Formal Methods for Controlling Dynamical Systems

Calin Belta
Mechanical Engineering, Boston University, Boston, MA, USA

Abstract

In control theory, complicated dynamics such as systems of (nonlinear) differential equations are mostly controlled to achieve stability. This fundamental property, which can be with respect to a desired operating point or a prescribed trajectory, is often linked with optimality, which requires minimization of a certain cost along the trajectories of a stable system. In formal methods, rich specifications, such as formulas of temporal logics, are checked against simple models of software programs and digital circuits, such as finite transition systems. With the development and integration of cyber physical and safety critical systems, there is an increasing need for computational tools for verification and control of complex systems from rich, temporal logic specifications. The current approaches to formal synthesis of (optimal) control strategies for dynamical systems can be roughly divided in two classes: abstraction-based and optimization-based methods. In this entry, we provide a short overview of these techniques.

Keywords

Formal verification · Model checking · Hybrid systems · Optimal control

Introduction

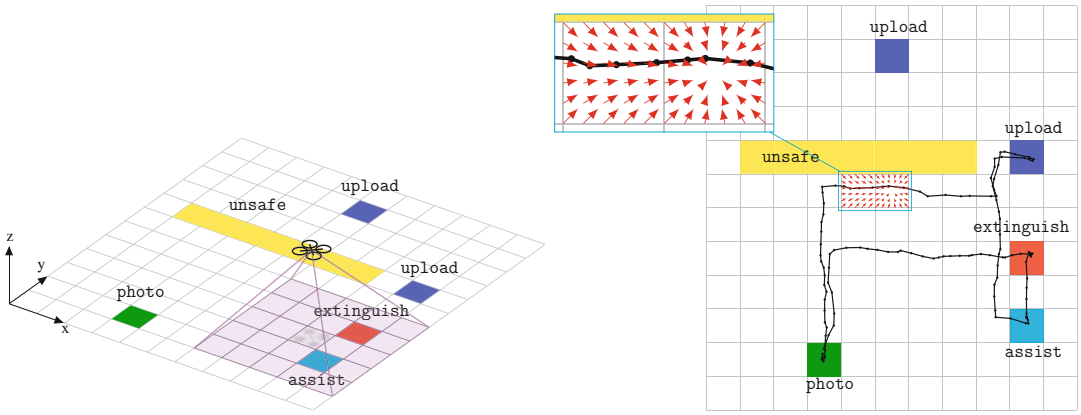
Temporal logics, such as computation tree logic (CTL) and linear temporal logic (LTL), have been customarily used to specify the correctness of computer programs and digital circuits modeled as finite-state transition systems. The problem of analyzing such a model against a temporal logic formula, known as formal analysis or model checking, has received a lot of attention during the past 40 years, and several efficient algorithms and software tools are available. The formal synthesis problem, in which the goal is to design or control a system from a temporal logic specification, has not been studied extensively until a few years ago.

Control and optimal control are mature research areas with many applications. They cover a large spectrum of systems, including (weighted) finite deterministic transition systems (i.e., graphs for which available transitions can be deterministically chosen at every node), finite purely nondeterministic systems (where an action at a state enables several transitions and their probabilities are not known), finite Markov decision processes (MDP), and systems with infinite state and control sets. For the latter, optimal control problems usually involve costs penalizing the deviation of the state from a reference trajectory and the control effort.

The connection between (optimal) control and formal methods is an intriguing problem with potentially high impact in several applications. By combining these two seemingly unrelated areas, the goal is to control the behavior of a system subject to correctness constraints. Consider, for example, an autonomous vehicle involved in a persistent surveillance mission in a disaster relief application, where dynamic service requests can only be sensed locally in a neighborhood around the vehicle (see Fig. 1). The goal is to accomplish the mission while at the same time maximizing the likelihood of servicing the local requests and possibly minimizing the energy spent during the motion. The correctness requirement can be expressed as a temporal logic formula (see the caption of Fig. 1), while the resource constraints translate to minimizing a cost over the feasible trajectories of the vehicle.

Formal Synthesis of Control Strategies

Current works on combining control and formal methods can be roughly divided into two



Formal Methods for Controlling Dynamical Systems, Fig. 1 (**a**) An autonomous air vehicle is deployed from a high level, temporal logic global specification over a set of static, known requests (photo and upload) occurring at the regions of a known environment, e.g., "Keep taking photos and upload current photo before taking another photo." This specification translates to the following LTL formula: **GF** photo $\wedge$ **G** (photo $\rightarrow$ (photo **U**($\neg$photo **U** upload))), where **G**, **F**, and **U** are the temporal operators Globally (Always), Future (Eventually), and Until; $\wedge$, $\rightarrow$, $\neg$ are Boolean operators for conjunction, implication, and negation, respectively. While moving in the environment, the vehicle can locally sense dynamically changing events, such as survivors, and fires, which generate (local) service requests, and unsafe areas, which need to be avoided. The goal is to accomplish the global mission while at the same time maximizing the likelihood of servicing the local requests and staying away from unsafe areas. (**b**) By using an accurate quad-rotor kinematic model, input-output linearizations/flat outputs, precise state information from a motion capture system, and control-to-facet results for linear and multi-affine systems, this problem can be (conservatively) mapped to a control problem for a finite transition system. This can be deterministic or nondeterministic if single-facet or multiple-facet controllers in the output space are used, respectively. (Example adapted from Ulusoy and Belta 2014)

main classes: automata-based methods and optimization-based methods.

Automata-Based Methods

Automata-based methods are based on the observation that a temporal logic formula, such as an LTL formula, can be mapped to an automaton in such a way that the language accepted by the automaton is exactly the language satisfying the formula. Depending on the desired expressivity of the specification language, these automata can be well-known finite state automata (FSA) (the acceptance condition is reaching a set of final states), Büchi automata (the acceptance condition is reaching a set of final states infinitely often), or, in the most general case, Rabin automata (the acceptance condition is to visit a set of "good" states infinitely often and a set of "bad" states finitely many times). For finite systems, the control problem reduces to a game on the product between a system, such as a transition system or an MDP, and the automaton obtained from the specification. The winning condition, which ensures correctness, is the Rabin (Büchi, FSA) acceptance condition of the automaton. The cost can be average reward/cost per stage and adapted objectives that reflect the semantics of the temporal logic, such as average reward/cost per cycle.

For infinite systems, automata-based approaches are, in general, hierarchical, two-level methods. The bottom level is a continuous-to-continuous abstraction procedure, in which the possibly large state space and complex dynamics are mapped to a low-dimensional output space with simple dynamics. The most used techniques are input-output linearization and differential flatness. For a differentially flat system, its state and control variables can be expressed as a function of its outputs and its derivatives. The top level is a partition-based, continuous-to-discrete abstraction procedure, in which the output and control

spaces are partitioned. The partition can be driven by the specification or by a prescribed accuracy of the approximation. The quotient of the partition is a finite system that is in some way equivalent with the original, infinite system. The most used notion of equivalence is bisimulation. An example is shown in Fig. 1. The 12-dimensional quad-rotor dynamics of the quad-rotor shown in the left are differentially flat with four flat outputs (position and yaw), and up to four derivatives of the flat output are necessary to compute the original state and input. A two-dimensional section of the partition of the four-dimensional output space is shown on the right, together with the assignment of a vector field in two adjacent cells. Note that the dynamics corresponding to these vector fields "treat" all the states in a cell "in the same way": in the cell on the left, all the states will leave in finite time through the right facet; in the cell on the right, all states will stay inside for all times. Informally, these correspond to the bisimilarity equivalence mentioned above. The quotient of the partition is a finite transition system that is controlled from the temporal logic specification. The cost can penalize the execution time, travelled distance, etc.

The method described above is conservative. If a solution (of the automaton game) is not found at the top level, this does not mean that a controller does not exist for the original continuous system. Intuitively, the partition, and therefore the abstraction, might be too rough. Conservativeness can be reduced by refining the partition. Numerous partition techniques that exploit the connection between the dynamics of the system and the geometry of the partition have been proposed. An example is shown in Fig. 2. An example illustrating the compromise between correctness and optimality is shown in Fig. 3.

The expensive process of constructing the abstraction can be avoided for both deterministic and stochastic systems. Specifically, a dynamic programming problem can be formulated over the product of the continuous-time, continuous-state system, and the specification automaton. Approximate dynamic programming (ADP) approaches have been shown to work for both linear and nonlinear systems. Sampling-based policy iteration has also been used for optimal planning for a subclass of LTL specifications.

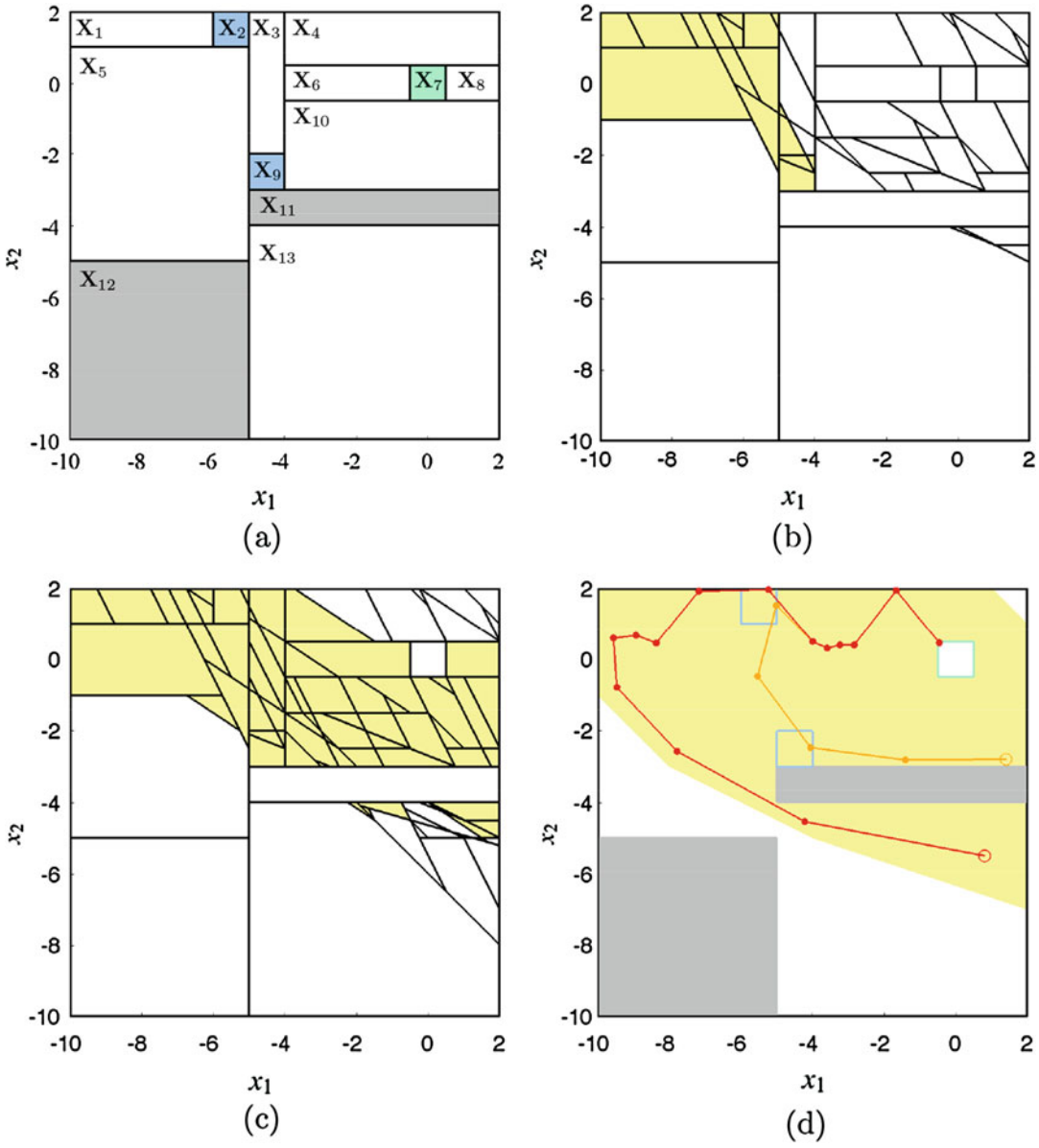
Optimization-Based Methods

There are roughly two classes of optimization-based methods for formal synthesis: mixed integer programming (MIP) methods and control barrier functions (CBF) methods.

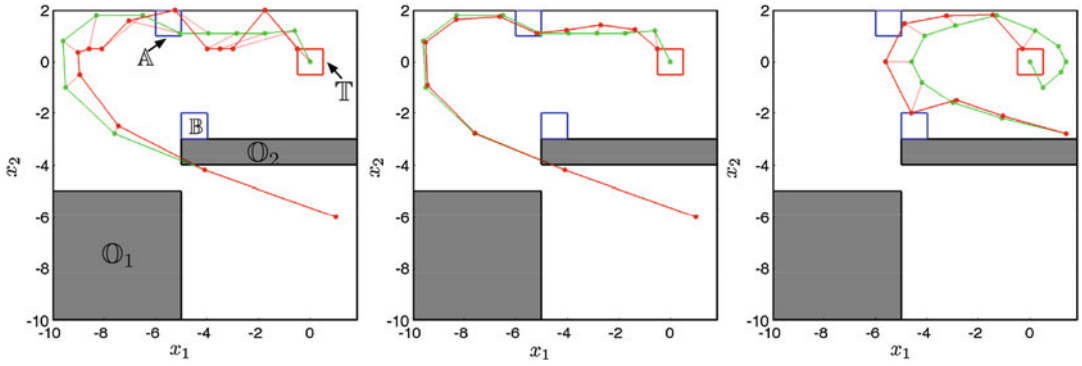
Central to the MIP-based methods are temporal logics with semantics over finite-time signals, such as signal temporal logic (STL) and metric temporal logic (MTL). For simplicity, in this article we focus on STL. The main difference between STL and LTL (see the caption of Fig. 1 for an example of an LTL formula) is that STL temporal operators are explicit (see Fig. 4 for an example of an STL formula). In addition to Boolean semantics, in which signals either satisfy or violate a formula, STL has quantitative semantics, which allow to assess the robustness of satisfaction. Specifically, given a formula ϕ and a signal x, the robustness $\rho(\phi, x)$ quantifies how well x satisfies ϕ. The more x satisfies ϕ, the larger $\rho(\phi, x)$ is (if x satisfies ϕ in Boolean semantics, then $\rho(\phi, x) > 0$). The more x violates ϕ, the smaller $\rho(\phi, x)$ is (if x violates ϕ in Boolean semantics, then $\rho(\phi, x) < 0$).

It can be shown that Boolean satisfaction of STL (MTL) formulas over linear predicates in the state x of a system can be mapped to the feasibility part of a mixed integer linear program (MILP) (i.e., a set of linear equalities and inequalities over the state and an additional set of integers). This observation implies that controlling a linear (or almost linear, such as piecewise affine, mixed logical, etc.) system, such that a linear or quadratic cost is optimized while satisfying STL formulas over linear predicates over its state, maps to solving a mixed integer linear program (MILP) or mixed integer quadratic program (MIQP), for which there exist computationally efficient solvers.

Moreover, it can be shown that the robustness function ρ is linear in state and the additional integer variables. Therefore, if robustness is



Formal Methods for Controlling Dynamical Systems, Fig. 2 A discrete-time double integrator system $x[t+1] = Ax[t] + Bu[t]$, with $A = [1\ 1;\ 0\ 1]$, $B = [0.5;\ 1]$, and $u \in [-2, 2]$, moving in a planar environment (**a**) partitioned into $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{13}$ is required to satisfy the following specification: "Visit region $\mathbb{A} = \mathbf{X}_2$ or region $\mathbb{B} = \mathbf{X}_9$, and then the target region $\mathbb{T} = \mathbf{X}_7$, while always avoiding obstacles $\mathbb{O}_1 = \mathbf{X}_{11}$ and $\mathbb{O}_2 = \mathbf{X}_{12}$, which translates to the syntactically co-safe LTL (scLTL) formula: $((\neg\mathbb{O}_1 \wedge \neg\mathbb{O}_2)\,\mathbf{U}\,\mathbb{T}) \wedge (\neg\mathbb{T}\mathbf{U}\,(\mathbb{A} \vee \mathbb{B}))$. Iterations 50, 80, and 108 of an automata-guided iterative partitioning procedure for the computation of the maximal set of satisfying initial states (and corresponding control strategies) are shown in (**b**), (**c**), and (**d**), respectively (the sets of satisfying initial states are shown in yellow). For linear dynamics and a particular choice of polytope-to-polytope controllers, the procedure is complete (i.e., guaranteed to terminate and to find the maximal set). The yellow region from (**d**) is the maximal set of satisfying initial states. Sample trajectories of the closed loop system are shown in **d**, where the initial states are marked by circles. The trajectories coincide in the last six steps before they reach the target region. (Example adapted from Belta et al. 2017)



Formal Methods for Controlling Dynamical Systems, Fig. 3 For the system described in Fig. 2, in addition to satisfying the temporal logic specification, the system is required to minimize a quadratic cost that penalizes the Euclidean distance from desired state and control trajectories, which are available to the system over a short finite-time horizon N. The reference trajectory is shown in green (satisfying in the left and middle and violating in the right), and the trajectory of the controlled system is shown in red. Pairs of points on the reference and controlled trajectory corresponding to the same time are connected. The left and middle cases correspond to increasing values of the horizon N for the same reference trajectory. Note that, in the situation shown in the right, where the reference trajectory violates the correctness specification, the controller "tries to compromise" between correctness and optimality. (Example adapted from Belta et al. 2017)

added (possibly weighted by a constant) to the cost defined above, the optimization problem remains a MILP (MIQP). Adding robustness in the cost allows to compromise between correctness and optimality (see Fig. 4 and its caption.)

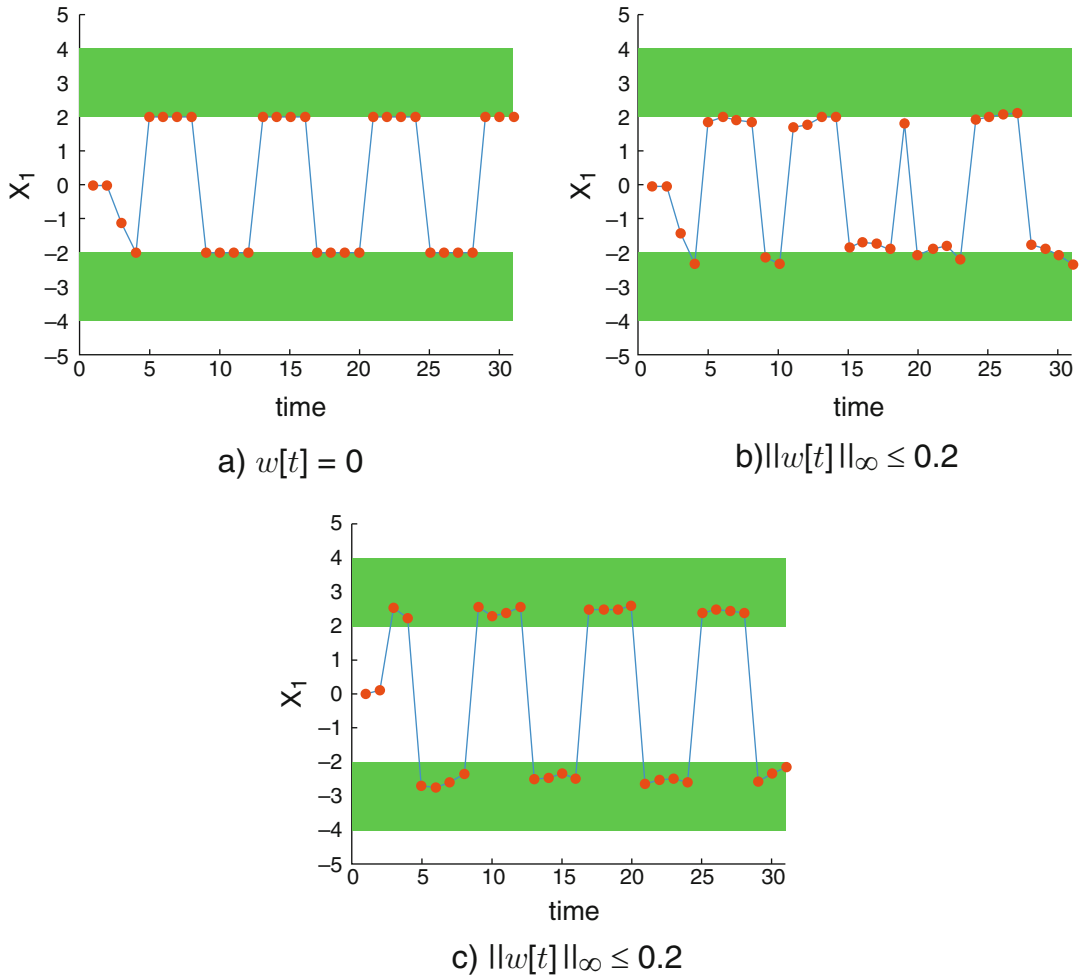
The main advantage of MIP-based methods is the seamless combination of correctness and optimality, with the added feature of robustness to satisfaction. Another advantage is scalability. Such methods produced results for systems with hundreds of state variables in seconds. The main limitation of such methods is that they are constrained to linear dynamics.

CBF-type methods address some of the limitations of the MIP-based methods. A pictorial representation of a CBF-based approach is shown in Fig. 5. In short, such methods work for a particular class of nonlinear control systems, called affine control systems, which is large enough to include many mechanical systems, such as unicycles, cars, etc., costs that are quadratic in controls, linear control constraints, and safety constraints expressed as set forward invariance. By dividing the time interval of interest into smaller time intervals, a nonlinear optimization problem can be reduced to a set of quadratic programs (QP). Richer, temporal logic correctness constraints can also be incorporated in this framework.

Summary and Future Directions

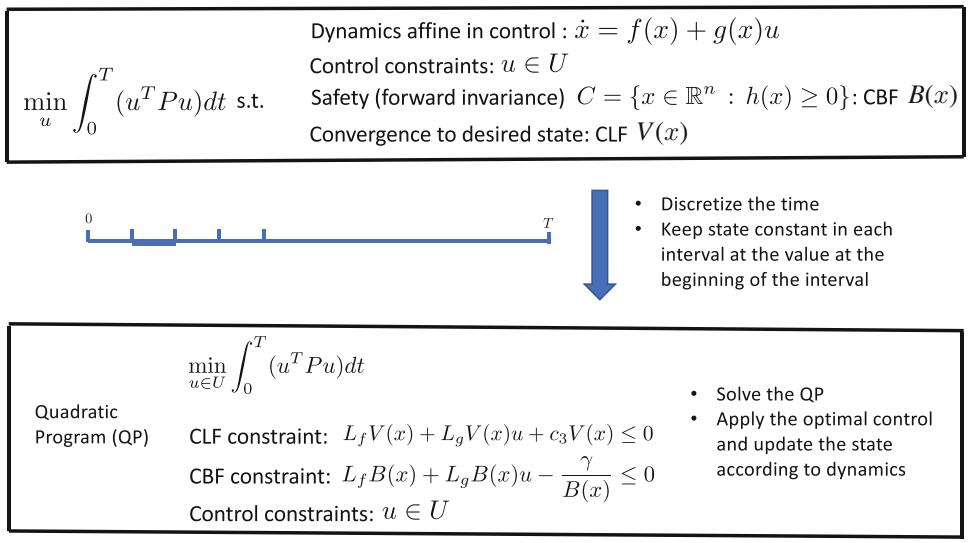
While provably correct, automata-based approaches to formal synthesis of control strategies are computationally expensive. Current research that addresses this limitation is focused on identifying system properties that facilitate the construction of the abstraction (e.g., passivity, disipativity) (Coogan and Arcak 2017) and compositional synthesis and assume guarantee-type approaches (Kim et al. 2017).

Optimization-based approaches based on mixed integer programming scale can quantify satisfaction and can compromise between correctness and optimality. However, they are restricted to linear dynamics. Optimization-based approaches based on control barrier functions and control Lyapunov functions can be used with nonlinear systems that are affine in controls. One of the main limitations of these types of approaches is that the optimization problems can



Formal Methods for Controlling Dynamical Systems, Fig. 4 A planar discrete-time (integrator) linear system $x[t+1] = Ax[t] + Bu[t] + w[t]$ with state $x = (x_1, x_2)$, control u, $A = [1\ \ 0.5;\ \ 0\ \ 0.8]$, $B = [0;\ \ 1]$ is affected by noise w. The specification requires that x_1 oscillate between $2 \leq x_1 \leq 4$ and $-4 \leq x_1 \leq -2$, with each interval being visited at least once within any five consecutive time steps. The corresponding STL formula is $\mathbf{G}_{[0,\infty)}\left(\mathbf{F}_{[0,4]}((x_1 \geq 2) \wedge (x_1 \leq 4))\right) \wedge \left(\mathbf{F}_{[0,4]}((x_1 \geq -4) \wedge (x_1 \leq -2))\right)$. Here, $\mathbf{F}_{[t_1,t_2]}\psi$ requires that ψ is eventually satisfied within $[t_1, t_2]$, while $\mathbf{G}_{[t_1,t_2]}\psi$ means that ψ is true for all times in $[t_1, t_2]$. The control effort $\sum_{t=0}^{H-1} |u|^2$ is minimized in (**a**) and (**b**), while in (**c**) the "robust" version of the cost $\sum_{t=0}^{H-1} -M(\rho - |\rho|) + |u|^2$ is considered (H is a time horizon that is large enough to be able to decide the truth value of the formula). It can be seen that, if only the control effort is minimized, the resulting strategy is not robust to noise. If robustness is included in the cost, then the produced trajectory satisfies the specification even if noise is added to the system. (Example adapted from Sadraddini and Belta 2015)

easily become infeasible, especially when many constraints (state limitations, control constraints, CBF and CLF constraints) become active. One possible approach to this problem is to soften some constraints that are not critical, such as those induced by CLFs. Another possible approach would be to use machine learning techniques to increase feasibility. For example, for a robot moving in an environment cluttered with obstacle, the configuration space can be sampled close to the obstacles, and the samples could be classifier depending on whether



Formal Methods for Controlling Dynamical Systems, Fig. 5 CBF-based approach to provably correct and optimal control synthesis: An affine control system is required to minimize a quadratic cost while converging to a desired final state and satisfying a safety specification expressed as the forward invariance of a set C and polyhedral control constraints U. Assuming that a CBF $B(x)$ can be constructed for C, then safety is guaranteed if the CBF constraint is satisfied. If a control Lyapunov function (CLF) $V(x)$ can be constructed, then exponential convergence to the desired state is guaranteed if the CLF constraint is satisfied. By discretizing the time and keeping the state constant in each time interval, both constraints become linear in control, and the problem reduces to a set of QPs. L_f and L_g denote Lie derivatives along f and g, respectively. P is a positive definite matrix and c_3 and *gamma* are positive constants

the corresponding QP is feasible or not. The (differentiable) classifier could be used as an extra CBF. Another active area of research is integrating correctness specifications given in rich, temporal logic specifications. Very recent results (Lindemann and Dimarogonas 2019) show that STL specifications can be enforced using CBF.

Cross-References

- ▸ Discrete Event Systems and Hybrid Systems, Connections Between
- ▸ Motion Description Languages and Symbolic Control
- ▸ Supervisory Control of Discrete-Event Systems
- ▸ Validation and Verification Techniques and Tools

Recommended Reading

Two popular textbooks on formal methods for finite systems are Baier et al. (2008) and Clarke et al. (1999). Research monographs and textbooks that cover abstraction-based methods include Lee and Seshia (2015), Alur (2015), Tabuada (2009) and Belta et al. (2017). The reader interested in the mixed integer programming approach to the optimization-based techniques is referred to Karaman et al. (2008), Raman et al. (2014) and Sadraddini and Belta (2015). The metrics with quantitative semantics that enable such techniques were introduced in Maler and Nickovic (2004) and Koymans (1990). Optimization-based techniques using control Lyapunov functions and control barrier functions are covered in Galloway et al. (2013) and Ames et al. (2014).