Quantitative Implementation Strategies for Safety Controllers

Philipp J. Meyer\textsuperscript{2}, Matthias Rungger\textsuperscript{1}, Michael Luttenberger\textsuperscript{2}, Javier Esparza\textsuperscript{2}, and Majid Zamani\textsuperscript{1}

\textsuperscript{1} Professorship for Hybrid Control Systems
Department of Electrical and Computer Engineering
Technical University of Munich

\textsuperscript{2} Chair for Foundations of Software Reliability and Theoretical Computer Science
Department of Informatics
Technical University of Munich

Abstract. We consider the symbolic controller synthesis approach to enforce safety specifications on perturbed, nonlinear control systems. In general, in each state of the system several control values might be applicable to enforce the safety requirement and in the implementation one has the burden of picking a particular control value out of possibly many. We present a class of implementation strategies to obtain a controller with certain performance guarantees. This class includes two existing implementation strategies from the literature, based on discounted payoff and mean-payoff games. We unify both approaches by using games characterized by a single discount factor determining the implementation. We evaluate different implementations from our class experimentally on two case studies. We show that the choice of the discount factor has a significant influence on the average long-term costs, and the best performance guarantee for the symbolic model does not result in the best implementation. Comparing the optimal choice of the discount factor here with the previously proposed values, the costs differ by a factor of up to 50. Our approach therefore yields a method to choose systematically a good implementation for safety controllers with quantitative objectives.

Keywords: Symbolic Controller Synthesis; Discounted Payoff Games; Mean-Payoff Games; Safety Controller

1 Introduction

The symbolic controller synthesis approach\textsuperscript{21} has gained considerable attention within the Cyber-Physical System research community. The approach constructs a finite-state abstraction (a.k.a. symbolic model) of a continuous-space, continuous-time system, and reduces the control problem over the continuous

\textsuperscript{*} Partially supported by the DFG project “Game-based Synthesis for Industrial Automation” and the Institute for Advanced Study of the Technische Universität München.
system to the computation of a winning strategy in a finite-state game. Following this paradigm, it is guaranteed by construction that the winning strategy delivers a correct controller for the concrete system, i.e., the refined controller enforces the given specification over the original system.

In this work, we follow the symbolic approach to enforce safety specifications for perturbed, nonlinear control systems. Safety specifications are one of the most fundamental requirements and are ubiquitous in the analysis and control of dynamical systems [24]. Given a system, the main goal is to synthesize a controller that enforces the system to evolve within a safe set for all times. Via the symbolic approach, the possible behaviours of the continuous system are safely over-approximated leading to a 2-player safety game over a finite (but large) arena. The controller plays against an adversary that represents the nondeterminism, which arises from the discretization of the continuous system and possibly from model uncertainties and perturbations. From a theoretical perspective, solving safety games over finite arenas is trivial: one simply has to identify and then (iteratively) deactivate all actions which allow the adversary to force a play out of the safety region. After deactivating all dangerous actions, the controller is essentially free to play in any way within the winning region. It is well-known that there exists a unique maximal (or permissive or nondeterministic) optimal winning strategy, which assigns to each state all possible actions available to the controller that guarantee that the system stays within the safety region. In general, there might be multiple actions available to the controller and the natural question is whether we can make use of this flexibility to further optimize the controller with respect to some quantitative objectives. Examples of such quantitative conditions that arise quite naturally in the control of physical systems are:

- minimization of the number of input switches (lazy controller);
- minimization of the deviation from a reference value (reference tracking);
- minimization of energy consumption;
- minimization of relative change between inputs (avoiding jerkiness).

In this paper, we follow ideas in [19,13] and use possibly discounted mean-payoff games to synthesize deterministic controllers that enforce the system to stay within the safety region and minimize certain cost functions. Specifically, we consider normalized \( \lambda \)-discounted cost functions (with \( \lambda \in [0,1] \))

\[
(1 - \lambda) \sum_{i=0}^{\infty} \lambda^i c(i)
\]

and limit-average cost functions

\[
\limsup_{T \to \infty} \frac{1}{1 + T} \sum_{i=0}^{T} c(i)
\]

where \( c(i) \) represents a certain cost (to be defined later) that is associated with the system behavior at time \( i \). We follow standard nomenclature, and call the re-
sulting games whose goal is to minimize those cost functions $\lambda$-discounted payoff safety games ($\lambda$-DPSG) and mean-payoff safety game (MPSG), respectively.

We set up an effective tool chain for symbolic controller synthesis to enforce safety specifications under mean-payoff objectives. In the first step, we employ SCOTS [18] to obtain the symbolic model; subsequently, we compute the winning strategies for the resulting DPSGs and MPSGs with the state-of-the-art GPU solver described in [14]; finally, the performance of the controller in the continuous-time system is evaluated by means of simulations conducted with MATLAB. We measure the performance by the limit-average cost (2) for controllers obtained from both DPSGs and MPSGs.

We present two case studies and conduct a systematic analysis to illustrate the effects of different choices of cost functions and discount parameters. In the first case study, we consider the regulation of the room temperature and the humidity of the air in a building equipped with a heating, ventilation and air conditioning (HVAC) system introduced in [5,10]. In the second example, we consider regulation of the popular cart-pole system, e.g. studied in [17]. For each of the different objectives, we associate both $\lambda$-discounted cost functions and limit-average cost functions, and solve the corresponding games. Our main finding is that performance guarantees come at a price: while the limit-average cost function provides the strongest, global performance guarantee on the actual limit-average cost over the complete execution of the controller, it typically leads to controllers that perform poorly. Using discounted cost functions leads to controllers with only weaker limit-average cost guarantees, which however achieve better limit-average costs in the simulation. We also present experimental results about the values of $\lambda$ leading to controllers with the best limit-average costs.

Related work. Quantitative objectives are a natural requirement in specifying the desired system behavior. Consequently, there exist several different approaches to augment language containment specifications, e.g. defined in linear temporal logic, with quantitative properties [8,12,13,15,22,23,24]. A considerable number of approaches focus on finite horizon specifications, like reachability [8,12,15,22] or co-safety [9] specifications, which leads to an optimization of the transient behavior. Our work, on the contrary, focuses on optimizing the average, long-term behavior. This is also the focus of [23,24]. However, the approach of these papers is incomparable with ours: while their work is applicable not only to safety specifications, but also to more general temporal logic specifications, it is restricted to specific classes of systems. In particular, [23] is restricted to deterministic symbolic models, and [24] is restricted to mixed logical dynamical systems and differentially flat systems. None of these classes contains the control systems considered in this work. While DPSG and MPSG in the context of symbolic synthesis have been considered in [13] and [19], this work is the first one to systematically study the effects of different cost functions and discount factors on the performance of the resulting refined controllers.

Structure of the paper. Section 2 presents the control problem. Section 3 presents the symbolic approach to controller synthesis. Section 4 presents the game-
theoretic approach to quantitative safety synthesis. Section 5 describes the cost functions and our two case studies: the HVAC system and the cart-pole system. Section 6 presents the results of our experimental evaluation. Section 7 summarizes our findings.

2 Control Systems and Objectives

We study nonlinear control systems of the form

$$\dot{\xi}(t) \in f(\xi(t), u) + [-w, w]$$

where $f$ is given by $f : \mathbb{R}^n \times \bar{U} \to \mathbb{R}^n$ and $\bar{U} \subseteq \mathbb{R}^m$. The vector $w = [w_1, \ldots, w_n] \in \mathbb{R}_{\geq 0}^n$ is a perturbation bound and denotes the hyper-interval

$$[-w, w] := [-w_1, w_1] \times \cdots \times [-w_n, w_n].$$

We can use the perturbations to take into account model uncertainties and other adversarial effects. Given an interval $I \subseteq \mathbb{R}$, we define a solution of (3) on $I$ under (constant) input $u \in \bar{U}$ as an absolutely continuous function $\xi : I \to \mathbb{R}^n$ that satisfies (3) for almost every (a.e.) $t \in I$ [7].

In order to facilitate a digital controller implementation, which operates in discrete time, we consider the sampled behavior of the continuous-time system (3). Let $\tau \in \mathbb{R}_{>0}$ be the sampling time. We assume that the controller has access to the system state only at integer multiples of the sampling times $k\tau$, $k \in \mathbb{Z}_{\geq 0}$. Similarly, we assume that the control input is updated only at the sampling times $k\tau$, $k \in \mathbb{Z}_{\geq 0}$ and held constant otherwise. See Figure 1 for an illustration of this process. We cast the sampled behavior of (3) as a simple system [16].

For sets $A$ and $B$, a set-valued map from $A$ to $B$ is a set $f \subseteq A \times B$, written as $f : A \rightrightarrows B$, with $f(a) := \{ b \in B \mid (a, b) \in f \}$ for every $a \in A$.

![Fig. 1. Sample-and-hold implementation of a controller.](image-url)
Definition 1. A simple system (with initial states) \( S \) is a quadruple

\[
S := (X, X_0, U, F)
\]

where the state alphabet \( X \), the initial state alphabet \( X_0 \subseteq X \), and the input alphabet \( U \) are nonempty sets, and the transition function \( F \) is a set-valued map \( F : X \times U \rightrightarrows X \). The behavior of the simple system \( S \) is defined by

\[
\mathcal{B}(S) := \{(u, x) \in (U \times X)^{[0; T]} \mid x(0) \in X_0, \\
\quad \forall t \in [0; T-1] \; x(t+1) \in F(x(t), u(t)) \\
\quad \text{and } (T < \infty \implies F(x(T-1), u(T-1)) = \emptyset)\}.
\]

We say that a simple system \( S \) of the form (4) represents the \( \tau \)-sampled behavior of (3), if \( X = \mathbb{R}^n \), \( U = \bar{U} \), and the transition function satisfies for all arguments

\[
F(x, u) = \{x' \mid \exists \text{ a solution } \xi \text{ of (3) on } [0, \tau] \text{ under } u : \xi(0) = x \land \xi(\tau) = x'\}.
\]

A safety specification for (4) is simply a set \( Z \subseteq U \times X \). A simple system \( S \) together with a specification \( Z \) constitute a safety problem \((S, Z)\).

Even though, in this framework, we obtain safety guarantees only with respect to the \( \tau \)-sampled behavior of (3), it is straightforward to work with a slightly modified safety specification in the synthesis procedure, which then implies safety guarantees with respect to the continuous-time behavior of (3), see e.g. [11].

A solution of a safety problem \((S, Z)\), where \( S \) is given in (4), is a set-valued map \( K : X \rightrightarrows U \) (i.e. \( \forall x \in X : K(x) \subseteq U \)) such that the closed loop \( K \times S \), which is the simple system \((X, X_0, U, F_K)\), whose transition function is defined by

\[
F_K(x, u) := \{x' \in X \mid x' \in F(x, u) \land u \in K(x)\}
\]

satisfies the safety specification, i.e.,

\[
\mathcal{B}(K \times S) \subseteq Z^{[0; \infty]}.
\]

In this context, we refer to \( K \) as a controller for \( S \). The domain of a controller \( K \) is the set \( \mathcal{D}(K) := \{x \in X \mid K(x) \neq \emptyset\} \).

Let \( K \) be a solution of a safety problem \((S, Z)\) with \( S \) given in (4). An implementation of \( S \) and \( K \) is a function \( K_{\text{imp}} : X \times U \to U \) that satisfies \( K_{\text{imp}}(x, u) \in K(x) \) for all \( x \in \mathcal{D}(K) \) and \( u \in U \). The closed loop \( K_{\text{imp}} \times S \) resulting from the implementation \( K_{\text{imp}} \) is the simple system \( K_{\text{imp}} \otimes S := (X \times U, X_0 \times U, \{0\}, F_{\text{imp}}) \), whose transition function is defined by

\[
F_{\text{imp}}((x, u), 0) := \{(x', u') \in X \times U \mid x' \in F(x, u') \land u' = K_{\text{imp}}(x, u)\}.
\]

For the remainder, we identify the behavior of \( K_{\text{imp}} \otimes S \) with the set that results from the projection of \( \mathcal{B}(K_{\text{imp}} \otimes S) \) onto \((X \times U)^{[0; \infty]}\). Given this identification, it is straightforward to see that the inclusion

\[
\mathcal{B}(K_{\text{imp}} \otimes S) \subseteq \mathcal{B}(K \times S)
\]
holds. Hence, any closed loop $K_{\text{imp}} \otimes S$ resulting from an implementation $K_{\text{imp}}$ of $K$ satisfies the safety specification $Z$.

Subsequently, we assume we are given a cost function 

$$c : U \times X \times U \to \mathbb{R}$$

and a discounting factor $\lambda \in [0, 1]$ which we use to determine the implementation strategy. Specifically, we would like to find an implementation $K_{\text{imp}}$ of a solution $K$ of $(S, Z)$ that minimizes the worst-case, discounted cost function

$$J(K_{\text{imp}} \otimes S, c, \lambda) := \sup_{(u,x) \in B(K_{\text{imp}} \otimes S)} \left(1 - \lambda\right) \sum_{i=0}^{\infty} \lambda^i c(u(t), x(t), u(t + 1)).$$

For the case $\lambda = 1$, we consider the above function with the limit value for $\lambda \to 1^-$, where it is converges (see e.g. Appendix H of [6] for a proof) to the limit-average cost given by

$$J(K_{\text{imp}} \otimes S, c, 1) := \sup_{(u,x) \in B(K_{\text{imp}} \otimes S)} \limsup_{T \to \infty} \frac{1}{1 + T} \sum_{i=0}^{T} c(u(t), x(t), u(t + 1)).$$

The quadruple $(S, Z, c, \lambda)$ constitutes a valuated safety problem.

3 Symbolic Synthesis

Within the symbolic controller synthesis paradigm [16,21,25] a controller $K$ for $S$ is not computed directly, but a finite representation $\hat{S}$ of $S$ is used as a substitute in the synthesis process. The procedure is roughly summarized in three major steps: first, a finite representation $\hat{S}$, i.e., the symbolic model, of $S$ is computed; second, the synthesis problem is algorithmically solved with respect to $\hat{S}$; third, the obtained solution $\hat{K}$ is refined or transferred to a controller $K$ for $S$. The correctness of this approach is usually ensured by relating the plant $S$ with the symbolic model $\hat{S}$ by a system relation. In this work we follow [16], in which symbolic models are related with the plant via feedback refinement relations. Notably feedback refinement relations are appealing as they facilitate a particularly easy controller refinement procedure compared to other system relations [16], see also the controller refinement equation (11).

Let $S$ be given in (4). Consider a simple system

$$\hat{S} := (\hat{X}, \hat{X}_0, \hat{U}, \hat{F})$$

whose states $\hat{x} \in \hat{X}$, also referred to as cells, are subsets of $X$, i.e., $\hat{x} \subseteq X$, and whose input alphabet $\hat{U}$ is a subset of $U$. The system $\hat{S}$ is a symbolic model of $S$, if there exists a strict relation $Q \subseteq X \times \hat{X}$ so that for all $u \in \hat{U}$ and $(x, \hat{x}) \in Q$ we have

$$A relation R \subseteq A \times B is strict, if for every a \in A there exists b \in B so that (a,b) \in R.$$
inequality holds:

$$
\hat{F}(\hat{x}, u) \neq \emptyset \text{ implies } F(x, u) \neq \emptyset \text{ and } Q(F(x, u)) \subseteq \hat{F}(\hat{x}, u).
$$

The relation $Q$ is called a feedback refinement relation from $S$ to $\hat{S}$. The first condition ensures that every cell $\hat{x}$ that is related to an initial state of $S$ is an initial state of $\hat{S}$. The second condition ensures, that if a state-input pair $(\hat{x}, u)$ is non-blocking, i.e., $\hat{F}(\hat{x}, u) \neq \emptyset$, then $(x, u)$ is non-blocking for every related state $x$. Additionally, every cell $\hat{x}'$ that is related to a successor state $x' \in F(x, u)$ is also a successor of $(\hat{x}, u)$, i.e., $\hat{x}' \in \hat{F}(\hat{x}, u)$.

A safety specification $\hat{Z}$ for (9) is derived from a safety specification $Z$ for (4) by

$$
\hat{Z} := \{(u, \hat{x}) \in \hat{U} \times \hat{X} \mid \{u\} \times \hat{x} \subseteq Z\}. \quad (10)
$$

In the refinement of a controller for the symbolic model $\hat{S}$ to a controller for the concrete system $S$, the feedback refinement relation $Q$ is interpreted as a set-valued map $Q : X \rightrightarrows \hat{X}$ and is used to translate concrete states $x \in X$ to related abstract states $\hat{x} \in Q(x)$. The refined controller $K$ for $S$ is then simply given by the composition of the map $Q$ with the controller $\hat{K} : \hat{X} \rightrightarrows \hat{U}$ for $\hat{S}$ by $K := \hat{K} \circ Q$. In this context, $K$ is referred to as refined controller. The correctness of this procedure is ensured by the following result recalled from [16 Thm. VI.3].

**Theorem 1.** Let $(S, Z)$ be a safety problem with $S$ given in (4). Let $\hat{S}$ and $\hat{Z}$ be given according to (9) and (10), respectively. Suppose that $Q$ is a feedback refinement relation from $S$ to $\hat{S}$. If $\hat{K}$ solves $(\hat{S}, \hat{Z})$ then $K := \hat{K} \circ Q$ is a controller for $S$ which solves $(S, Z)$.

Suppose that $K_{\text{imp}}$ is an implementation of $\hat{S}$ and $\hat{K}$, then we obtain an implementation of $S$ and $K$ by

$$
K_{\text{imp}}(u, x) := \hat{K}_{\text{imp}}(u, P(x)) \quad (11)
$$

where $P : X \to \hat{X}$ picks for every $x$ a related cell $\hat{x}$, i.e., $(x, P(x)) \in Q$ for all $x \in X$. Again, $K_{\text{imp}}$ represents a refinement of the implementation $\hat{K}_{\text{imp}}$ derived for the symbolic model $\hat{S}$ and controller $\hat{K}$.

Let $(S, Z, c, \lambda)$ be a valued safety problem with $S$ given in (11). Consider the valued symbolic problem $(\hat{S}, \hat{Z}, \hat{c}, \hat{\lambda})$ with $\hat{S}$ and $\hat{Z}$ given according to (9) and (10), respectively. Suppose that $Q$ is a feedback refinement relation from $S$ to $\hat{S}$ and

$$
c(u, x, u') \leq \hat{c}(u, \hat{x}, u') \quad (12)
$$

holds for all $(x, \hat{x}) \in Q$ and $u, u' \in \hat{U}$. For $\lambda = 1$ in (19) and for $\lambda \in [0, 1]$ in (13), it is shown that the worst-case costs associated with the controller $K_{\text{imp}}$ derived in (11) from an implementation $\hat{K}_{\text{imp}}$ of $\hat{S}$ and $\hat{K}$ (for any qualified function $P$) is upper bounded by the worst-case costs associated with $K_{\text{imp}}$, i.e., the following inequality holds:

$$
J(K_{\text{imp}} \otimes S, c, \lambda) \leq J(\hat{K}_{\text{imp}} \otimes \hat{S}, \hat{c}, \lambda) \quad (13)
$$
In this work, we use SCOTS to compute symbolic models $\hat{S}$ of systems $S$ that represent the $\tau$-sampled behavior of continuous-time control systems (3). The feedback refinement relation $Q$ is given by the set-membership relation, i.e., $(x, \hat{x}) \in Q$ iff $x \in \hat{x}$.

4 Quantitative Safety Synthesis

In this section, we formulate quantitative games whose solutions lead to deterministic implementations of safety controllers that minimize certain cost functions. Specifically, we introduce mean-payoff games (MPG) and $\lambda$-discounted payoff games ($\lambda$-DPG) played on the subarena induced by the winning region of the safety game. In an MPG the goal of the controller is to minimize the limit-average costs accumulated along an infinite play

$$\limsup_{T \to \infty} \frac{1}{1 + T} \sum_{i=0}^{T} \gamma(v_i, v_{i+1}),$$

(14)

and in a $\lambda$-DPG the goal is to reduce the discounted costs

$$(1 - \lambda) \sum_{i=0}^{\infty} \lambda^i \gamma(v_i, v_{i+1}),$$

(15)

where $\lambda \in [0, 1)$ is the discount factor, and $\gamma$ is the cost function on the game. The intuition is that for $\lambda$ close to 0 the players only focus on optimizing w.r.t. the near future (as $\lambda^0 = 1$ and $\lambda^k \approx 0$ for $k \gg 1$), while for $\lambda$ close to 1 they focus more on optimizing the cost in the long run.

In the rest of the section we first present these games more formally, and recall some well known results. Then we show in detail how to construct the games for a given symbolic system. Moreover, we show how to derive an implementation of a safety controller from the solution of the 2-player games.

4.1 Mean-payoff games and $\lambda$-discounted payoff games

Both an MPG and a $\lambda$-DPG are played on an arena, which is a weighted directed bipartite graph consisting of the nodes $V = V_{\text{min}} \cup V_{\text{max}}$, the edges $E \subseteq (V_{\text{min}} \times V_{\text{max}} \cup V_{\text{max}} \times V_{\text{min}})$ with $E_{\text{min}} := E \cap V_{\text{min}} \times V_{\text{max}}$ and $E_{\text{max}} := E \cap V_{\text{max}} \times V_{\text{min}}$, and $\gamma : E \to \mathbb{Q}$. The nodes $V_{\text{min}}, V_{\text{max}}$ belong to the two players $P_{\text{min}}$ and $P_{\text{max}}$, respectively, and each edge $(u, v) \in E$ is assigned a rational cost $\gamma(u, v) \in \mathbb{Q}$ that $P_{\text{min}}$ has to pay to $P_{\text{max}}$. In a play, a pebble is placed on a starting node $v_0$. At each step, the player owning the current node chooses a successor of the node, and moves the pebble to it. A play is an infinite sequence $\{v_i\}_{n}^\infty$ of nodes visited by the moves. The goal of $P_{\text{min}}$ is to minimize the (maximal) value of (14) in an MPG and the value of (15) in a $\lambda$-DPG, respectively, while $P_{\text{max}}$ has the opposite goal.

A memoryless strategy for player $P_{\text{min}}$ is simply a function $\sigma : V_{\text{min}} \to V_{\text{max}}$ such that $\sigma(v)$ is a successor of $v$ in the arena. Analogously, memoryless strategies
for player $P_{\text{max}}$ are defined. It is known that for both MPG and $\lambda$-DPG memoryless strategies suffice to play optimally, i.e., there exist memoryless strategies $\sigma_{\text{min}}, \sigma_{\text{max}}$ and a valuation $v: V \to \mathbb{R}$ s.t. when $P_{\text{min}}$ uses $\sigma_{\text{min}}$ to determine to where to move the pebble – no matter how $P_{\text{max}}$ chooses to move – the resulting average cost for $P_{\text{min}}$ will be at most $v(\nu)$ for $\nu$ the node in which the pebble has been placed initially, and symmetrically for $P_{\text{max}}$ using $\sigma_{\text{max}}$. It is also known that for $\lambda \to 1^-$ the optimal game values in the $\lambda$-DPG (i.e., the costs of the plays corresponding to the optimal strategies) converge to the optimal values of the MPG [6]. In fact, there is some $\lambda_0 \in [0, 1) \cap \mathbb{Q}$ depending only on the given arena such that optimal strategies in the $\lambda$-DPG w.r.t. any $\lambda \geq \lambda_0$ and optimal strategies in the MPG coincide [1].

### 4.2 From valuated safety problems to games

Let $(\hat{S}, \hat{Z}, \hat{c}, \lambda)$ be a valuated safety problem with $\hat{S} = (\hat{X}, \hat{X}_0, \hat{U}, \hat{F})$. As the symbolic model is finite, we can use the well-known fixed point algorithms [3, 21] implemented in SCOTS, to solve the abstract safety problem and obtain a safety controller $\hat{K}: \hat{X} \to \hat{U}$ as a solution of $(\hat{S}, \hat{Z})$.

In order to obtain an implementation of $\hat{K}$ that optimizes the value for the quantitative problem, we create an arena (the same for all $\lambda$-DPG including MPG) by associating the controller with $P_{\text{min}}$, who chooses the input for a given state, and the environment with $P_{\text{max}}$, who chooses the successor in accordance with the transition relation, where the cost of an arena edge is given by the cost function we are using. As in the next section we consider cost functions that also depend on the last input issued by the controller to the system, we extend the state space with the last used input.

With the domain of the controller given by $\mathcal{D}(\hat{K})$ the nodes of the arena are

$$V_{\text{min}} := \mathcal{D}(\hat{K}) \times \hat{U} \subseteq \hat{X} \times \hat{U}, \quad V_{\text{max}} := \hat{K} \subseteq \hat{X} \times \hat{U}$$

and edges

$$E_{\text{min}} := \{(\langle x, u \rangle, \langle x, u' \rangle) \in V_{\text{min}} \times V_{\text{max}} | u' \in \hat{K}(x)\}$$

$$E_{\text{max}} := \{(\langle x, u \rangle, \langle x', u \rangle) \in V_{\text{max}} \times V_{\text{min}} | x' \in \hat{F}(x, u)\}$$

where the edges in $E_{\text{min}}$ and $E_{\text{max}}$ are assigned the costs

$$\gamma_{\text{min}}(\langle x, u \rangle, \langle x, u' \rangle) := \hat{c}(u, x, u') \quad \text{and} \quad \gamma_{\text{max}}(\langle x, u \rangle, \langle x', u \rangle) := 0,$$

respectively.

To solve a $\lambda$-DPG with $\lambda \in [0, 1)$, we use fixed point iteration on the fixed point equations derived for both players (see e.g. [6]). It follows from Banach’s fixed-point theorem that this converges to the least and only fixed point, and

This in fact means that we consider strategies with bounded memory.
additionally that it converges quickly unless $\lambda$ is close to 1. For the case $\lambda = 1$ we solve the resulting MPG using the tool presented in [14].

The output in each case is an optimal strategy $\sigma_{\text{min}} : V_{\text{min}} \rightarrow V_{\text{max}}$ for $P_{\text{min}}$. An implementation of $\hat{S}$ and $\hat{K}$ from $\sigma_{\text{min}}$ is given by

$$\hat{K}_{\text{imp}}(x, u) := \pi_{\hat{U}}(\sigma_{\text{min}}(x, u))$$

where $\pi_{\hat{U}}$ is the projection of a pair $(x, u)$ onto $\hat{U}$, i.e., $\pi_{\hat{U}}(x, u) = u$.

5 Case Studies

In order to analyse the influence of the discount factor $\lambda$ on the synthesized controller implementation, we study both a HVAC system and the classical cart-pole system in Section 5.1 and 5.2, respectively. For both case studies, we will consider the following cost functions:

- $c_{\text{IS}}(u, x, u') = \begin{cases} 0 & \text{if } u = u' \\ 1 & \text{if } u \neq u' \end{cases}$ (input switches) (IS)
- $c_{\text{DR}}(u, x, u') = \|\pi_r(x) - r\|^2$ (deviation from reference) (DR)
- $c_{\text{EC}}(u, x, u') = \|u - u_0\|^2$ (energy consumption) (EC)
- $c_{\text{ID}}(u, x, u') = \|u - u'\|^2$ (input deviation) (ID)

These are instantiated with a reference point $r \in \mathbb{R}^m$ (e.g. the optimal temperature), a projection $\pi_r : X \rightarrow \mathbb{R}^m$ onto the coordinates of the reference point and the input $u_0$ consuming minimal energy (e.g. where no actuators are active).

The function $c_{\text{IS}}$ is used to minimize the average amount of input switches and $c_{\text{DR}}$ is used to minimize the average deviation from a reference point. These two criteria were already introduced in [19]. The function $c_{\text{EC}}$ assumes that the distance of an input from an energy-minimal input correlates with the amount of energy consumed by using this input. Then we can use this function to minimize the average amount of energy consumed. The function $c_{\text{ID}}$ minimizes the change in successive inputs, which could e.g. lead to less jerky trajectories. We will not analyze the latter two costs seperately, but use them as intermediate functions to derive the following combined cost function, which is used in [13]:

$$c_{\text{CC}}(u, x, u') = \frac{1}{3} \left( \frac{c_{\text{DR}}(u, x, u')}{\text{max}_{\text{DR}}} + \frac{c_{\text{EC}}(u, x, u')}{\text{max}_{\text{EC}}} + \frac{c_{\text{ID}}(u, x, u')}{\text{max}_{\text{ID}}} \right)$$

(CC)

Here, the normalizing factors $\text{max}_{\text{DR}}$, $\text{max}_{\text{EC}}$ and $\text{max}_{\text{ID}}$ are the maximal values of the respective cost functions within the domain of the safety controller $\hat{K}$.

We derive the cost function $\hat{c}(u, \hat{x}, u')$ required by [12] by taking the maximum of $c(u, x, u')$ for all $(x, \hat{x}) \in Q$. Additionally, as we require rational costs for solving MPGs, the costs are rounded with a precision of 6 decimal digits.
5.1 Heat, ventilation and air conditioning

In our first example, we synthesize an implementation of a safety controller for a heat, ventilation and air conditioning (HVAC) system. We follow closely the setup described in [5,10], which considers a rooftop unit that is used to regulate the temperature and to circulate the air in different zones in a building to keep the air at a comfortable level. The HVAC system consists of a packaged direct expansion cooling rooftop unit (RTU) that conditions one zone in a single story building, which is equipped with a two-stage compressor, a multi-speed fan and modulating economiser dampers. It is assumed that the economiser dampers remain in constant position and are not available for control. A control unit is used to regulate the temperature and the humidity of the air within the regulated zone within a desired comfort interval, despite the presence of disturbances. We refer the interested readers to [5,10] for a more detailed description of the HVAC.

A linear dynamical system with four states that approximates the local system behavior at a pre-specified nominal behavior given by the zone set-points with fixed heating loads, moister loads and RTU actions serves as basis of the design scheme. The nominal dynamics in (3) is described by \( f(x, u) = Ax + Bu \), where the matrices are determined from data during nominal operation as follows

\[
A = 10^{-4} \cdot \begin{pmatrix}
-28 & -5.6 & 0 & 0 \\
0 & -8.3 & 0 & 0 \\
0 & 0 & -17 & 1 \\
0 & 0 & 0 & -2.8
\end{pmatrix}, \quad B = 10^{-4} \cdot \begin{pmatrix}
-0.8 & -1.7 \\
0 & 5.8 \\
-1.7 & 0.08 \\
0 & 2.3
\end{pmatrix}.
\]

The inputs of the system denoted by \( \nu_1(t) \) and \( \nu_2(t) \) represent the fan angular velocity and compressor angular velocity, respectively, and are restricted for all times \( t \in \mathbb{R}_{\geq 0} \) to the stage values

\[
\nu(t) \in U := \{-25, 0, 25\} \times \{-50, 0, 50\}.
\]

In order to account for input uncertainties (which according to [10] are also used to account for model uncertainties) we use a perturbation bound of \( w = B \cdot (10, 10)^\top \) in (3). Notably our disturbance model is rather simple yet powerful and unlike to [5,10], we do not assume that the disturbance signal is constant during sampling times.

The first and the third elements of the state vector represent the zone temperature in degree Celsius and the zone relative humidity in \%, respectively. The values are restricted to lie within \([-1, 1]\) and \([-5, 5]\), respectively. As a result, we obtain as safety specification

\[
Z := [-1, 1] \times \mathbb{R} \times [-5, 5] \times \mathbb{R}.
\]

For the HVAC system, we can interpret the quantitative specifications as the following:

1. (IS): Find a lazy controller that minimizes the amount of input switches. In this system, this reduces the wear on the compressor and fan.
2. **(DR)**: Find a controller with minimal deviation from the optimal comfort level. We set the reference point \( r = (0, 0) \) and \( \pi_r(x) = (x_1, x_3) \), i.e. temperature and relative humidity should be kept close to the normalized optimal value, which corresponds to 21°C and 50% humidity.

3. **(CC)**: Simultaneously minimize comfort, energy consumption and relative change in inputs. The minimal energy input \( u_0 = (-25, -50) \) corresponds to having the fan and compressor turned off.

   We will construct optimized controllers for each of the cost functions, with different values of \( \lambda \), and evaluate them by comparing the limit-average cost in the long-term when simulating the system.

   We introduce a disturbance signal \( \omega \) during simulation by instantiating equation (3) by \( \dot{\xi}(t) = f(\xi(t), u) + \omega(t) \), and consider the following disturbance signals:

   \[
   \omega_{\text{sin}}(t) := (10 \sin \left( \frac{t}{2\pi} \right), -10 \sin \left( \frac{t}{2\pi} \right))^T ;
   \omega_{\text{con}}(t) := (10, -10)^T .
   \]

   We use \( \eta = (0.2, 1, 0.4, 10) \) as the discretization parameter for constructing the symbolic model and \( \tau = 100 \text{ sec} \) as the sampling time.

### 5.2 Cart-pole system

In this example, we synthesize an implementation of a safety controller that ensures that the pole, which is attached to a cart, stays within a neighborhood of the upright position. The cart-pole system is illustrated in Figure 2.

\[
\begin{align*}
\dot{x}_1 &= \omega_{\text{sin}}(t) := (10 \sin \left( \frac{t}{2\pi} \right), -10 \sin \left( \frac{t}{2\pi} \right))^T ; \\
\dot{x}_2 &= f_1(x, u) = x_2, \\
\dot{x}_3 &= f_3(x, u) = x_4, \\
\dot{x}_4 &= f_4(x, u) = u,
\end{align*}
\]

where \( \alpha = 1 \) and \( \beta = 0.0125 \). For this example, we assume that there are no disturbances and set \( \omega = 0 \). While in [17], a reachability problem has been
solved to regulate the cart-pole from the downward facing position to the upright position, we focus on the safety problem to force the cart-pole to stay in the upright position, i.e., the first state $x_1$ is constrained to $\left[\frac{1}{2}\pi, \frac{3}{2}\pi\right]$. As in [17, 20], we constrain the third coordinate $x_3$ to $[-2.4, 2.4]$. For the velocity coordinates $x_2$ and $x_4$ we use the constraints $[-1, 1]$ and $[-1.4, 1.4]$, respectively. We enforce an input bound of $U = [-5, 5]$. We use SCOTS to synthesize a safety controller $K$ with the discretization parameter $\eta = (0.05, 0.1, 0.1, 0.1)$ and the sampling time $\tau = 0.35$ sec.

Here, we can interpret the quantitative specifications as the following:

1. **(IS):** Find a lazy controller that minimizes the amount of input switches. In this system, this avoids frequent changes in acceleration, which causes stress in the system.
2. **(DR):** Find a controller with minimal deviation from the reference point $r = (\pi, 0)$ with $\pi_r(x) = (x_1, x_3)$. This reference point corresponds to the pole being in upright position and the cart being in the center.
3. **(CC):** Simultaneously minimize deviation from the reference point, energy consumption and relative change in acceleration. The minimal energy input $u_0 = 0$ corresponds to no acceleration.

As for the HVAC system, we construct optimized controllers for each cost function with different values of $\lambda$ and compare their long-term performance w.r.t. these cost functions in the simulation.

### 6 Experimental Evaluation

For each of the two case studies, we constructed the symbolic system $\hat{S}$ and solved the safety game to obtain a safety controller $\hat{K}$. Then for each of the three cost functions $c_{IS}$, $c_{DR}$ and $c_{CC}$ and for different values of $\lambda$, we solved the resulting $\lambda$-DPG and translated the optimal strategies into an implementation $K_{imp}$. We choose the values of $\lambda \in [0, 1]$ to include the two extrema: $\lambda = 0$, which results in a controller greedily optimizing one step, and $\lambda = 1$, which results in a controller giving an optimal solution for the limit-average cost on the symbolic system. Additionally, we chose $\lambda = 1/2$, which is the value chosen in [13], and several values spaced more closely towards the boundaries.

Then we simulate each controller on its respective system, using two different disturbance functions for the HVAC system, and measure the the limit-average cost for the cost function for which the controller is optimized. Assessing the performance of the obtained controller w.r.t. the limit average has two reasons:

- As the controllers are assumed to run indefinitely on the system, the limit-average cost is usually the value that is actually the most relevant.
- This allows us to compare controllers obtained for different values of $\lambda$, i.e. whether it is preferable to chose $\lambda$ close to 0 so that $P_{min}$ only optimizes w.r.t. the near future or $\lambda$ close or equal to 1 so that the far future becomes more and more important for $P_{min}$.
In the simulations, the systems display periodic behaviour, and we run the simulation sufficiently long enough until the limit-average cost stabilizes.

The obtained values are summarized in Table 2. The sizes of the symbolic models and times needed to construct them, to construct the arena, and to solve the respective games are given in Table 1.

Table 1. Size of the symbolic model for each system and times in seconds to construct the safety controller $\hat{K}$, the game arena $G$, and the maximum times over all cost functions $c$ to solve the $\lambda$-DPG for any $\lambda \leq \frac{15}{16}$ and to solve the MPG.

| System     | $|\hat{X}|$ | $|U|$ | $|\hat{F}|$ | $\hat{K}$ | $G$ | $\lambda$-DPG | MPG |
|------------|-----------|------|---------|----------|----|---------------|-----|
| HVAC       | $1.4 \cdot 10^3$ | 12   | $3.8 \cdot 10^6$ | 16.78    | 55.37 | 8.49         | 477 |
| Cart-pole  | $2.3 \cdot 10^6$ | 101  | $3.1 \cdot 10^9$ | 3558     | 1306 | 156          | 8846|

Table 2. Values of the limit-average cost for different cost functions and controllers on the two systems. For each system and cost function $c$, we synthesize controllers for each given value of $\lambda$. We then simulate the controller, possibly with different disturbance signals $\omega$, and measure the limit-average cost w.r.t. the cost function $c$. The entry for each value of $\lambda$ lists the result of this limit-average cost with the respective controller. The entry for $v$ lists the upper guaranteed bound on the limit-average cost by the controller from the MPG with $\lambda = 1$. The best values in each row are marked in bold.

| System     | $c$   | $\omega$ | $0$ | $\frac{1}{16}$ | $\frac{1}{8}$ | $\frac{1}{4}$ | $\frac{1}{2}$ | $\frac{3}{4}$ | $\frac{7}{8}$ | $\frac{15}{16}$ | $1$ | $v$ |
|------------|-------|----------|-----|----------------|---------------|---------------|---------------|---------------|---------------|----------------|-----|-----|
| HVAC       | $c_{IS}$ | $\omega_{sin}$ | 0.082 | $0.051$ | $0.051$ | $0.051$ | $0.051$ | $0.051$ | $0.051$ | $0.051$ | 0.750 |
|           |       | $\omega_{con}$ | $0.278$ | $0.167$ | $0.167$ | $0.167$ | $0.167$ | $0.167$ | $0.167$ | $0.175$ | 5.410 |
| Cart-pole  | $c_{IS}$ | $\omega_{sin}$ | 2.631 | $0.069$ | $0.069$ | $0.069$ | $0.069$ | 0.079 | 0.165 | 1.919 |
|           |       | $\omega_{con}$ | $12.784$ | $0.109$ | $0.109$ | $0.109$ | $0.109$ | $0.126$ | 0.137 | 2.757 |

| System     | $c_{CC}$ | $\omega_{sin}$ | 0.147 | 0.137 | 0.136 | 0.132 | 0.130 | 0.126 | 0.119 | 0.123 | 0.148 | 0.796 |
| Cart-pole  | $c_{CC}$ | $\omega_{con}$ | 0.346 | 0.346 | 0.346 | 0.346 | 0.336 | 0.261 | $0.170$ | $0.170$ | $0.213$ | 1.246 |

In our experiments, the controller with an optimal choice of $\lambda$ is sometimes up to 50 times better than a controller for a different $\lambda$. Also, there is no single optimal choice for $\lambda$: depending on the system (resp. the symbolic model) and the cost function, it may be necessary to choose different values. In particular, even though the controller for $\lambda = 1$ gives an optimal solution for the symbolic
system, its actual performance on the continuous-time system is almost never optimal when compared to the implementations obtained for smaller values of $\lambda$.

Note that our approach can handle systems with up to several million states and a billion transitions (see Table 1), and that most of the computation time is spent on constructing the safety controller and the unweighted arena (i.e. without edge costs). As this basic arena is independent of the choice of $c$ and $\lambda$, it can be reused to solve different $\lambda$-DPGs. Solving such a game is fast unless $\lambda$ tends to 1, and therefore our tool chain allows to easily construct and evaluate controllers for several different values of $\lambda$ and choices of the cost function $c$.

Our explanation for the subpar performance of the controller obtained for $\lambda = 1$ is that this reflects the conservative over-approximation used in the construction of the symbolic model. In the game on the symbolic system, the adversary player ($P_{\text{max}}$) chooses successors for sequences of inputs from the symbolic states, which may lead to transition sequences that can never occur in the concrete system. This overestimation of reachable states increases as the length of the sequence increases. In the case of an MPG, the controller ($P_{\text{min}}$) optimizes its inputs to guard against the worst-case for arbitrarily long sequences. It may choose any strategy that achieves the optimal value $v$, even a very conservative one. As seen in Table 2, the optimal value of the MPG is usually much higher than the actually achievable minimal costs. To support our hypothesis, we tried to simulate the worst-case behaviour by extracting the optimal strategy $\sigma_{\text{max}}$ of $P_{\text{max}}$ from the MPG for the HVAC system and testing if we could choose a disturbance $\omega(t) \in ([-10, 10] \times [-10, 10]) \cap (\mathbb{Z} \times \mathbb{Z})$ in each time step leading to a successor in accordance with $\sigma_{\text{max}}$. However, this was impossible most of the time, and we could not enforce the worst-case costs. While this approach does not cover all possible disturbance signals, it still gives an indication that many transitions of the symbolic system can not occur in the concrete system.

To reduce this gap, one might choose a finer approximation, however this is often impossible due to the increasing size of the symbolic system. Using discounted payoff functions comes at no additional cost and leads to better performing controllers. This is due to their focus on optimization towards the near future, which offsets the error from the overapproximation.

7 Summary

In this paper we studied different quantitative implementation strategies of safety controllers for perturbed, nonlinear, control systems. We use normalized $\lambda$-discounted costs in order to define the costs accumulated along a run of the system. The normalization allows us to also cover the limit-average costs for $\lambda \to 1^-$ thereby also obtaining a unified presentation of previous results. We present two case studies and conduct a systematic analysis to illustrate the effects of different choices of cost functions and discount parameters. We show that carefully choosing $\lambda$ allows us to reduce the limit-average cost associated with the refined controller quite drastically when compared to the fixed values of $\lambda = \frac{1}{2}$ and $\lambda \to 1^-$ found in [13] and [19], respectively. Our explanation for
this is that by carefully choosing $\lambda$ the safe, yet pessimistic over-approximation used for the construction of the symbolic model can be offset at least to some extent. With our existing tool-chain it is effectively possible to sample for different choices of $\lambda$ in a reasonable amount of time for symbolic models consisting of up to several millions of states.

References

1. Andersson, D., Miltersen, P.B.: The Complexity of Solving Stochastic Games on Graphs, pp. 112–121. Springer (2009)
2. Aubin, J.P.: Viability theory. Birhäuser (1991)
3. Bertsekas, D., Rhodes, I.B.: On the minimax reachability of target sets and target tubes. Automatica 7, 233–247 (1971)
4. Blanchini, F., Miani, S.: Set-theoretic methods in control. Springer (2008)
5. Brocchini, C., Falsone, A., Manganini, G., Holub, O., Prandini, M.: A chance-constrained approach to the quantized control of a heat ventilation and air conditioning system with prioritized constraints. In: Proc. of the 22nd Int. Symp. on Mathematical Theory of Networks and Systems (2016)
6. Filar, J., Vrieze, K.: Competitive Markov Decision Processes. Springer (1996)
7. Filippov, A.F.: Differential equations with discontinuous righthand sides, Mathematics and its Applications (Soviet Series), vol. 18. Kluwer Academic Publishers Group, Dordrecht (1988), translated from the Russian
8. Girard, A.: Controller synthesis for safety and reachability via approximate bisimulation. Automatica 48(5), 947–953 (2012)
9. Gol, E.A., Lazar, M., Belta, C.: Temporal logic model predictive control. Automatica 56, 78–85 (2015)
10. Holub, O., Zamani, M., Abate, A.: Efficient HVAC controls: A symbolic approach. In: Proc. of the 15th European Control Conference (2016)
11. Maidens, J.N., Kaynama, S., Mitchell, I.M., Oishi, M.M., Dumont, G.A.: Lagrangian methods for approximating the viability kernel in high-dimensional systems. Automatica 49(7), 2017–2029 (2013)
12. Mazo, M., Tabuada, P.: Symbolic approximate time-optimal control. Systems & Control Letters pp. 256–263 (2011)
13. Meyer, P., Girard, A., Witrant, E.: Safety control with performance guarantees of cooperative systems using compositional abstractions. IFAC-PapersOnLine 48(27), 317–322 (2015)
14. Meyer, P.J., Luttenberger, M.: Solving mean-payoff games on the GPU. In: Proc. of the 14th Int. Symp. on Automated Technology for Verification and Analysis (2016)
15. Reissig, G., Rungger, M.: Abstraction-based solution of optimal stopping problems under uncertainty. In: Proc. of the 52nd IEEE Conf. on Decision and Control. pp. 3190–3196 (2013)
16. Reißig, G., Weber, A., Rungger, M.: Feedback refinement relations for the synthesis of symbolic controllers. IEEE Transactions on Automatic Control 62 (2016)
17. Reißig, G.: Abstraction based solution of complex attainability problems for decomposable continuous plants. In: Proc. of the 49th IEEE Conf. on Decision and Control. pp. 5911–5917. IEEE (2010)
18. Rungger, M., Zamani, M.: SCOTS: A tool for the synthesis of symbolic controllers. In: Proc. of the 19th Int. Conf. on Hybrid Systems: Computation and Control. pp. 99–104. ACM (2016)
19. Rungger, M., Reissig, G., Zamani, M.: Symbolic synthesis with average performance guarantees. In: Proc. of the 55th IEEE Conf. on Decision and Control. pp. 7404–7410. IEEE (2016)

20. Rungger, M., Stursberg, O.: On-the-fly model abstraction for controller synthesis. In: American Control Conference (ACC), 2012. pp. 2645–2650. IEEE (2012)

21. Tabuada, P.: Verification and Control of Hybrid Systems: A symbolic approach. Springer (2009)

22. Tazaki, Y., Imura, J.I.: Discrete abstractions of nonlinear systems based on error propagation analysis. IEEE Transactions on Automatic Control 57(3), 550–564 (2012)

23. Wolff, E.M., Topcu, U., Murray, R.M.: Optimal control with weighted average costs and temporal logic specifications. In: Robotics: Science and Systems (2012)

24. Wolff, E.M., Murray, R.M.: Optimal control of nonlinear systems with temporal logic specifications. In: Robotics Research, pp. 21–37. Springer (2016)

25. Zamani, M., Pola, G., Jr., M.M., Tabuada, P.: Symbolic models for nonlinear control systems without stability assumptions. IEEE Transactions on Automatic Control 57(7), 1804–1809 (July 2012)