du(x) - \mu \leq 0 \equiv \neg\exists x \in X : Du(x) - \mu > 0$, and similarly for an existential predicate.

Example 1: Consider a metallic rod of $100\,\mathrm{mm}$. The temperature at one end of the rod is fixed at $300\,\mathrm{K}$, while a heat source is applied to the other end. The temperature of the rod follows a heat equation similar to (1). We want the temperature distribution of the rod to be within $3\,\mathrm{K}$ of the linear profile $\mu(x) = \frac{x}{4} + 300$ at all times between 4 and 5 seconds in the section between 30 and 60 mm. Furthermore, the temperature should never exceed $345\,\mathrm{K}$ at the point where the heat source is applied ($x = 100$). We can formulate such a specification using the following S-STL formula:

$$\begin{aligned}\phi_{ex} = &\mathbf{G}_{[4,5]}\big((\forall x \in [30, 60] : u(x) - (\frac{x}{4} + 303) < 0)\wedge \\ &(\forall x \in [30, 60] : u(x) - (\frac{x}{4} + 297) > 0)\big)\wedge \\ &\mathbf{G}_{[0,5]}(\forall x \in [100, 100] : u(x) - 345 < 0)\,.\end{aligned} \tag{6}$$

Throughout this paper, we will use the notation $S(\dots) \models \psi$ to denote that the trajectory of system S (either a PDE system or its corresponding FEM system) with initial value s_0 and boundary conditions g satisfies formula ψ (either in regular STL or in S-STL).

IV. Problem Formulation and Approach

Let $\partial\Omega$ be the boundary of Ω. We consider $\partial\Omega$ to be partitioned in four regions $\{\partial\Omega_d, \partial\Omega_n, \partial\Omega_D, \partial\Omega_N\}$. Let $u(x, t) : \bar{\Omega} \times [0, T) \to \mathbb{R}^n$ be the state of a system evolving according to the IBVP $\Sigma(u_0, g_d, g_n, v_D, v_N)$:

$$\Sigma(\dots)\begin{cases} f(x, t, u, \frac{\partial u}{\partial t}, \frac{\partial u}{\partial x_i}, \dots) = 0, \text{on } \Omega \times (0, T)\,, \\ u(x, t) = g_d(x, t), \forall x \in \partial\Omega_d, \forall t \in (0, T)\,, \\ \frac{\partial u}{\partial n}(x, t) = g_n(x, t), \forall x \in \partial\Omega_n, \forall t \in (0, T)\,, \\ u(x, t) = v_D(x, t), \forall x \in \partial\Omega_D, \forall t \in (0, T)\,, \\ \frac{\partial u}{\partial n}(x, t) = v_N(x, t), \forall x \in \partial\Omega_N, \forall t \in (0, T)\,. \end{cases} \tag{7}$$

In the above, f is a function of space, time, state and all state derivatives, n is the normal vector to $\partial\Omega$, g_d and g_n are the Dirichlet and Neumann prescribed boundary conditions respectively, and v_D and v_N are the Dirichlet and Neumann boundary control inputs respectively. We will use Σ to refer to the system given by (7), including the partition of $\partial\Omega$, with unspecified initial value and boundary conditions.

Problem 1 (Boundary Control Synthesis Problem): Given a PDE Σ, an initial value u_0, prescribed boundary conditions g_d and g_n, an S-STL formula ψ over a set of predicates Λ, and admissible control sets V_D and V_N, synthesize control inputs $v_D \in V_D$ and $v_N \in V_N$ such that the trajectory of $\Sigma(u_0, g_d, g_n, v_D, v_N)$ satisfies ψ.

In the above formulation, the admissible control inputs are given in their most general form as sets of allowed control functions (note that from (7), $v_D : \partial\Omega_D \times (0, T) \to \mathbb{R}^n$ and $v_N : \partial\Omega_N \times (0, T) \to \mathbb{R}^n$). In practice, they are described in terms of control inputs constraints, such as $v_D(x, t) < 100, \forall x \in \partial\Omega_D, \forall t \in (0, T)$, as well as assumptions on the form of the function, such as piecewise affine in t and x. We also make the following assumptions: the function f in (7) is linear, and only a finite number of derivatives of u appear; and the sets V_D and V_N are described as polytopes in the parameters of a linear parameterization of the control inputs.

We solve Problem 1 by reformulating it into a synthesis problem for a discrete-time linear system instead. This is done in three steps, described in Sec. V, by considering the FEM approximation to the real solution, a spatial discretization and a temporal discretization. Then, we encode the resulting linear system and specification in a MILP and solve for the control inputs that maximize the robustness with respect to the discretized specification, which is described in Sec. VI. As a biproduct of our approach, we can formulate and solve a verification problem where the objective is to check the satisfaction of the system against the specification for all admissible control inputs.

V. Discretization of STL for PDEs

In this section we define a series of reformulations of an S-STL formula over Λ such that satisfaction of each successive reformulation guarantees that of the previous one. A diagram showing the hierarchy of the reformulations is shown in Fig. 1. The theory developed in this section is stated for a general 1D PDE for ease of notation, although it applies to higher dimensional PDEs with minimal changes.

We first reformulate the specification ψ so that we can work with the approximation given by Σ_{FEM}. Recall that

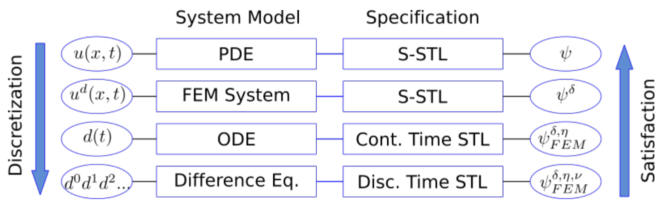


Fig. 1: Summary of our approach. The PDE model is discretized in three steps to obtain a set of difference equations. The S-STL specification is rewritten following the same steps so that satisfaction in the discretized model is preserved.

we denote as $u(x,t)$ the trajectory of Σ^1 and $u^d(x,t)$ the piecewise linear approximation obtained by interpolating the trajectory of Σ_{FEM}. In the following, we assume the partition $\{x_i\}_{i=1}^m$ is proposition preserving with respect to the set of regions $\{X_i\}_{i=1}^m$[2]. Suppose we are given an a priori bound, $\epsilon_i(x,t)$, for the pointwise difference between the ith derivative of the trajectory and the FEM approximation, i.e., $|\frac{\partial d^i}{\partial x^i}u(x,t) - \frac{\partial d^i}{\partial x^i}u^d(x,t)| \le \epsilon^i(x,t)$.

Definition 1: Let $\lambda \in \Lambda$ be a predicate and let $\delta : \bar{\Omega} \to \mathbb{R}$ be a continuous and differentiable function. The perturbation of λ by δ, λ^δ, is the predicate given by the tuple $(Q_\lambda, X_\lambda, \mu_\lambda^\delta, D_\lambda)$, where $\mu_\lambda^\delta(x,u) = \mu_\lambda(x,u) + \delta(x)$.

Definition 2: Let ψ be an S-STL formula over Λ in negation normal form and let $\delta = \{\delta_i\}_{i=0,1,\dots}$ be a set of continuous functions $\delta_i : \bar{\Omega} \to \mathbb{R}$. The conservative correction of ψ by δ, ψ^δ, is an S-STL formula in negation normal form obtained by substituting every predicate λ that appears in ψ in the following way:

- If λ is not preceded by a negation operator and $D_\lambda = \frac{d^i}{dx^i}$, then it is substituted by λ^{δ_i}.
- Otherwise, it is substituted by $\lambda^{-\delta_i}$

We will also use Λ^δ to refer to the set of all δ and $-\delta$ corrections of predicates in Λ.

Theorem 1: If $u^d \models \psi^\delta$, with $\delta_i(x) = \max_t \epsilon_i(x,t)$, then $u \models \psi$.

Proof: We only need to consider a predicate λ and its negation. Let i be the order of the derivative D_λ. We have

$$|\frac{\partial^i}{\partial x^i}u(x,t) - \mu_\lambda(x) - \frac{\partial^i}{\partial x^i}u^d(x,t) + \mu_\lambda(x)| = \\ |\frac{\partial^i}{\partial x^i}u(x,t) - \frac{\partial^i}{\partial x^i}u^d(x,t)| \le \delta_i(x)\,. \quad (8)$$

Thus, $\frac{\partial^i}{\partial x^i}u(x,t) - \mu_\lambda(x) \ge \frac{\partial^i}{\partial x^i}u^d(x,t) - \mu_\lambda(x) - \delta_i(x)$, which proves the result for $\psi = \lambda$. For $\psi = \neg\lambda$, note that satisfaction is equivalent to $\frac{\partial^i}{\partial x^i}u(x,t) - \mu_\lambda(x) < 0$ for x quantified opposite to Q_λ. Finally, from (8) we have $\frac{\partial^i}{\partial x^i}u(x,t) - \mu_\lambda(x) \le \frac{\partial^i}{\partial x^i}u^d(x,t) - \mu_\lambda(x) + \delta_i(x)$. ∎

Example 2: Assume we obtain a bound $\delta_0(x) = 0.25$ for the system Σ^H described in Ex. 1 and its FEM approximation. Consider the same specification ϕ_{ex} defined in the example. The perturbed specification, ϕ_{ex}^δ is then:

$$\begin{aligned}\phi_{ex}^\delta = &\mathbf{G}_{[4,5]}\big(\\ &(\forall x \in [30,60] : u(x) - (\frac{x}{4} + 303 - 0.25) < 0) \wedge \\ &(\forall x \in [30,60] : u(x) - (\frac{x}{4} + 297 + 0.25) > 0)\big) \wedge \\ &\mathbf{G}_{[0,5]}(\forall x \in [100,100] : u(x) - 345 - 0.25 < 0)\,.\end{aligned} \quad (9)$$

If the FEM approximation to Σ^H satisfies ϕ_{ex}^δ, we can conclude Σ^H satisfies ϕ_{ex}.

Theorem 1 gives us a way to conservatively solve Problem 1 using the approximation obtained from Σ_{FEM}. However, we still need to deal with continuous functions in continuous time. Our next step will be to reformulate the specification ψ^δ into an STL formula that can be checked against the trajectory of Σ_{FEM}, i.e., $d(t)$.

Let $\Lambda_{FEM}^\delta = \{\alpha_{\lambda,e} | \lambda \in \Lambda^\delta, e \in E_\lambda\} \cup \{\beta_{\lambda,j} | \lambda \in \Lambda^\delta, j \in J_\lambda, D_\lambda = id\}$, where $E_\lambda = \{e | [x_e, x_{e+1}] \subseteq X_\lambda\}, J_\lambda = \{j | x_j \in X_\lambda, [x_{j-1}, x_j] \not\subseteq X_\lambda, [x_j, x_{j+1}] \not\subseteq X_\lambda\}$, and satisfaction of $\alpha_{\lambda,e}$ and $\beta_{\lambda,j}$ by a continuous-time signal $d : [0,T] \to \mathbb{R}^n$ is defined in the following way:

$$d[t] \models \alpha_{\lambda,e} \iff D_\lambda u^d(x_e^m, t) - \mu_\lambda(x_e^m) > 0\,, \quad (10)$$

$$d[t] \models \beta_{\lambda,j} \iff d_j(t) - \mu_\lambda(x_j) > 0\,, \quad (11)$$

where $x_e^m = \frac{x_e + x_{e+1}}{2}$ is the midpoint of element e. Note that $D_\lambda u^d(x_e^m, t)$ is a function of $d(t)$. The α predicates simplify a predicate λ so that only a representative value (the midpoint) is checked at each element. The β predicates are needed in those cases where X_λ is a single point. The robustness degree for these predicates is defined as in Sec. II-B. Note that this set of predicates includes the perturbations defined in Def. 2. We define perturbations of predicates in Λ_{FEM}^δ by a real number k in a manner analogous to Def. 1, and we denote it as α^k.

Definition 3: The STL formula over Λ_{FEM}^δ, $\psi_{FEM}^{\delta,\eta}$, corresponding to an S-STL formula in negation normal form, ψ^δ, over Λ^δ, with $\eta = \{\eta_i\}_{i=0,1,\dots}, \eta_i : \bar{\Omega} \to \mathbb{R}$, is a formula obtained by substituting every predicate λ in ψ^δ by the formula $\bigoplus_{e \in E_\lambda} \gamma_{\lambda,e} \oplus \bigoplus_{j \in J_\lambda} \beta_{\lambda,j}$, where $\oplus = \wedge$ if $Q_i = \forall$ and $\oplus = \vee$ otherwise, and $\gamma_{\lambda,e}$ is defined as the following STL formula:

- If λ is not preceded by a negation operator, then $\gamma_{\lambda,e} = \alpha_{\lambda,e}^{-k_e^\lambda}$.
- Otherwise, $\gamma_{\lambda,e} = \alpha_{\lambda,e}^{k_e^\lambda}$.

In both cases, for $D_\lambda = \frac{d^i}{dx^i}$ and $l_e = x_{e+1} - x_e$,

$$k_e^\lambda = \frac{l_e}{2}\left(\max_{c \in [x_e, x_{e+1}]} |\mu'_\lambda(c)| + \max_{c \in [x_e, x_{e+1}]} \eta_i(c)\right)\,. \quad (12)$$

Theorem 2: If $d \models \psi_{FEM}^{\delta,\eta}$, with $\eta_i(x) \ge \max_t |\frac{\partial^{i+1} u^d}{\partial x^{i+1}}(x,t)|$, then $u^d \models \psi^\delta$.

Proof: We only need to consider a predicate λ and its negation. We assume $Q_\lambda = \forall$, the other case is similar. Let $X_\lambda = [x_a, x_b]$ and i be the order of the derivative D_λ. First note that satisfying λ is equivalent to satisfying all predicates of the set $\{(Q_\lambda, [x_e, x_{e+1}], \mu_\lambda, D_\lambda) | e \in E_\lambda\}$. For any $e \in$

[1]In what follows we use Σ to refer to $\Sigma(u_0, g_d, g_n, v_D, v_N)$.

[2]I.e., the X_i sets are unions of the regions defined by the partition. This is well defined since the X_i sets are closed and there are a finite number of them. In higher dimensions, we also require the X_i to be are polytopes.

E_λ, let $x^m = \frac{x_e + x_{e+1}}{2}$, $h = \frac{x_{e+1} - x_e}{2}$. For $x \in [x_e, x_{e+1}]$ we have:

$$
\begin{aligned}
&|\frac{\partial^i}{\partial x^i} u^d(x,t) - \mu_\lambda(x) - \frac{\partial^i}{\partial x^i} u^d(x^m,t) + \mu_\lambda(x^m)| \leq \\
&h \max_{c \in [x_e, x_{e+1}]} |\mu'_\lambda(c) + \frac{\partial^{i+1} u^d}{\partial x^{i+1}}(c,t)| \leq \\
&h \left(\max_{c \in [x_e, x_{e+1}]} |\mu'_\lambda(c)| + \max_{c \in [x_e, x_{e+1}]} |\frac{\partial^{i+1} u^d}{\partial x^{i+1}}(c,t)| \right) \leq \\
&h \left(\max_{c \in [x_e, x_{e+1}]} |\mu'_\lambda(c)| + \max_{c \in [x_e, x_{e+1}]} \eta_i(c) \right) = K_e \,.
\end{aligned}
\tag{13}
$$

Then,

$$
\begin{aligned}
\frac{\partial^i}{\partial x^i} u^d(x,t) - \mu_\lambda(x) \geq \frac{\partial^i}{\partial x^i} u^d(x^m,t) - \mu_\lambda(x^m) - K_e = \\
D_\lambda u^d(x^m,t) - \mu_\lambda(x^m) - K_e \,,
\end{aligned}
\tag{14}
$$

so $d \models \alpha_{\lambda,e}^{-K_e}$ implies $u^d \models (Q_\lambda, [x_e, x_{e+1}], \mu_\lambda, D_\lambda)$ and the theorem holds for $\psi^\delta = \lambda$. To prove it for the negated predicate, we can follow an argument similar to the one in the proof of Thm. 1. ∎

Example 3: Continuing with Ex. 2, assume we obtained the FEM approximation using the partition $\{10i | i \in 0, ..., 10\}$ and we found the bound $\eta_0(x) = 0.15$. The perturbed STL specification corresponding to ψ^δ is

$$
\begin{aligned}
\phi_{ex,FEM}^{\delta,\eta} = \mathbf{G}_{[4,5]} \big(\\
&(y_4 - 311.75 - 0.25 - 5 * (0.25 + 0.15) < 0) \wedge \\
&(y_5 - 314.25 - 0.25 - 5 * (0.25 + 0.15) < 0) \wedge \\
&(y_6 - 316.75 - 0.25 - 5 * (0.25 + 0.15) < 0) \wedge \\
&(y_4 - 305.75 - 0.25 - 5 * (0.25 + 0.15) > 0) \wedge \\
&(y_5 - 308.25 - 0.25 - 5 * (0.25 + 0.15) > 0) \wedge \\
&(y_6 - 310.75 - 0.25 - 5 * (0.25 + 0.15) > 0) \big) \wedge \\
&\mathbf{G}_{[0,5]} \big(d_{11} - 345 < 0 \big) \,,
\end{aligned}
\tag{15}
$$

where $y_e = u^d(x_e^m) = \frac{d_e + d_{e+1}}{2}$. If we prove satisfaction of $\phi_{ex,FEM}^{\delta,\eta}$ by Σ_{FEM}^H, we can conclude the interpolation satisfies ϕ_{ex}^δ.

Finally, we reformulate the specification into an STL formula with discrete time semantics that can be checked against the trajectory of a time discretization of Σ_{FEM} with time interval $\Delta t \in \mathbb{R}_{>0}$. There are several options at this point: the simplest one is to consider Σ_{FEM} as a first order linear system (augmenting the state space if needed) and define the time discretization as the following difference equation:

$$
\Sigma_{FEM}^{\Delta t}(d(0), g) \quad \begin{cases} \tilde{d}^{k+1} = \tilde{A} \tilde{d}^k + \tilde{b}(g) \,, \\ \tilde{d}^0 = d(0) \,, \end{cases}
\tag{16}
$$

where $\tilde{A} = e^{A \Delta t}$ and $\tilde{b}(g) = -e^{A \Delta t} A^{-1} (e^{-A \Delta t} - I) b(g)$. Note that, in theory, $d(k \Delta t) = \tilde{d}^k, \forall k \in \mathbb{N}$. However, in practice one needs to numerically compute the exponential matrices in $\tilde{A}$ and $\tilde{b}$, which introduces an approximation error difficult to control. As an alternative, we can use any numerical integration algorithm with fixed time step appropriate to the specific PDE system under study, several of which have been thoroughly analyzed in the FEM literature, such as the Newmark family. Similar to the FEM approximation, assume we have a bound on the approximation error of the integration algorithm, $\max_k |d_j(k\Delta t) - \tilde{d}_j^k| \leq \epsilon_j^d, i = 1, 2, ...$

We define a conservative correction of $\psi_{FEM}^{\delta,\eta}$ by a real vector $\nu = (\nu^y, \nu^d)$, $\psi_{FEM}^{\delta,\eta,\nu}$, in a similar way to Def. 3, noting that, for $D_\lambda = \frac{\partial^i}{\partial x^i}$, the predicate $\gamma_{\lambda,e}$ is perturbed using the constant $l_{i,e}^y = \Delta t \nu_{i,e}^y$ and the predicate $\beta_{\lambda,j}$ is perturbed using the constant $l_j^d = \Delta t \nu_j^d$. We abuse the notation for STL formulas so that we can consider satisfaction of the discrete time signal $\tilde{d}$. In particular, $\tilde{d}[t] \models \mu \iff \mu(\tilde{d}^{\lfloor t/\Delta t \rfloor}) > 0$.

Theorem 3: If $\tilde{d} \models \psi_{FEM}^{\delta,\eta,\nu}$, with $\nu_j^d \geq \max_t |\dot{d}_j(t)| + \epsilon_j^d$ and $\nu_{i,e}^y \geq \max_t |\frac{d}{dt} D^i u^d(x_e^m, t)| + D^i u^{\epsilon^d}(x_e^m)$, then $d \models \psi_{FEM}^{\delta,\eta}$.

Proof: Similar to Thm. 2. ∎

Example 4: Continuing with Ex. 3, assume we discretize $\Sigma_{FEM}^H(u_0, g)$ with perfect accuracy using the timestep $\Delta t = 0.005$ and let $\nu = (0.5, 0.5, ...)$. The final corrected specification is

$$
\begin{aligned}
&\phi_{ex,FEM}^{\delta,\eta,\nu} = \mathbf{G}_{[4,5]} \big(\\
&(y_4 - 311.75 - 0.25 - 5 * (0.25 + 0.15) + 0.5 * 0.005 < 0) \wedge \\
&(y_5 - 314.25 - 0.25 - 5 * (0.25 + 0.15) + 0.5 * 0.005 < 0) \wedge \\
&(y_6 - 316.75 - 0.25 - 5 * (0.25 + 0.15) + 0.5 * 0.005 < 0) \wedge \\
&(y_4 - 305.75 - 0.25 - 5 * (0.25 + 0.15) + 0.5 * 0.005 > 0) \wedge \\
&(y_5 - 308.25 - 0.25 - 5 * (0.25 + 0.15) + 0.5 * 0.005 > 0) \wedge \\
&(y_6 - 310.75 - 0.25 - 5 * (0.25 + 0.15) + 0.5 * 0.005 > 0) \big) \wedge \\
&\mathbf{G}_{[0,5]} \big(d_{11} - 345 + 0.5 * 0.005 < 0 \big) \,,
\end{aligned}
\tag{17}
$$

We can guarantee the satisfaction of $\phi_{ex,FEM}^{\delta,\eta}$ by Σ_{FEM}^H if $\Sigma_{FEM}^{H,\Delta t}$ satisfies $\phi_{ex,FEM}^{\delta,\eta,\nu}$.

The main result in this work is a corollary of the previous theorems, which allows us to solve Problem 1 by solving a control problem for discrete-time linear systems with regular STL constraints.

Theorem 4: If $\Sigma_{FEM}^{\Delta t} \models \psi_{FEM}^{\delta,\eta,\nu}$, with δ, η and ν defined as in Thms. 1 to 3 and $d_i^0 = u_0(x_i), i = 1, ..., n$, with u_0 an initial value for Σ, then $\Sigma \models \psi$.

We can also obtain a bound for the robustness of the trajectory of Σ with respect to the original specification by making the following observation:

Theorem 5: If u, u^d, d and $\tilde{d}$ are trajectories of Σ, interpolation of Σ_{FEM}, Σ_{FEM} and $\Sigma_{FEM}^{\Delta t}$, respectively, and ψ is an S-STL formula over Λ, then the following inequality holds:

$$
r(\psi, u, t) \geq r(\psi^\delta, u^d, t) \geq r(\psi_{FEM}^{\delta,\eta}, d, t) \geq r(\psi_{FEM}^{\delta,\eta,\nu}, \tilde{d}, t) \,.
\tag{18}
$$

Proof: It follows from the proofs of Thms. 1 to 3. ∎

VI. MILP Formulation of Control Synthesis

In this section, we solve Problem 1 by formulating an optimization problem using the corrected STL specification defined in the previous section. Our formulation is equivalent to an MILP, which we solve using an off-the-shelf solver such as Gurobi [13]. The optimization problem takes the following form:

$$r_m = \max \ r(\psi^{\delta,\eta,\nu}_{FEM}, \tilde{d}, 0) \quad \text{s.t. } (16), v_D \in V_D, v_N \in V_N \,. \tag{19}$$

In the above, (16) should be substituted by the appropriate difference equations resulting from the chosen ODE integration algorithm. After solving (19), if $r_m > 0$, then ψ is satisfied by the controlled system using as control inputs the optimal solution for v_D and v_N. This optimization problem is clearly non-convex, due to the $\max$ and $\min$ operators present in the definition of robustness, and the objective function is non-differentiable. However, we can apply the technique described in [14] to represent robustness as mixed-integer linear constraints. On the other hand, the system dynamics are linear and our assumption on the shape of the admissible control sets V_D and V_N implies that they can be encoded as linear constraints. Therefore, (19) is a MILP. Also note that by using the robustness degree as cost function, (19) is always feasible and a control input will be produced even if it does not correspond to a satisfying trajectory. In this case, r_m would be negative and we can think of the resulting optimal values for v_D and v_N as best effort inputs.

The computational time needed to solve a MILP grows exponentially with the number of binary variables. In our encoding, we introduce one binary variable for each argument to a $\min$ or $\max$ function. Thus, the number of binary variables is proportional to the length of time intervals (in the discrete sense), the number of boolean connectives in the original formula ψ and the length of the discretized predicates obtained in Def. 3. In terms of parameters of the problem and solution, the length of the discrete-time intervals is proportional to the length of the time intervals in ψ and inversely proportional to Δt; regarding the spatial discretization of the predicates, its length is proportional to the size of the spatial domains and inversely proportional to the size of the partition (i.e., the distance between two nodes in the partition).

VII. Examples

In this section we solve some instances of Problem 1 for a heat equation as well as an elastic wave propagation system and discuss the conservativeness of our approach as well as the computational performance. We implemented our framework using Python 2.7 and Gurobi 7.0 as our MILP solver. We ran our implementation in an Intel Core i7 at 2.4GHz and 16GB RAM.

A. Heat Equation

We start by solving a control synthesis problem for the system and specification described in Ex. 1. We assume the rod is made of two different materials: the section from 30 to 60 mm is made of a material with parameters $E_a = 1500 \cdot 10^3$, $\rho_a = 4.5 \cdot 10^{-6}$ and $c_a = 0.38 \cdot 10^9$, while the rest of the rod is made of a material with parameters $E_b = 800 \cdot 10^3$, $\rho_b = 4 \cdot 10^{-6}$ and $c_b = 0.466 \cdot 10^9$. The applied heat source appears in (5) as an additional force in the right hand side, $f_{nodal}(t) = (0, ..., 0, U(t))^T$, with $U(t)$ constrained to be continuous, piecewise linear and $U(t) \in [0, 10^6], \forall t$. We predefine the interpolation times for $U(t)$ to $\{0.5i | i = 0, 1, ...\}$. The rod starts at temperature $300\,\mathrm{K}$ at all points. We partition the spatial domain into a uniform mesh of different sizes and integrate the resulting FEM system using the trapezoidal rule with $\Delta t = 0.05$. For simplicity, we do not consider the integration error, and we approximate the ϵ, η, ν bounds by taking 100 samples of system trajectories with randomized control inputs, obtaining from them approximate maximum spatial and time derivatives, and also the maximum approximation error when considering the true system trajectory the one obtained from an FEM model with 200 elements and $\Delta t = 0.005$. This process is automatic and takes less than 20 seconds.

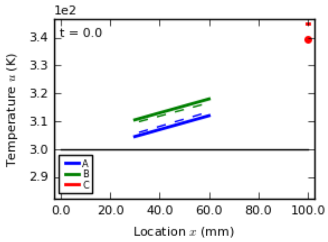
We used a 30 element mesh to synthesize a control input from the specification ϕ_{ex}, which produces a trajectory with robustness $r(\phi_{ex}) \geq 0.66$. The control input and snapshots of the temperature evolution are shown in Fig. 2, along with the temperature profiles considered in ϕ_{ex} (labeled A, B and C in the order they appear in the formula) and a visualization of the total correction applied during the discretization steps. It took 6 seconds to solve the resulting MILP with 13741 variables, 664 of them binary, and 14386 constraints. Furthermore, we show in Fig. 2d the relationship between the number of elements of the FEM partition and the conservativeness and computational complexity of the method. Note that for a 10 element mesh, the method fails to obtain a (provably) satisfying control input, and as quality of the mesh improves, so does the bound on the robustness.

B. Elastic Wave Propagation

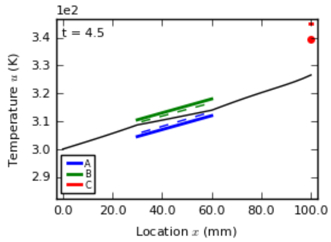
Consider a steel and brass rod of length $L = 100\,\mathrm{m}$, the section between $30\,\mathrm{m}$ and $60\,\mathrm{m}$ being brass, with densities $\rho_{st} = 8 \cdot 10^3\,\mathrm{kg/m^3}$, $\rho_{br} = 8.5 \cdot 10^3\,\mathrm{kg/m^3}$ and Young's modulus $E_{st} = 200\,\mathrm{GPa}$, $E_{br} = 100\,\mathrm{GPa}$, with one end fixed and a time-variant force $U(t)$ applied to the other end. Assume the rod is initially at rest. The displacement $u(x,t)$ of the rod obeys the following IBVP:

$$\Sigma^M(0, U) \quad \begin{cases} \rho \dfrac{\partial^2 u}{\partial t^2} - E \dfrac{\partial^2 u}{\partial x^2} = 0, \text{ on } \Omega \times (0, T)\,, \\ u(0, t) = 0, \ \forall t \in (0, T)\,, \\ E \dfrac{\partial u}{\partial x}(L, t) = U(t), \ \forall t \in (0, T)\,, \\ u(x, 0) = 0, \ \forall x \in \Omega\,, \\ \dfrac{\partial u}{\partial t}(x, 0) = 0, \ \forall x \in \Omega\,. \end{cases} \tag{20}$$

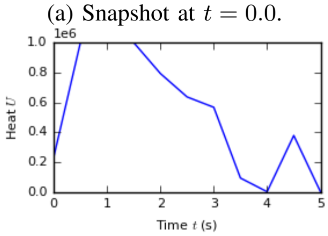
Note that in this mixed material case, $\rho = \rho(x)$ and $E = E(x)$ are the density and Young's modulus of the rod at each point. We build an FEM approximation using a uniform partition with 20 elements and integrate the resulting second order system using the trapezoidal rule [1] to obtain a time discretization with time interval $\Delta t = 5 \cdot 10^{-3}\,\mathrm{s}$. We obtain approximate bounds for ϵ, η, ν as in Sec. VII-A.



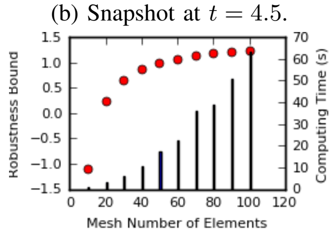
(a) Snapshot at $t = 0.0$.



(b) Snapshot at $t = 4.5$.

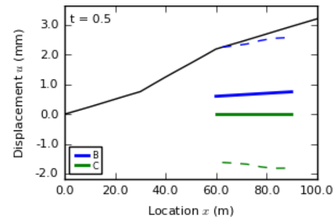


(c) Inputs for ϕ_{ex}.

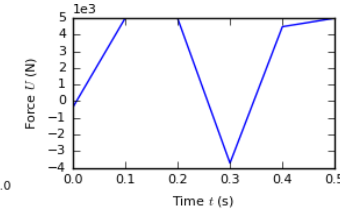


(d) Influence of mesh quality.

Fig. 2: In Figs. 2a and 2b we show snapshots of the satisfying trajectory for ϕ_{ex} in black, with predicate profiles and their corrections in solid and dashed lines respectively, except for the C profile, which is shown as a single dash and dot. In Fig. 2c we show the synthesized input. In Fig. 2d, we show the computing time (in black bars) and lower bound of the robustness (in red dots) as we increase the number of elements in the mesh.

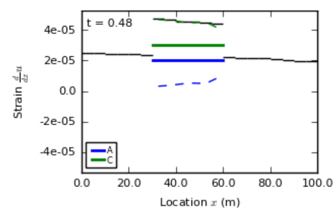


(a) Snapshot at $t = 0.5$.

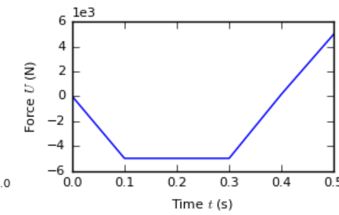


(b) Input for ϕ_1.

Fig. 3: Snapshot of the satisfying trajectory for ϕ_1 and synthesized control inputs. Trajectory is shown in black, predicate profiles in solid colored lines and corresponding corrected profiles in dashed lines.



(a) Snapshot at $t = 0.48$.



(b) Input for ϕ_2.

Fig. 4: Snapshot of the satisfying trajectory for ϕ_2 and synthesized control inputs. Trajectory is shown in black, predicate profiles in solid colored lines and corresponding corrected profiles in dashed lines.

We formulate a control synthesis problem where $U(t)$ is a Neumann control input constrained to be continuous, piece-wise affine and $U(t) \in [-5000\,\mathrm{N}, 5000\,\mathrm{N}], \forall t$. Throughout this section, we predefine the interpolation times for $U(t)$ to $\{0.1i | i = 0, 1, ...\}$. First, we formulate a specification where we require that the rod must be stretched over a given profile for a period of time, then compressed for at least one instant in a given interval, then a choice is given between holding the compressed pattern or returning to the stretched one, and finally the stretched pattern must be achieved within a time interval. Requirements where a material must be stretched or compressed following a profile are important in manufacturing processes such as forming. The resulting S-STL formula is the following:

$$\begin{aligned}\phi_1 = &\mathbf{G}_{[0.1,0.3]}(\forall x \in [60, 90] : u(x) - \mu_B > 0) \wedge \\ &\mathbf{F}_{[0.3,0.4]}(\forall x \in [60, 90] : u(x) - \mu_C < 0) \wedge \\ &\big(\mathbf{G}_{[0.45,0.5]}(\forall x \in [60, 90] : u(x) - \mu_C < 0) \vee \\ &\mathbf{G}_{[0.45,0.5]}(\forall x \in [60, 90] : u(x) - \mu_B > 0)\big) \wedge \\ &\mathbf{F}_{[0.5,0.55]}(\forall x \in [60, 90] : u(x) - \mu_B > 0)\,,\end{aligned} \tag{21}$$

where $\mu_B = 0.005x \cdot 10^{-3} + 0.3$ and $\mu_C = 0$. The specific form of the target profiles was selected considering the feasible shape changes the rod can achieve. The resulting $U(t)$ is shown in Fig. 3b, which corresponds to a trajectory with a robustness degree of at least 0.0003. We show a snapshot of the trajectory in Fig. 3a. It took 136 seconds to solve the MILP, which has 14604 variables, 730 of them binary, and 16035 constraints.

Finally, we consider a specification involving the strain, $\frac{d}{dx}u$, of the system, where the objective is to keep the strain in the brass section under a safe profile for some time, and then increase it so that the material yields or breaks within a time window. For this example we predefine $U(0) = 0$. The corresponding S-STL formula is the following:

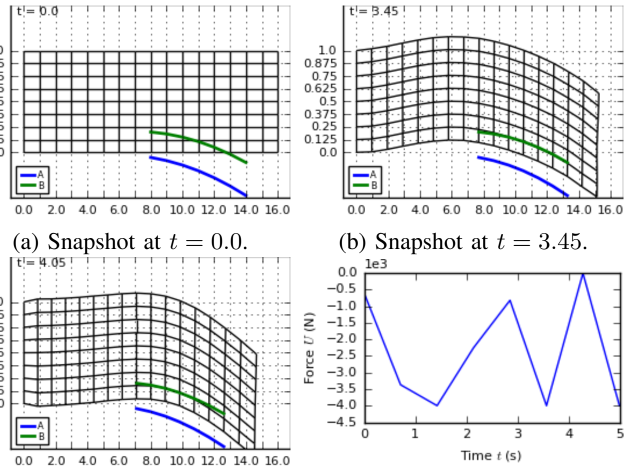
$$\begin{aligned}\phi_2 = &\mathbf{G}_{[0.1,0.4]}\big(\forall x \in [30, 60] : \frac{d}{dx}u(x) - \mu_A < 0\big) \wedge \\ &\mathbf{F}_{[0.4,0.5]}\big(\forall x \in [30, 60] : \frac{d}{dx}u(x) - \mu_C > 0\big)\,,\end{aligned} \tag{22}$$

where $\mu_A = 2 \cdot 10^{-5}, \mu_C = 3 \cdot 10^{-5}$. The control input obtained is shown in Fig. 4b. It corresponds to a trajectory with $1.26 \cdot 10^{-6}$ robustness and it was synthesized in 6 seconds. We show a snapshot of the trajectory in Fig. 4a.

C. 2D Beam

We consider now a 2D beam of length $L = 16\,\mathrm{m}$, width $c = 1\,\mathrm{m}$, Young's modulus $E = 1 \cdot 10^7$ and Poisson's ratio $v = 0.3$, initially at rest, assuming no body force and following a 2D linear isotropic elasticity theory with boundary conditions given for all t as follows:

$$\begin{aligned}&u_1(0, y) = u_2(0, y) = 0, && y \in [0, c]\,, \\ &h(x, y) = (0, 0), && x \in (0, L), y \in [0, c]\,, \\ &h_2(L, y) = 0, && y \in [0, c]\,, \\ &h_1(L, y) = \frac{y - .25}{.25}U, && y \in [.25, .45]\,, \\ &h_1(L, y) = \big(1 - \frac{y - .45}{.25}\big)U, && y \in [.45, .75]\,,\end{aligned} \tag{23}$$



(a) Snapshot at $t = 0.0$. (b) Snapshot at $t = 3.45$. (c) Snapshot at $t = 4.05$. (d) Input for ϕ.

Fig. 5: Snapshots of the satisfying trajectory for ϕ and synthesized input. The trajectory is shown as the deformation of the domain, represented in black by the mesh used in the FEM. The predicate profiles are shown in colored lines.

where $U = U(t)$ is the compressive force applied at the free end of the beam and h is the traction. The boundary conditions specify that the left end of the beam is fixed and a force is applied to the right end, distributed so that the maximum force is applied at $y = 0.45\,\mathrm{m}$, decaying linearly until $y = 0.25\,\mathrm{m}$ and $y = 0.75\,\mathrm{m}$.

We want to synthesize an input force such that the beam buckles. For this specific setup, we only need to specify the vertical displacement profile in part of the bottom boundary. We formulate the following S-STL specification:

$$\begin{aligned} \phi &= \mathbf{G}_{[3.45,4.05]}(A \wedge B)\,, \\ A &= \forall x \in \{x \in \Omega | 8 \le x_1 \le 14, x_2 = 0\}(u_2(x) > \mu_A(x))\,, \\ B &= \forall x \in \{x \in \Omega | 8 \le x_1 \le 14, x_2 = 0\}(u_2(x) < \mu_B(x))\,, \end{aligned} \tag{24}$$

where μ_A and μ_B are the quadratic functions depicted in Fig. 5a (labeled A and B). The specification states that the vertical displacement at part of the bottom boundary must be within the two given profiles at all times between $3.45\,\mathrm{s}$ and $4.05\,\mathrm{s}$. Note that a specification defining the shape of a material such as this one can be automatically constructed from a target shape plus a maximum allowed deviation.

We used a regular 8x4 9-node quadratic element mesh and integrated the second order system using the trapezoidal rule with $\Delta t = 75\,\mathrm{ms}$. The synthesized $U(t)$ is shown in Fig. 5d and corresponds to a trajectory with a robustness degree of at least 13. We show snapshots of the evolution of the beam shape in Fig. 5. It took 5546 seconds to solve the problem.

VIII. Conclusion

In this paper, we formulated and solved a boundary control synthesis problem for systems governed by a PDE with specifications given in an extension to STL. Our solution relies on the approximation of the PDE using the FEM, which reduces the problem to the control synthesis of a discrete-time linear system under regular STL constraints. The reformulation requires correcting the predicates in the formula using the FEM approximation errors, as well as the derivatives of the approximated solution and the predicate functions, to account for approximation and discretization errors. Finally, the resulting control problem is encoded as a MILP, which we show can be solved in minutes in several 1D and 2D examples. The method can be trivially adapted to solve verification problems as well.

Two key issues should be addressed in future work. First, the error and derivative bounds should be obtained in a systematic and formally correct way. Second, we have found the approach to be too conservative when using low quality meshes, while higher quality meshes prohibitively increase the size of the MILP, specially in 2D examples. Thus, the scalability of our method should be studied further.

References

[1] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Mineola, NY: Dover Publications, 2000.

[2] M. Krstic and A. Smyshlyaev, *Boundary Control of PDEs: A Course on Backstepping Designs*. Siam, 2008.

[3] T. Meurer and A. Kugi, "Tracking control for boundary controlled parabolic PDEs with varying parameters: Combining backstepping and differential flatness," *Automatica*, vol. 45, no. 5, pp. 1182–1194, May 2009.

[4] R. Gerth, D. Peled, M. Y. Vardi, and P. Wolper, "Simple On-the-fly Automatic Verification of Linear Temporal Logic," in *Protocol Specification, Testing and Verification XV*. Springer US, 1996, pp. 3–18.

[5] A. Donzé and O. Maler, "Robust Satisfaction of Temporal Logic over Real-Valued Signals," in *Formal Modeling and Analysis of Timed Systems*. Springer Berlin Heidelberg, Sept. 2010, no. 6246, pp. 92–106.

[6] M. Kloetzer and C. Belta, "A Fully Automated Framework for Control of Linear Systems from Temporal Logic Specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, Feb. 2008.

[7] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*, Dec. 2014, pp. 81–87.

[8] R. Grosu, S. A. Smolka, F. Corradini, A. Wasilewska, E. Entcheva, and E. Bartocci, "Learning and Detecting Emergent Behavior in Networks of Cardiac Myocytes," *Commun. ACM*, vol. 52, no. 3, pp. 97–105, Mar. 2009.

[9] E. Bartocci, E. A. Gol, I. Haghighi, and C. Belta, "A Formal Methods Approach to Pattern Recognition and Synthesis in Reaction Diffusion Networks," *IEEE Transactions on Control of Network Systems*, vol. PP, no. 99, pp. 1–1, 2016.

[10] I. Haghighi, A. Jones, Z. Kong, E. Bartocci, R. Gros, and C. Belta, "SpaTeL: A Novel Spatial-temporal Logic and Its Applications to Networked Systems," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*. New York, NY, USA: ACM, 2015, pp. 189–198.

[11] Ezio Bartocci, Luca Bortolussi, Michele Loreti, and Laura Nenzi, "Monitoring Mobile and Spatially Distributed Cyber-Physical Systems," in *Formal Methods and Models for System Design (MEMOCODE), 2017 ACM/IEEE International Conference On*, 2017.

[12] H. Abbas, H. Mittelmann, and G. Fainekos, "Formal property verification in a conformance testing framework," in *2014 Twelfth ACM/IEEE Conference on Formal Methods and Models for Codesign (MEMOCODE)*, Oct 2014, pp. 155–164.

[13] I. Gurobi Optimization, "Gurobi Optimizer Reference Manual," *URL: http://www. gurobi. com*, 2016.

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