Duality-Based Nested Controller Synthesis from STL Specifications for Stochastic Linear Systems

Susmit Jha[1], Sunny Raj[2], Sumit Kumar Jha[2], and Natarajan Shankar[1]

[1] Computer Science Laboratory, SRI International, USA
{susmit.jha,shankar}@sri.com
[2] Computer Science Department, University of Central Florida, Orlando, USA
{sraj, jha}@eecs.ucf.edu

Abstract. We propose an automatic synthesis technique to generate provably correct controllers of stochastic linear dynamical systems for Signal Temporal Logic (STL) specifications. While formal synthesis problems can be directly formulated as exists-forall constraints, the quantifier alternation restricts the scalability of such an approach. We use the duality between a system and its proof of correctness to partially alleviate this challenge. We decompose the controller synthesis into two subproblems, each addressing orthogonal concerns - stabilization with respect to the noise, and meeting the STL specification. The overall controller is a nested controller comprising of the feedback controller for noise cancellation and an open loop controller for STL satisfaction. The correct-by-construction compositional synthesis of this nested controller relies on using the guarantees of the feedback controller instead of the controller itself. We use a linear feedback controller as the stabilizing controller for linear systems with bounded additive noise and over-approximate its ellipsoid stability guarantee with a polytope. We then use this over-approximation to formulate a mixed-integer linear programming (MILP) problem to synthesize an open-loop controller that satisfies STL specifications.

1 Introduction

Cyber-physical systems can be conceptually decomposed into a physical plant and a controller. The complex interaction between the plant and the controller often necessitates an hierarchical control. While high-level decisions are typically made by a supervisory controller, traditional control laws such as PID control are typically used at low levels. These controllers at different levels are often designed in isolation, and then plugged into a hierarchical framework to build an ad hoc implementation that can be evaluated through simulations and in-the-field experiments. For safety-critical systems, design of such hierarchical controller often relies on the worst-case characterization of independently designed controllers in each layer, which leads to overly conservative design with low performance. This problem is becoming even more acute with the

growing complexity of cyber-physical systems. Hence, there is a pressing need for automatic synthesis techniques that can co-design controllers at different layers in a synergistic way for an optimal yet safe hierarchical control of cyber-physical systems.

Safety-critical applications of cyber-physical systems necessitate providing assurance and safety certification of the controllers. Approaches based on barrier certificates [5, 33] and Lyapunov functions [15] are applicable to proving stability, asymptotic convergence, and safety of continuous control laws but their extensions to hierarchical controls and stochastic dynamics is difficult. Automatic synthesis of controllers from high-level specifications [10, 30, 27, 13] either in the open-loop setting or model-predictive and reactive setting have also been studied [35, 12]. These methods ensure that the synthesized controllers are correct by construction. While these techniques based on mixed integer linear programming (MILP) have been shown to scale well, they are limited to linear deterministic dynamics. More recently, extensions to uncertainty in dynamics and observations have also been proposed [19] using a chance-constraint programming formulation. But these methods are restricted to Gaussian noise and use a less scalable semi-definite programming formulation. Further, these offline synthesis methods try to be robust to worst-case noise which makes them very conservative. Thus, safe controller design for high-level temporal properties in presence of noisy dynamics remains a challenge.

In this paper, we study the problem of synthesizing safe control for linear, discrete-time plant with bounded disturbance against high-level temporal logic specifications expressed in signal temporal logic (STL). A natural paradigm for designing controllers for reach-avoid properties in presence of noise comprises of designing an open-loop controller ignoring noise, followed by a tracking controller to drive the trajectory towards the reference trajectory in presence of noise. We formulate a bottom-up approach to controller synthesis which does not ignore the interdependencies between the two controllers. We first synthesize a stabilizing controller to reject noisy disturbances and then use its stability certificate to formulate a less conservative robust open-loop controller synthesis problem for STL specifications using MILP. The novel contributions in this paper are as follows:

- We extend the MILP based controller synthesis approach for signal temporal logic (STL) specifications to dynamics with bounded noise.
- We present a new approach for nesting controllers that allows composing correctness guarantee of the low-lever noise-canceling controller during synthesis, enabling a compositional proof of correctness of the overall nested controller.
- We experimentally validated the effectiveness of the controller synthesis approach on a set of case-studies.

We discuss related work and background in Section 2 and Section 3. We formulate the controller synthesis problem in Section 4. We present the proposed synthesis approach in Section 5 and experimental evaluation in Section 6.

2 Related Work

We briefly discuss related work on formal synthesis of controllers from high-level specifications, and compare and contrast with our proposed approach.

Synthesis of safe control using reachability analysis has been extensively studied in literature where the specification is restricted to reach-avoid properties requiring that a particular target state be reached while avoiding unsafe states [29, 28, 39]. More recently, safe control optimization techniques have been developed which allow exploration of control parameter space and online learning of optimal controller while remaining safe [2, 4]. These techniques rely on learning probabilistic model of uncertainty either offline or online at runtime and computation of reachable sets. Our approach is orthogonal to techniques for estimating or modeling uncertainty, and we focus on the synthesis of safe control for an additive noise model. The control of stochastic systems has also been extensively investigated beginning with the work of Pontryagin [31] and Bellman [3], and extending to more recent literature [23, 34, 33, 8, 18]. The goal of these techniques is to determine a control policy that maximizes the probability of remaining within a safe set during a finite time horizon [1]. In contrast, we consider a bounded noise model and require deterministic safety with respect to high-level temporal specifications.

Temporal logic such as linear temporal logic have been used for high-level specification of goals. Controller design with respect to high-level specifications for linear dynamics model has been studied in [12, 40, 21, 41], and extended to polynomial systems [9] and other nonlinear systems using piecewise linear approximation [42, 16, 6]. The synthesis techniques can be broadly classified into automata theoretic and constraint-based approaches. Automata theoretic techniques for controller synthesis from temporal specifications such as LTL are based on discrete, finite-state and symbolic abstraction of the system. Then, the solution of a two player game on the abstracted game graph is obtained by composing the discrete abstraction with a discrete controller. While these techniques can be used with nonlinear dynamics in principle, the discrete abstraction severely limits their scalability for high dimensional models. Our approach is closer to constraint-solving based methods. While extension of satisfiability solving to deal with continuous dynamics has been studied in literature [11, 22], we adopt the use of mixed-integer linear programming for solving the open loop synthesis problem which is sufficient for modeling linear dynamics. We use signal temporal logic (STL) for specifying the requirements of the controller. STL has been proposed as an extension of linear temporal logic for specifying behavior of continuous and hybrid systems [10]. It combines dense time modalities with numerical predicates over continuous state variables. Automatic synthesis of controllers from STL properties using mixed integer linear programming has proved to be an efficient and scalable approach [35], and more recently, it has been recently extended to chance-constraints [19, 17, 36]. While these previous extensions require computationally expensive second order cone programming, we present an

MILP formulation of STL controller synthesis for linear dynamical system with additive but bounded noise. Further, we demonstrate the nested controller synthesis approach that uses an online noise canceling controller in conjunction with an offline open loop controller. This nested approach leads to less conservative formulation than a direct offline robust formulation that considers worst-case noise.

Invariant based methods that rely on generating barrier certificates or Lyapunov invariants [5, 15] have been also well-studied in literature. Invariant based control can be combined with other high-performance controllers to provide guarantees in a Simplex architecture [38]. More recently, it has been extended to synthesize switching control [24, 32] for a family of dynamical systems by formulating a finite game graph that consists of the switching surfaces as the existential nodes and the choices of the dynamics as the universal nodes. Instead of switching between different modes or two different controllers, we use the invariant guarantee provided by the noise-canceling lower-level controller to formulate a nested safe but less conservative open-loop synthesis problem. Our work is closest to nested controller synthesis methods [37, 14]. Our approach considers general STL properties and is not limited to reach avoid properties. Further, we use the guarantee provided by the low-level controller as a dual to synthesize the nested open loop STL controller without explicitly composing the two controllers.

3 Preliminaries

We consider a discrete-time linear system Σ of the form

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \omega_t \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is the dynamics matrix, $\mathbf{x}_t \in X \subseteq \mathbb{R}^n$ is the system state, $B \in \mathbb{R}^{n\times m}$ is the control input matrix, $\mathbf{u}_t \in U \subseteq \mathbb{R}^m$ is the controller input, and $\omega_t \in D \subseteq \mathbb{R}^n$ is the bounded additive noise disturbance. $\mathbb{R}$ denotes the set of reals, X, U are closed polytopes that represents the set of all possible states and feasible control inputs. D represents bounded noise, that is,

$$\forall \omega \in D \quad \omega^T M^T M \omega \leq \Omega^2 \tag{2}$$

If M is identity, the above is the familiar 2-norm bound. We choose a generic M-norm since noise in different dimensions may have asymmetric significance. The set of initial states of the system is denoted by $\mathbf{X}_0$. We denote a sequence of control inputs $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}$ of length N by $\overline{\mathbf{u}}_N$, and a sequence of noise disturbances $\omega_0, \omega_1, \ldots, \omega_{N-1}$ of length N by $\overline{\omega}_N$. We say $\overline{\omega}_N \in D^N$ if $\omega_i \in D$ for $i \in [0, N-1]$, and $\overline{\mathbf{u}}_N \in U^N$ if $\mathbf{u}_i \in U$ for $i \in [0, N-1]$. Starting from an initial state $\mathbf{x}_0 \in \mathbf{X}_0$ and applying the control inputs $\overline{\mathbf{u}}_N$ and noise disturbances $\overline{\omega}_N$, the horizon-N trajectory of the system $\mathbf{x}_0\mathbf{u}_0, \mathbf{x}_1\mathbf{u}_1, \ldots, \mathbf{x}_N\mathbf{u}_N$ is denoted by $\tau(\mathbf{x}_0, \overline{\mathbf{u}}_N, \overline{\omega}_N)$.

For specifying the requirements on the controlled dynamical system, we use signal temporal logic (STL). Let $\mathbb{B}$ denotes the set of Boolean values $\top, \bot$

denoting true and false respectively. An STL formula can be constructed recursively using the following grammar:

$$\phi \;:=\; \pi^{\mu} \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid G_{[a,b]}\phi \mid F_{[a,b]}\phi \mid \phi_1 U_{[a,b]}\phi_2$$

where π^{μ} is an atomic predicate $X \times U \to \mathbb{B}$ whose truth value is determined by the sign of a signal $\mu : X \times U \to \mathbb{R}$. $\tau(\mathbf{x}_0, \overline{\mathbf{u}}_N, \overline{\omega}_N) \models \phi$ denotes that the trajectory $\tau(\mathbf{x}_0, \overline{\mathbf{u}}_N, \overline{\omega}_N)$ satisfies an STL formula ϕ. When the arguments are obvious from context, we also denote it by $\tau \models \phi$ and $\tau[i]$ denotes the i-th element $\mathbf{x}_i\ \mathbf{u}_i$ in the sequence. Informally, $\tau \models G_{[a,b]}\phi$ if ϕ holds at every time step between a and b. $\tau \models F_{[a,b]}\phi$ if ϕ holds at some time step between a and b. $\tau \models \phi_1 U_{[a,b]}\phi_2$ holds if ϕ_1 holds at every time step before ϕ_2 holds and ϕ_2 holds at some time step between a and b. Formally, the validity of a formula ϕ with respect to the run τ is defined inductively as follows:

$$\begin{aligned}
\tau \models \phi \quad &\iff \quad \tau[0] \models \phi \\
\tau[t_k] \models \pi^{\mu} \quad &\iff \quad \mu(\mathbf{x}_k, \mathbf{u}_k) > 0 \\
\tau[t_k] \models \neg\phi \quad &\iff \quad \tau[t_k] \not\models \phi \\
\tau[t_k] \models \phi_1 \wedge \phi_2 \quad &\iff \quad \tau[t_k] \models \phi_1 \wedge \tau[t_k] \models \phi_2 \\
\tau[t_k] \models G_{[a,b]}\phi \quad &\iff \quad \forall t \in [t_k + a, t_k + b] \;\; \tau[t] \models \phi \\
\tau[t_k] \models F_{[a,b]}\phi \quad &\iff \quad \exists t \in [t_k + a, t_k + b] \;\; \tau[t] \models \phi \\
\tau[t_k] \models \phi_1 U_{[a,b]}\phi_2 \quad &\iff \quad \exists t_1 \in [t_k + a, t_k + b] \;\; (\tau[t_1] \models \phi_2 \\
&\qquad\qquad \wedge \forall t_2 \in [t_k + a, t_1] \tau[t_2] \models \phi_1)
\end{aligned}$$

Bounded-time STL contains no unbounded temporal operators and the bound of ϕ is the maximum over the sum of all nested upper bounds on the trajectory operators. The bound of ϕ is a conservative bound on the trajectory length required to decide its satisfiability.

Typical properties such as reach-avoid can be easily encoded as an STL formula. For example, if we require a vehicle to reach a particular destination region while avoiding obstacles. The STL specification for a vehicle starting in state $\mathbf{x}_0$ and reaching $\mathcal{R}^{dest}$ within T time-steps while avoiding obstacles $\mathcal{R}^{obs_1}, \ldots, \mathcal{R}^{obs_k}$ is $F_{[0,T]}\mathcal{R}^{dest}(\mathbf{x}) \wedge G_{[0,T]}(\neg\mathcal{R}^{obs_1}(\mathbf{x}) \wedge \ldots \neg \wedge \mathcal{R}^{obs_k}(\mathbf{x}))$. Any region of interest $\mathcal{R}$ (destination or an obstacle) can be approximated using a union of polytopes, represented by a disjunction of conjunction of linear constraints. For soundness, the choice of under or over approximation depends on the region being approximated. For example, one would under-approximate the destination region while over-approximating the obstacles to ensure that a feasible trajectory with respect to this approximation can safely reach within the destination region while avoiding the obstacles. So, we restrict the atomic predicates in the signal temporal logic formulas to be linear inequalities, that is, the signals μ are restricted to be linear combinations of state variables and control inputs.

We use mixed integer linear programming (MILP) encoding of an STL formula [35]. A variable z_t^{ϕ} is introduced for an STL formula ϕ with horizon N,

and MILP constraints are formulated on this variable such that $z_t^\phi = 1$ if and only if ϕ holds at time t. Let M be sufficiently large and ϵ be sufficiently small, the MILP constraints corresponding to z_t^ϕ can be generated as follows:

$$
\begin{aligned}
&\mathit{not}(z, z') && \equiv z = 1 - z' \\
&\mathit{and}(z, [z_1, \ldots, z_n]) && \equiv \bigwedge_{i=1}^{n} (z \leq z_i) \ \wedge \ z \geq \sum_{i=1}^{n} z_i - n + 1 \\
&\mathit{or}(z, [z_1, \ldots, z_n]) && \equiv \bigwedge_{i=1}^{n} (z \geq z_i) \ \wedge \ z \leq \sum_{i=1}^{n} z_i \\
&\texttt{encode}(\mu(\mathbf{x}, \mathbf{u}) > 0, t) && \equiv \mu(\mathbf{x}_t, \mathbf{u}_t) \leq M z_t^\mu - \epsilon \ \wedge \ -\mu(\mathbf{x}_t, \mathbf{u}_t) \leq M(1 - z_t^\mu) - \epsilon \\
&\texttt{encode}(\neg\phi, t) && \equiv \mathit{not}(z_t^{\neg\phi}, z_t^\phi) \ \wedge \ \texttt{encode}(\phi, t) \\
&\texttt{encode}(\phi_1 \wedge \phi_2, t) && \equiv \mathit{and}(z_t^{\phi_1 \wedge \phi_2}, [z_t^{\phi_1}, z_t^{\phi_2}]) \wedge \texttt{encode}(\phi_1, t) \wedge \texttt{encode}(\phi_2, t) \\
&\texttt{encode}(\phi_1 \vee \phi_2, t) && \equiv \mathit{or}(z_t^{\phi_1 \vee \phi_2}, [z_t^{\phi_1}, z_t^{\phi_2}]) \wedge \texttt{encode}(\phi_1, t) \wedge \texttt{encode}(\phi_2, t) \\
&\texttt{encode}(G_{[a,b]}\phi, t) && \equiv \mathit{and}(z_t^{G_{[a,b]}\phi}, [z_{t+a}^\phi \ \ldots \ z_{t+b}^\phi]) \ \wedge \ \bigwedge_{t'=a}^{b} \texttt{encode}(\phi, t + t') \\
&\texttt{encode}(F_{[a,b]}\phi, t) && \equiv \mathit{or}(z_t^{F_{[a,b]}\phi}, [z_{t+a}^\phi \ \ldots \ z_{t+b}^\phi]) \ \wedge \ \bigwedge_{t'=a}^{b} \texttt{encode}(\phi, t + t') \\
&\texttt{encode}(\phi_1 U_{[a,b]} \phi_2, t) && \equiv \texttt{encode}(\ G_{[0,a]}\phi_1 \ \wedge \ F_{[a,b]}\phi_2 \ \wedge \ F_{[a,a]}(\phi_1 U \phi_2), t \ ) \\
&\texttt{encode}(\phi_1 U \phi_2, t) && \equiv \mathit{or}(z_t^{\phi_2}, \mathit{and}(z_t^{\phi_1}, z_{t+1}^{\phi_1 U \phi_2})) \wedge \texttt{encode}(\phi_1, t) \wedge \texttt{encode}(\phi_2, t) \\
& && \wedge \ \mathit{if}(t < N) \ \mathit{then} \ \texttt{encode}(\phi_1 U \phi_2, t+1) \ \mathit{else} \ z_N^{\phi_1 U \phi_2} = z_N^{\phi_2}
\end{aligned}
$$

We also briefly review the linear state feedback control used for stabilizing a system. Given a system $\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t$, let the feedback controller be given by $\mathbf{u}_t = -K\mathbf{x}_t$. The dynamics of the controlled system is given by

$$\mathbf{x}_{t+1} = (A - BK)\mathbf{x}_t$$

This system is stable if and only if the spectral radius of $(A - BK)$ is less than 1, that is, $(A - BK)$ is contracting. The linear stabilizing feedback controller synthesis problem is to solve the following problem:

$$\exists K \ \exists P \succeq 0 \ \ (A - BK)^T P (A - BK) \prec P$$

Unfortunately, this is not a semi-definite program (SDP) since the matrix inequality is not linear in the decision variables P and K.

4 Problem Definition

In this section, we formulate the problem of synthesizing safe control for a stochastic linear dynamical system so that the system satisfies the given STL specification.

Controller synthesis problem: Given a system Σ of the form: $\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \omega_t$ with initial state $\mathbf{x}_0$, a high-level signal temporal logic (STL) specification ϕ with horizon N, the controller synthesis problem is as follows:

$$\exists \mathbf{u}_0 \forall \omega_0 \forall \mathbf{x}_1 \exists \mathbf{u}_1 \forall \omega_1 \forall \mathbf{x}_2 \ldots \exists \mathbf{u}_{N-1} \forall \omega_{N-1} \forall \mathbf{x}_N \quad \tau(\mathbf{x}_0, \overline{\mathbf{u}}_N, \overline{\omega}_N) \models \phi$$

where $\mathbf{x}_t \in X, \mathbf{u}_t \in U, \omega_t \in D = \{\omega \mid \omega^T M^T M \omega \leq \Omega^2\}$. The control inputs are generated by a controller $\texttt{cntlr}(\mathbf{x}_0, \phi, \mathbf{x}_t, X, U, D)$ which maps the initial state, STL specification, and current state to the control input assuming that the disturbance $\omega_t \in D$ and ensuring that the states $x_t \in X$ and control inputs $u_t \in U$. We use this controller function for Skolemization and elimination of the existential quantifiers on the control inputs. The controller synthesis problem can then be written as an exists-forall problem.:

$$\exists \texttt{cntlr}\ \forall \omega_0 \forall \mathbf{x}_1 \forall \omega_1 \forall \mathbf{x}_2 \ldots \forall \omega_{N-1} \forall \mathbf{x}_N \ \ \tau(x_0, \overline{\mathbf{u}}_N, \overline{\omega}_N) \models \phi \text{ where } \mathbf{x}_t \in X, \mathbf{u}_t \in U,$$
$$\mathbf{u}_t = \texttt{cntlr}(\mathbf{x}_0, \phi, \mathbf{x}_t, X, U, D),\ \mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \omega_t,\ \omega_t^T M^T M \omega_t \leq \Omega^2 \quad (3)$$

Instead of requiring the controller to have to store the entire history of states and the noise, we have restricted the controller `cntlr` to generate a control input using only the current state, the initial state and the STL specification, in addition to the sets X, U and D. The goal is to synthesize such a controller which can satisfy the STL specification even in presence of noise.

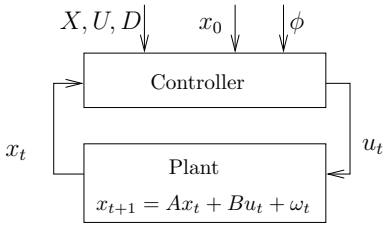


Fig. 1. Controller has access to the STL specification ϕ to be satisfied, an initial state $\mathbf{x}_0$ of the system, the bounding sets X, U and D. It continuously receives the current state $\mathbf{x}_t$ of the system. It produces control inputs $\mathbf{u}_t$ which can be computed offline (for example, in open-loop control), online (for example, in feedback control) or a combination of offline and online (for example, in nested control presented in Section 5). The presence of noise ω_t makes completely offline safe control synthesis very conservative as the synthesis algorithm has to consider worst-case accumulative effect of noise.

5 Controller Synthesis

We first describe over-approximation of elliptical bounds on the noise that will be used in controller synthesis. Given $\mathcal{D}_M^\Omega = \omega^T M^T M \omega \leq \Omega^2$ which restricts

the noise in an ellipse, we can over approximate this ellipse using hyperboxes that are axis-parallel or parallel to orthogonal[3] eigenvectors (sketched for two dimensions in Figure 2). We construct an over-approximation of possible disturbances $\mathcal{OA}(M,\Omega) \supseteq \mathcal{D}_M^\Omega$ by taking the intersection of the two over-approximations.

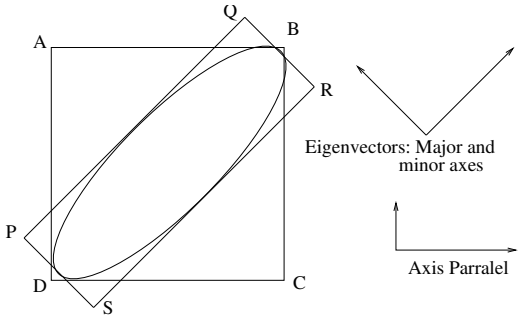


Fig. 2. Given an elliptical $\mathcal{D}_M^\Omega$ in two dimensions defined by $\omega^T M^T M \omega \leq \Omega^2$, $ABCD$ is the axis parallel hyperbox $\mathcal{P}_M^\Omega$ over-approximating the ellipse. The eigenvectors of the ellipse correspond to the major and minor axes of the ellipse. $PQRS$ is the hyperbox $\mathcal{E}_M^\Omega$ over-approximating the ellipse by bounding the eigenvectors. $\mathcal{OA}(M,\Omega) = \mathcal{P}_M^\Omega \cap \mathcal{E}_M^\Omega$ is the polytope corresponding to the intersection of these two hyperboxes represented by the conjunction of linear constraints of both hyberboxes.

Before presenting the nested controller synthesis approach, we discuss two straightforward solutions to the the synthesis problem by direct application of standard control theoretic techniques.

Open loop robust controller. The synthesis problem can be solved using a robust controller that considers the worst-case noise and synthesizes control with respect to it for the given horizon N. This method aims at jointly addressing the satisfiability of the mission specification in STL and robustness with respect to bounded noise without decomposing the problem. The following mixed-integer linear constraints formulate the finite horizon open loop robust controller synthesis problem:

$$\forall t \in [0,N]\ \mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \omega_t$$
$$\texttt{encode}(\phi,0), \omega \in \mathcal{OA}(M,\Omega), \mathbf{x} \in X, \mathbf{u} \in U$$

where `encode` is MILP encoding of the specification ϕ, $\mathcal{OA}(M,\Omega)$ is the polytope overapproximation of the disturbance set D, and X, U are conjunction of linear constraints restricting the states and control inputs to allowed polytopes. Consequently, the state at time t in the above formulation is

[3] Eigenvectors are orthogonal since $M^T M$ is symmetric

given by $\mathbf{x}_t = A^t\mathbf{x}_0 + (A^{t-1}B\mathbf{u}_0 + A^{t-2}B\mathbf{u}_1 + \ldots + B\mathbf{u}_t) + (A^{t-1}\omega_0 + A^{t-2}\omega_1 + \ldots + \omega_t)$. This considers the worst-case noise irrespective of the actual noise experienced at runtime and consequently leads to very conservative controller design.

Tracking controller. The second alternative is to decompose the synthesis problem by first ignoring the noise and synthesizing an open-loop controller to satisfy the STL specification. At runtime, a tracking controller can be used to ensure that the system tracks the noise-free state trajectory corresponding to the open-loop controller synthesis problem. The open loop controller is synthesized by solving the following mixed integer linear program:

$$\forall t \in [0, N]\ \mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t$$
$$\texttt{encode}(\phi, 0), \mathbf{x} \in X, \mathbf{u} \in U$$

The satisfiable solution to these linear constraints yield $\mathbf{x}_t$ and $\mathbf{u}_t$ for $t \in [0, N]$. Once we have these reference signals, we use the standard pole placement method to design a feedback controller that tracks this reference. In practice, solving the feasibility problem corresponding to the linear program yields *borderline* solutions which just barely satisfy the constraints. This is the consequence of search methods used to solve these problems. This makes it even more difficult to design tracking controller which prevents the trajectory from failing the STL specification in presence of noise.

Nested controller. While a closed loop solution is needed to be not overly conservative, we also require it to have correctness guarantees similar to the open loop robust controller. We accomplish this by first designing a noise stabilizing controller and then using its robustness guarantees to synthesize the open loop control inputs that robustly satisfies the STL. In rest of this section, we describe this two step design of nested controller in detail:

Feedback Controller: Given the linear dynamical system $\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t^{fb} + \omega_t$ and a linear feedback stabilization controller $\mathbf{u}_t^{fb} = -K\mathbf{x}_t$, the deviation from the reference trajectory is given by

$$\mathbf{x}_{t+1} - \mathbf{x}_{t+1}^{ref} = (A - BK)(\mathbf{x}_t - \mathbf{x}_t^{ref}) + \omega_t$$

We need to find K such that $A - BK$ is stable. While this problem of stabilizing linear systems can be solved using a variety of control theoretic methods, this choice is orthogonal to the nested control approach proposed in this paper. We use pole placement approach [20] similar to the tracking controller and ensure that the poles lie in the left half plane. This guarantees that $A - BK$ is stable and the spectral radius $\rho(A - BK) < 1$.

Lemma 1. *Given a feedback control matrix K that stabilizes $A - BK$, the transform $A - BK$ is a contracting transform and $\forall \mathbf{d}\ \mathbf{d}^T(A - BK)^T M^T M(A - BK)\mathbf{d} \leq \mathbf{d}^T M^T M\mathbf{d}$.*

While this contraction could be provided as a stability guarantee to be used by the open loop controller synthesis in the second step, we can further refine this guarantee by considering a new shape of the ellipsoid invariant that ensures maximum contraction due to the feedback controller K. This refinement of guarantee is important for obtaining less conservative yet correct open loop controller. The following semidefinite programming problem yields the optimal shape M' and the corresponding contraction rate κ.

$$\min_{M',\kappa} \kappa \quad \text{subject to } (A-BK)^T M'^T M'(A-BK) \preceq \kappa^2 \; M'^T M'$$

Lemma 2. *Given a feedback control matrix K that stabilizes $A-BK$ and the solution of the above optimization problem M', κ, $\forall \mathbf{d}$ $\mathbf{d}^T(A-BK)^T M'^T M'(A-BK)\mathbf{d} \leq \kappa^2 \; \mathbf{d}^T M'^T M'\mathbf{d}$.*

Noise reshaping: We solve the following optimization problem to reshape the bounds on the noise to conform to optimum shape discovered above.

$$\min_{\Omega'} \Omega' \quad \text{subject to } \forall \mathbf{d} \; \mathbf{d}^T M^T M \mathbf{d} \leq \Omega^2 \Rightarrow \mathbf{d}^T M'^T M' \mathbf{d} \leq \Omega'^2$$

The implication between quadratic constraints allow the use of S-lemma [7] to formulate a semidefinite programming formulation. After we have obtained the bound Ω'^2, we can extend the guarantee provided by the feedback controller for t timesteps as given in Theorem 1. The proof of theorem follows from Lemma 2 and the repeated use of triangular inequality.

Theorem 1. *Given a feedback control matrix K that stabilizes $A-BK$, the optimal shape and bound of stabilization guarantee M', κ and corresponding noise bound Ω', the state $\mathbf{x}_t$ at time step t satisfies $(\mathbf{x}_t - \mathbf{x}_t^{ref})^T M'^T M'(\mathbf{x}_t - \mathbf{x}_t^{ref}) \leq S(\kappa,t)\Omega'$ where $\kappa < 1$ since K stabilizes $A-BK$ and $S(\kappa,t) = (1+\kappa^2+\ldots+\kappa^t)$.*

In solving for the feedback controller K, we did not have to fix the reference trajectory and as long as we use this feedback controller at runtime with $\mathbf{u}^{fb} = -K(\mathbf{x}_t - \mathbf{x}_t^{ref})$, Theorem 1 guarantees the upper bound on possible deviation from any chosen reference trajectory $\mathbf{x}^{ref}$. We will use this guarantee in synthesizing the open loop STL controller $\mathbf{u}^{open}$ and selecting the corresponding $\mathbf{x}^{ref}$. The stabilization certificate provided by the feedback control synthesis step to the open loop control synthesis step is a triplet (M', Ω', κ).

STL Controller: Figure 3 summarizes the overall synthesis approach and illustrates how the runtime control $\mathbf{u}_t$ is obtained by adding the open loop control input and the feedback control input. We recall the Equation 3 that summarizes the exists-forall formulation of the controller synthesis problem, and decompose the controller into two controllers. The first controller is an open loop controller that lays a reference trajectory while the second controller

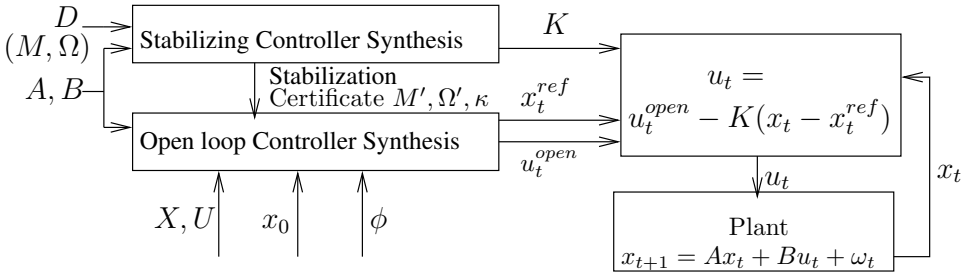


Fig. 3. Nested Controller for STL satisfaction in presence of noise. The stabilizing controller uses the noise bound D to find the required feedback controller K and obtain corresponding stability guarantee. The open loop controller synthesis only relies on the stability guarantee provided by the stabilizing feedback controller.

is the feedback controller described earlier to stabilize against noise.

$$\exists \texttt{cntlr}^{fb}\ \exists \texttt{cntlr}^{open}\ \forall\omega_0\forall\mathbf{x}_1\forall\omega_1\forall\mathbf{x}_2\ldots\forall\omega_{N-1}\forall\mathbf{x}_N\ \ \tau(x_0,\overline{\mathbf{u}}_N,\overline{\omega}_N)\models\phi$$
$$\mathbf{u}_t^{fb}=\texttt{cntlr}^{fb}(\mathbf{x}_t,\mathbf{x}_t^{ref},D),\mathbf{u}_t^{ref}=\texttt{cntlr}^{open}(\mathbf{x}_0,\phi,X,U,\texttt{cntlr}^{fb})$$
$$\text{where } \mathbf{x}_t\in X,\mathbf{u}_t=\mathbf{u}_t^{ref}+\mathbf{u}_t^{fb}\in U,\omega_t\in\mathcal{D}_M^{\Omega},\ \mathbf{x}_{t+1}=A\mathbf{x}_t+B\mathbf{u}_t+\omega_t$$

Instead of generating the open loop controller taking into account the feedback controller, we use duality to only require the stabilization guarantee (M',Ω',κ) to be available to the feedback controller. This guarantee can be used to eliminate the forall quantification over noise in the above formulation, and using Theorem 1, we obtain the following:

$$\exists \texttt{cntlr}^{open}\ \forall\mathbf{x}_1\forall\mathbf{x}_2\ldots\forall\mathbf{x}_N\ \ \tau(x_0,\overline{\mathbf{u}}_N,\overline{\omega}_N)\models\phi$$
$$\mathbf{u}_t^{ref}=\texttt{cntlr}^{open}(\mathbf{x}_0,\phi,X,U,M',\Omega',\kappa),(\mathbf{x}_t-\mathbf{x}_t^{ref})\in\mathcal{D}_M'^{S(\kappa,t)\Omega'}$$
$$\text{where } \mathbf{x}_t\in X,\mathbf{u}_t=\mathbf{u}_t^{ref}+\mathbf{u}_t^{fb}\in U,\ \mathbf{x}_{t+1}=A\mathbf{x}_t+B\mathbf{u}_t+\omega_t$$

Finally, we use the polytope approximation $\mathcal{OA}(M',S(\kappa,t)\Omega')$ of the elliptical constraint $\mathcal{D}_M'^{S(\kappa,t)\Omega'}$ as described earlier to obtain the following MILP program that solves the open loop controller synthesis problem using the stability guarantee of the feedback controller:

$$\forall t\in[0,N]\ \mathbf{x}_{t+1}^{ref}=A\mathbf{x}_t+B\mathbf{u}_t^{ref},\ (\mathbf{x}_t-\mathbf{x}_t^{ref})\in\mathcal{OA}(M',S(\kappa,t)\Omega')$$
$$\texttt{encode}(\phi,0),\mathbf{x}_t\in X,\mathbf{u}_t=\mathbf{u}_t^{ref}-K(\mathbf{x}_t-\mathbf{x}_t^{ref})\in U$$

The following theorem summarizes the soundness of the proposed approach to synthesize nested controller.

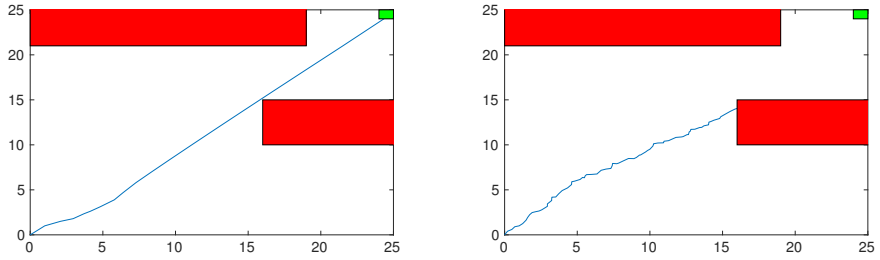
Theorem 2. *Given a dynamical system $\mathbf{x}_{t+1}=A\mathbf{x}_t+B\mathbf{u}_t+\omega_t$ with bounds on state, control and noise (X,U,D), if the MILP formulation of the synthesis problem is feasible and finds a controller $\mathbf{u}_t=\mathbf{u}_t^{ref}+\mathbf{u}_t^{fb}$, then the system starting at $\mathbf{x}_0$ satisfies the STL specification ϕ even in the presence of bounded noise.*

6 Case Studies

In this section, we present three case-studies to demonstrate the effectiveness and efficiency of the proposed approach to synthesize nested controller. All experiments were conducted on 8-core 2.8GHz Intel® Xeon® CPU with 16GB RAM using Matlab®. The first case-study involves controlling a vehicle moving in a map with obstacles. The second case-study is on smart grid control adapted from [26], and the third is indoor climate control case-study [25, 35].

6.1 Case Study 1: Simple vehicle model

In this case study, we consider a robot that is expected to navigate in a 2-dimensional grid. The dynamics in each dimension, x and y, is given by a simple double integrator model. It starts at the bottom left corner and is required to reach the top right corner within 10 time units. The robot must avoid the two obstacles shown as red rectangles in in Fig. 4. This requirement can be captured by the corresponding signal temporal logic property $F_{[0,10]}(24 \leq x_t \leq 25 \wedge 24 \leq y_t \leq 25) \wedge G_{[0,10]}(\neg(0 \leq x_t \leq 19 \wedge 21 \leq y_t \leq 25) \wedge \neg(16 \leq x_t \leq 25 \wedge 10 \leq y_t \leq 15))$. The control input to the model is the acceleration. The bounded noise ω is given by an uniform distribution between 0.2 and -0.2 added to the x and y dimensions of the position.

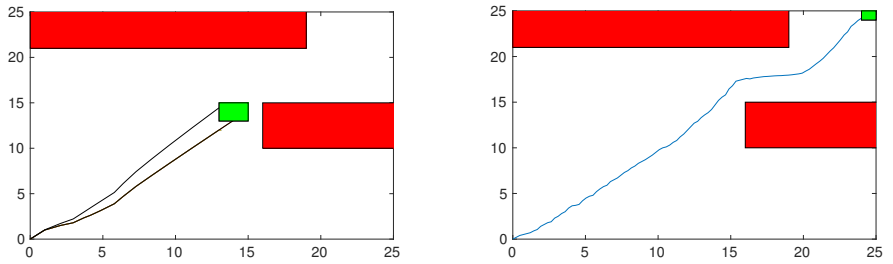


(a) Reference trajectory of STL controller (b) Tracking controller is not safe

Fig. 4. The open loop STL controller synthesis results into barely satisfying trajectory and the tracking controller is unable to prevent the vehicle from colliding with the obstacles. The obstacles are shown in red and the final destination region in green.

Figure 4(a) shows the trajectory of the robot obtained by an open loop STL controller in a noise-free environment. The resulting trajectory of the robot correctly satisfies the specification; however, there is little tolerance for error in the trajectory as it almost grazes past one of the obstacles. This is a consequence of how constraint solvers work in general. MILP solvers are good at finding a satisfying instance for given set of constraints but they are likely to find barely satisfying instances than robustly satisfying trajectories. In fact, the introduction of noise ω into the robot dynamics causes the robot to crash into one of the

obstacles, as shown in Figure 4(b). A traditional tracking controller fails to safely follow the reference trajectory in presence of noise.



(a) Robust controller (b) Proposed nested controller

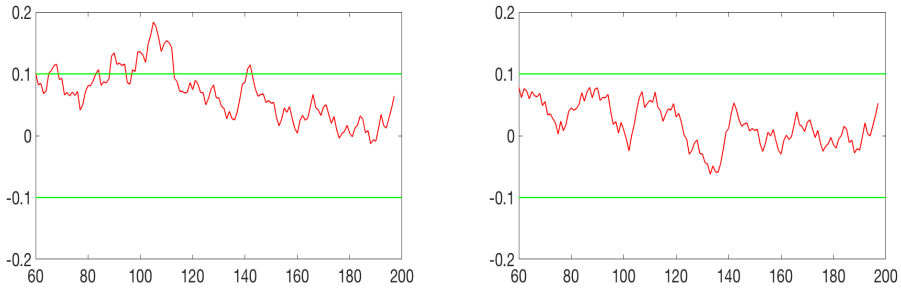
Fig. 5. Robust controller with reduced noise is able to synthesize a safe controller for relaxed specification – the uncertainty region illustrates the conservativeness of the robust controller synthesis method. It fails to solve the original problem. The nested controller is able to generate a safe controller even in the presence of noise.

The offline robust controller synthesis does not find a feasible safe controller. So, we relax the specification to $F_{[0,5]}(13 \leq x_t \leq 15 \wedge 13 \leq y_t \leq 15) \wedge G_{[0,5]}(\neg(0 \leq x_t \leq 19 \wedge 21 \leq y_t \leq 25) \wedge \neg(16 \leq x_t \leq 25 \wedge 10 \leq y_t \leq 15))$ requiring the vehicle to reach the unit square region around $15, 15$ instead of $25, 25$. We also reduced the noise to 0.1. We plot the resultant trajectory and the bounds on the uncertainty region around the trajectory in Figure 5. This illustrates how the robust controller conservatively models noise during offline synthesis and fails to find a safe controller for the original specification. The proposed nested controller synthesis approach can find a safe controller in 8min 39s for the original specification and the noise model, and a significant fraction of this runtime (3m 46s) is spent in formulating the MILP problem.

6.2 Case Study 2: Smart Grid Control

Our second case study is the smart grid model described in [26]. Each grid area contains a turbine, a generator and a governor. An automatic generation control (AGC) regulates the grid frequency using a proportional integral control. The AGC also ensures that the net interchange power between neighboring areas is maintained at predefined values. The Area Control Error (ACE) measures the difference between the predefined and actual electrical generation within an area while compensating for frequency differences. The system is described using a 13×13 dimensional A matrix and a 12×4 dimensional B matrix with two sources of noise and two control inputs. Our controller synthesizes both the control inputs while responding to changes in both sources of noise. Our model also requires that the magnitude of the control input to the system stay bounded by 0.6 and should evolve slowly with no more than a difference of 0.2 between two control

inputs. A specification of interest is to ensure that the absolute value of the ACE falls below 0.1 within 60 time units. A tracking controller is unable to satisfy the specification, as shown in Figure 6(a). We synthesize a nested STL-feedback controller for holding the absolute value of ACE below 0.1 against perturbations in the area-wise power demand. The synthesis of nested controller took 11m 28s. Figure 6(b) shows that the nested controller satisfies the specification despite the noise.



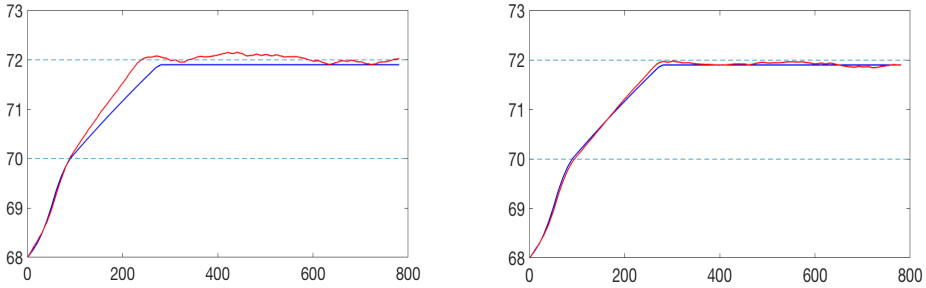
(a) Tracking controller (b) Nested controller

Fig. 6. Tracking controller is unable to maintain safety while the proposed nested controller keeps the system safe even in the presence of noise.

6.3 Case Study 3: Indoor Climate Control

Indoor climate control is a well-studied benchmark [25, 35] against which controllers have been designed using STL specifications. In this benchmark, a building with 4 rooms is modeled using a resistor-capacitor network. The rate of change of temperature of the i^{th} room depends on the difference between the temperature of this room and its neighboring rooms, the air flow into the room, the heat dissipation from windows, and the heat noise within the room from biological and electro-mechanical entities. While the original system is nonlinear, Euler's discretization method can be used to obtain a linear discrete-time system. We use such a linearization presented in [25] and also use their additive uncertainty model. The specification for controller synthesis is to "maintain a comfortable room temperature whenever the room is occupied". Formally, the specification can be written as a persistence STL property $F_{[0,T_{settle}]}G_{[0,T_{max}]}(T_t > 72 + \delta \wedge T_t < 72 - \delta)$ where $T_{settle} = 250, T_{max} = 500, \delta = 0.1$ in our experiments. Figure 7 shows the results obtained using the tracking controller synthesis and the nested controller synthesis method proposed in the paper. The synthesis of nested controller took 14m 24s. While the tracking controller is unable to satisfy the specification, the synthesized nested controller performs well in presence of

runtime noise. Figure 7(b) shows a sample run of the system with the synthesized nested controller.



(a) Tracking Controller (b) Nested controller

Fig. 7. Indoor Climate Controller with the persistence specification to reach a target temperature zone and stay within it. Nested controller is able to satisfy the specification while tracking controller cannot do so. The robust control synthesis could not generate a controller to keep the temperature within the tight bounds in the specification.

7 Conclusion

We proposed a novel approach to generate provably correct controllers of stochastic linear dynamical systems for STL specifications. Our approach decomposes the synthesis problem into orthogonal subproblems of meeting the STL specification, and noise-cancellation. It uses the duality between a system and its proof of correctness to compose their solutions and construct a safe nested controller. We first synthesize a stabilizing controller to reject noise at runtime, and then use its stability guarantee to formulate a less conservative robust open-loop controller synthesis problem for STL specifications using mixed integer linear programming. We experimentally validated the effectiveness of the proposed controller synthesis approach on a set of case-studies, and compared it with robust and tracking control methods. The proposed nested controller is less conservative than robust controllers, and is guaranteed to maintain safety in contrast to tracking controllers. In future work, we are investigating extensions to parametric systems where the guarantee from identical individual controllers for sub-systems can be used to synthesize a higher-level supervisory safe controller.

Acknowledgements The authors acknowledge support from the National Science Foundation (NSF) Cyber-Physical Systems #1740079 project, NSF Software & Hardware Foundation #1750009 and #1438989 projects, US ARL Cooperative Agreement W911NF-17-2-0196, and DARPA under contract FA8750-16-C-0043.

References

1. Abate, A., Prandini, M., Lygeros, J., Sastry, S.: Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. Automatica **44**(11), 2724–2734 (2008)
2. Akametalu, A.K., Fisac, J.F., Gillula, J.H., Kaynama, S., Zeilinger, M.N., Tomlin, C.J.: Reachability-based safe learning with gaussian processes. In: 53rd IEEE Conference on Decision and Control. pp. 1424–1431. IEEE (2014)
3. Bellman, R., Bellman, R.E., Bellman, R.E.: Introduction to the mathematical theory of control processes, vol. 2. IMA (1971)
4. Berkenkamp, F., Schoellig, A.P.: Safe and robust learning control with gaussian processes. In: Control Conference (ECC), 2015 European. pp. 2496–2501. IEEE (2015)
5. Blanchini, F.: Set invariance in control. Automatica **35**(11), 1747–1767 (1999)
6. Bogomolov, S., Schilling, C., Bartocci, E., Batt, G., Kong, H., Grosu, R.: Abstraction-based parameter synthesis for multiaffine systems. In: Piterman, N. (ed.) Hardware and Software: Verification and Testing. pp. 19–35. Springer International Publishing, Cham (2015)
7. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear matrix inequalities in system and control theory, vol. 15. Siam (1994)
8. Cassandras, C.G., Lygeros, J.: Stochastic hybrid systems, vol. 24. CRC Press (2006)
9. Dang, T., Dreossi, T., Piazza, C.: Parameter synthesis through temporal logic specifications. In: Bjørner, N., de Boer, F. (eds.) FM 2015: Formal Methods. pp. 213–230. Springer International Publishing, Cham (2015)
10. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: FORMATS. pp. 92–106 (2010)
11. Eggers, A., Fränzle, M., Herde, C.: Sat modulo ode: A direct sat approach to hybrid systems. In: Cha, S.S., Choi, J.Y., Kim, M., Lee, I., Viswanathan, M. (eds.) Automated Technology for Verification and Analysis. pp. 171–185. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
12. Fainekos, G.E., Girard, A., Kress-Gazit, H., Pappas, G.J.: Temporal logic motion planning for dynamic robots. Automatica **45**(2), 343–352 (Feb 2009). https://doi.org/10.1016/j.automatica.2008.08.008, http://dx.doi.org/10.1016/j.automatica.2008.08.008
13. Fainekos, G.E., Pappas, G.J.: Robustness of temporal logic specifications for continuous-time signals. Theoretical Computer Science **410**(42), 4262–4291 (2009)
14. Fan, C., Mathur, U., Mitra, S., Viswanathan, M.: Controller synthesis made real: Reach-avoid specifications and linear dynamics. In: CAV 2018. Springer International Publishing
15. Haddad, W.M., Chellaboina, V.: Nonlinear dynamical systems and control: a Lyapunov-based approach. Princeton University Press (2011)
16. Huang, Z., Wang, Y., Mitra, S., Dullerud, G.E., Chaudhuri, S.: Controller synthesis with inductive proofs for piecewise linear systems: An smt-based algorithm. In: 2015 54th IEEE Conference on Decision and Control (CDC). pp. 7434–7439 (Dec 2015)
17. Jha, S., Raman, V.: Automated synthesis of safe autonomous vehicle control under perception uncertainty. In: NASA Formal Methods Symposium. pp. 117–132. Springer (2016)

18. Jha, S., Raman, V.: On optimal control of stochastic linear hybrid systems. In: Formal Modeling and Analysis of Timed Systems: 14th International Conference, FORMATS 2016, Proceedings. pp. 69–84 (2016). https://doi.org/10.1007/978-3-319-44878-7_5
19. Jha, S., Raman, V., Sadigh, D., Seshia, S.A.: Safe autonomy under perception uncertainty using chance-constrained temporal logic. Journal of Automated Reasoning **60**(1), 43–62 (Jan 2018). https://doi.org/10.1007/s10817-017-9413-9, https://doi.org/10.1007/s10817-017-9413-9
20. Kautsky, J., Nichols, N.K., Van Dooren, P.: Robust pole assignment in linear state feedback. International Journal of control **41**(5), 1129–1155 (1985)
21. Kloetzer, M., Belta, C.: A fully automated framework for control of linear systems from temporal logic specifications. IEEE Transactions on Automatic Control **53**(1), 287–297 (Feb 2008). https://doi.org/10.1109/TAC.2007.914952
22. Kong, S., Gao, S., Chen, W., Clarke, E.: dreach: δ-reachability analysis for hybrid systems. In: Baier, C., Tinelli, C. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. pp. 200–205. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
23. Koutsoukos, X., Riley, D.: Computational methods for reachability analysis of stochastic hybrid systems. In: HSCC, pp. 377–391. Springer (2006)
24. Liu, J., Prabhakar, P.: Switching control of dynamical systems from metric temporal logic specifications. In: IEEE International Conference on Robotics and Automation (2014)
25. Maasoumy, M., Razmara, M., Shahbakhti, M., Vincentelli, A.S.: Handling model uncertainty in model predictive control for energy efficient buildings. Energy and Buildings **77**, 377 – 392 (2014). https://doi.org/https://doi.org/10.1016/j.enbuild.2014.03.057, http://www.sciencedirect.com/science/article/pii/S0378778814002771
26. Maasoumy, M., Sanandaji, B.M., Sangiovanni-Vincentelli, A., Poolla, K.: Model predictive control of regulation services from commercial buildings to the smart grid. In: American Control Conference (ACC), 2014. pp. 2226–2233. IEEE (2014)
27. Maler, O., Nickovic, D., Pnueli, A.: Real time temporal logic: Past, present, future. In: International Conference on Formal Modeling and Analysis of Timed Systems. pp. 2–16. Springer (2005)
28. Mitchell, I., Tomlin, C.J.: Level set methods for computation in hybrid systems. In: International Workshop on Hybrid Systems: Computation and Control. pp. 310–323. Springer (2000)
29. Mitchell, I.M., Bayen, A.M., Tomlin, C.J.: A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. IEEE Transactions on automatic control **50**(7), 947–957 (2005)
30. Ouaknine, J., Worrell, J.: Some recent results in metric temporal logic. In: International Conference on Formal Modeling and Analysis of Timed Systems. pp. 1–13. Springer (2008)
31. Pontryagin, L.: Optimal control processes. Usp. Mat. Nauk **14**(3) (1959)
32. Prabhakar, P., García Soto, M.: Formal synthesis of stabilizing controllers for switched systems. In: Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control. pp. 111–120. HSCC '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3049797.3049822, http://doi.acm.org/10.1145/3049797.3049822
33. Prajna, S., Jadbabaie, A., Pappas, G.J.: A framework for worst-case and stochastic safety verification using barrier certificates. Automatic Control, IEEE Transactions on **52**(8), 1415–1428 (2007)

34. Prandini, M., Hu, J.: Stochastic reachability: Theory and numerical approximation. Stochastic hybrid systems, Automation and Control Engineering Series **24**, 107–138 (2006)
35. Raman, V., Donz, A., Maasoumy, M., Murray, R.M., Sangiovanni-Vincentelli, A., Seshia, S.A.: Model predictive control with signal temporal logic specifications. In: 53rd IEEE Conference on Decision and Control. pp. 81–87 (Dec 2014). https://doi.org/10.1109/CDC.2014.7039363
36. Sadigh, D., Kapoor, A.: Safe control under uncertainty with probabilistic signal temporal logic. In: Robotics: Science and Systems XII (2016), http://www.roboticsproceedings.org/rss12/p17.html
37. Schrmann, B., Althoff, M.: Optimal control of sets of solutions to formally guarantee constraints of disturbed linear systems. In: 2017 American Control Conference (ACC). pp. 2522–2529 (May 2017)
38. Seto, D., Krogh, B.H., Sha, L., Chutinan, A.: Dynamic control system upgrade using the simplex architecture. IEEE Control Systems **18**(4), 72–80 (1998)
39. Summers, S., Kamgarpour, M., Lygeros, J., Tomlin, C.: A stochastic reach-avoid problem with random obstacles. In: Proceedings of the 14th international conference on Hybrid systems: computation and control. pp. 251–260. ACM (2011)
40. Tabuada, P., Pappas, G.J.: Linear time logic control of discrete-time linear systems. IEEE Transactions on Automatic Control **51**(12), 1862–1877 (2006)
41. Wongpiromsarn, T., Topcu, U., Murray, R.M.: Receding horizon temporal logic planning. IEEE Transactions on Automatic Control **57**(11), 2817–2830 (Nov 2012). https://doi.org/10.1109/TAC.2012.2195811
42. Yordanov, B., Tumova, J., Cerna, I., Barnat, J., Belta, C.: Temporal logic control of discrete-time piecewise affine systems. IEEE Transactions on Automatic Control **57**(6), 1491–1504 (2012)