Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 54-5 (2021) 25–30

7th IFAC conference on Analysis & Design of Hybrid Systems
Repeatability Evaluated
2021

Assume/Guarantee Contracts for Dynamical Systems: Theory and Computational Tools [⋆]

Miel Sharf [*] **Bart Besselink** [**] **Adam Molin** [***]
Qiming Zhao [****] **Karl Henrik Johansson** [*]

[*] *Division of Decision and Control Systems, KTH Royal Institute of Technology, and Digital Futures. 10044 Stockholm, Sweden (e-mail:* {sharf,kallej}@kth.se*).*
[**] *Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen. 9700 AK Groningen, The Netherlands (e-mail:* b.besselink@rug.nl*).*
[***] *DENSO Automotive Deutschland GmbH. Freisinger Str. 21-23, 85386 Eching, Germany (e-mail:* a.molin@eu.denso.com*).*
[****] *DENSO International America, Inc. 24777 Denso Dr., Southfield, MI 48033, USA (e-mail:* qiming.zhao@na.denso.com*).*

Abstract: Modern engineering systems include many components of different types and functions. Verifying that these systems satisfy given specifications can be an arduous task, as most formal verification methods are limited to systems of moderate size. Recently, contract theory has been proposed as a modular framework for defining specifications. In this paper, we present a contract theory for discrete-time dynamical control systems relying on assume/guarantee contracts, which prescribe assumptions on the input of the system and guarantees on the output. We then focus on contracts defined by linear constraints, and develop efficient computational tools for verification of satisfaction and refinement based on linear programming. We exemplify these tools in a simulation example, proving a certain safety specification for a two-vehicle autonomous driving setting.

1. INTRODUCTION

Engineering systems are often comprised of many components having different types and functions, including sensing, control, and actuation. Moreover, systems are subject to many specifications, such as safety and performance. The former are captured using notions of set-invariance, and the latter are usually defined in terms of dissipativity. However, modern systems such as intelligent transportation systems and smart manufacturing systems have more complex specifications which cannot be captured by these frameworks, e.g. behaviour, tracking, and temporal logic specifications. Formal methods in control have been developed to address this issue (Belta et al. (2017); Tabuada (2009); Wongpiromsarn et al. (2010)). Unfortunately, the sheer size of modern engineering systems implies that formal verification methods are ineffective, as the need to discretize the state-space results in a curse of dimensionality. Such verification processes can also be extremely wasteful, as even a minuscule change to the dynamical system (e.g., a small change in one of its components) requires starting the verification processes from scratch.

In this paper we present a verification approach relying on contract theory. Contract theory was first developed in the field of software engineering as a modular approach to system design (Meyer (1992)), and it has proved useful for design of cyber-physical systems (Nuzzo et al. (2014, 2015); Phan-Minh et al. (2019)). Contracts prescribe assumptions on the environments a software component can act in, and guarantees on its behaviour in those environments (Benveniste et al. (2018)). We focus on assume/guarantee contracts, which put assumptions on the input to a software component and prescribe guarantees on its output. Computational tools for verifying that a given component satisfies a given contract are needed in order to apply the theory. In recent years, some attempts were made to define a contract theory for dynamical (control) systems. An assume/guarantee framework for continuous-time dynamical systems based on the notion of simulation was considered in Besselink et al. (2019)Assume/guarantee contracts have also been considered in Saoud et al. (2018, 2019), in which assumptions are made on the input signals and guarantees are on the state and output signals.

In this paper, we present a framework for assume/guarantee contracts that prescribe assumptions on the inputs and guarantees on the output, extending the framework of Saoud et al. (2018, 2019). First, we allow the requirement on the output to depend on the input, which is natural for sensor systems and tasks like tracking. Second, we do not limit the internal structure of the component, for instance, we do not specify its state. In particular, the analysis of a system can be conducted at a preliminary design stage, before we even know whether, for example, the controller will be static or not. We also define satisfaction, refine-

[⋆] This work was supported by DENSO Automotive Deutschland GmbH. A full version of the paper is available on `https://arxiv.org/abs/2012.12657`, see Sharf et al. (2020).

10.1016/j.ifacol.2021.08.469

ment, and cascaded composition for contracts. We then focus on contracts in which the assumptions and guarantees are prescribed using linear inequalities, and present efficient computational tools for verifying satisfaction and refinement based on linear programming (LP), which can be understood as a version of the k-induction method for model checking (Donaldson et al. (2011)). This is the main contribution of this paper.

The rest of the paper is organized as follows. Section 2 presents assume/guarantee contracts as well as the notions of satisfaction, refinement, and cascaded composition, and gives examples. Section 3 develops computational methods for verification of satisfaction and refinement. Section 4 provides a simulation example.

Notation We denote the collection of natural numbers by $\mathbb{N} = \{0, 1, 2, \ldots\}$. For two sets X, Y, we denote their Cartesian product by $X \times Y$. For a positive integer n, we denote the collection of all signals $\mathbb{N} \to \mathbb{R}^n$ by $\mathcal{S}^n$. For vectors $v, u \in \mathbb{R}^n$, we understand $v \le u$ as an entry-wise inequality. Given a state-space system (A, B, C, D), we denote the observability matrix $\mathcal{O}_m = [C^\top, (CA)^\top, \ldots, (CA^m)^\top]^\top$ and define the observability index ν as the minimal integer such that $\text{rank } \mathcal{O}_\nu = \text{rank } \mathcal{O}_{\nu+1}$.

2. ASSUME/GUARANTEE CONTRACTS

In this section, we define the class of systems for which we introduce an abstract framework of assume/guarantee contracts, as well as supporting notions such as satisfaction, refinement, and cascaded composition. This is an adaptation of the framework presented in Benveniste et al. (2018). In section 3, we will specialize to a class of contracts for which efficient computational tools can be introduced.

Definition 1. A system Σ is a tuple $(\mathcal{X}_0, A, B, C, D)$ with input $d \in \mathcal{S}^{n_d}$, output $y \in \mathcal{S}^{n_y}$, and state $x \in \mathcal{S}^{n_x}$. The set $\mathcal{X}_0 \subseteq \mathbb{R}^{n_x}$ is a set of initial conditions, and A, B, C, D are matrices of appropriate sizes such that the state evolution and output are given by the following equations:

$$\begin{cases} x(0) & \in \mathcal{X}_0 \\ x(k+1) & = Ax(k) + Bd(k),\ \forall k \in \mathbb{N} \\ y(k) & = Cx(k) + Dd(k),\ \forall k \in \mathbb{N}. \end{cases} \tag{1}$$

For signals $d \in \mathcal{S}^{n_d}$ and $y \in \mathcal{S}^{n_y}$, we write $y \in \Sigma(d)$ if there exists a signal $x \in \mathcal{S}^{n_x}$ such that $d(\cdot), x(\cdot), y(\cdot)$ satisfy (1).

It is essential to include the set of allowable initial states $\mathcal{X}_0$ in the definition of a system in order to discuss various specifications. For example, asking whether the output of a system lies in a given safe set is meaningless if we make no assumptions on the initial state, no matter the value of the input $d(\cdot)$.

Remark 1. Definition 1 can be generalized by allowing $\mathcal{X}_0$ to be dependent of $d(0)$. This is reasonable for systems trying to track $d(\cdot)$, assuming their initial tracking error is not too large. This is also reasonable for systems trying to avoid an obstacle whose position is defined by $d(\cdot)$, assuming the system does not start on top of the obstacle.

We wish to consider specifications on the behaviour of dynamical systems. A dynamical system can be thought of as a map from input signals $d(\cdot) \in \mathcal{S}^{n_d}$ to output signals $y(\cdot) \in \mathcal{S}^{n_y}$. As such, we can adopt the formulation of assume/guarantee contracts by merely making assumptions on the input variable $d(\cdot)$ and demanding guarantees on the output variable $y(\cdot)$ given the input $d(\cdot)$.

Definition 2. An assume/guarantee contract is a pair $(\mathcal{D}, \Omega)$ where $\mathcal{D} \subseteq \mathcal{S}^{n_d}$ are the assumptions and $\Omega \subseteq \mathcal{S}^{n_d} \times \mathcal{S}^{n_y}$ are the guarantees.

In other words, we put assumptions on the input $d(\cdot)$ and demand guarantees on the input-output pair $(d(\cdot), y(\cdot))$.

Example 1. We say that a system $d \mapsto y$ has finite ℓ_2-gain no more than β if for any $d \in \ell_2$, we have $y \in \ell_2$ and $\|y\|_{\ell_2} \le \beta \|d\|_{\ell_2}$. This property can be written as an assume/guarantee contract $\mathcal{C} = (\mathcal{D}, \Omega)$, where $\mathcal{D} = \ell_2$ and $\Omega = \{(d(\cdot), y(\cdot)) : \|y\|_{\ell_2} \le \beta \|d\|_{\ell_2}\}$.

Example 2. We say that a SISO system $d \mapsto y$ exponentially tracks constant signals with exponent $\lambda \in (0, 1)$ if for any constant input d, the output y satisfies $|y(k) - d(k)| \le \lambda |y(k-1) - d(k-1)|$ for all k. This property can be written as an assume/guarantee contract $\mathcal{C} = (\mathcal{D}, \Omega)$, where $\mathcal{D} = \{d(\cdot) : d(k+1) = d(k),\ \forall k\}$ and $\Omega = \{(d(\cdot), y(\cdot)) : |y(k+1) - d(k+1)| \le \lambda |y(k) - d(k)|,\ \forall k\}$.

Let us now define the notion of satisfaction. This notion connects systems and assume/guarantee contracts, by defining when a given system satisfies the specifications defined by a given contract.

Definition 3. We say that a system Σ satisfies $\mathcal{C} = (\mathcal{D}, \Omega)$ (or implements $\mathcal{C}$), and write $\Sigma \vDash \mathcal{C}$, if for any $d \in \mathcal{D}$ and any $y \in \Sigma(d)$, $(d, y) \in \Omega$.

2.1 Refinement and Composition

One of the greatest perks of contract theory is its modularity, as one can refine a contract on a composite system by "smaller" contracts on subsystems, which can be further refined by even "smaller" contracts on individual components. The two notions supporting this idea are refinement, defining when one contract is stricter than another, and composition, defining the coupling of multiple contracts. In this subsection, we define the notion of refinement for assume/guarantee contracts, as well as a restricted notion of contract composition for cascade systems. Computational tools for these notions will be provided in the next section. We start by defining refinement:

Definition 4. Let $\mathcal{C}_i = (\mathcal{D}_i, \Omega_i)$ be contracts for $i = 1, 2$. We say $\mathcal{C}_1$ *refines* $\mathcal{C}_2$ (and write $\mathcal{C}_1 \preccurlyeq \mathcal{C}_2$) if $\mathcal{D}_1 \supseteq \mathcal{D}_2$ and $\Omega_1 \cap (\mathcal{D}_2 \times \mathcal{S}^{n_y}) \subseteq \Omega_2 \cap (\mathcal{D}_2 \times \mathcal{S}^{n_y})$, where n_y is the dimension of the output.

Colloquially, $\mathcal{C}_1 \preccurlyeq \mathcal{C}_2$ if $\mathcal{C}_1$ assumes less than $\mathcal{C}_2$, but guarantees more given the assumptions.

Example 3. Consider two contracts used for tracking. The first, $\mathcal{C} = (\mathcal{D}, \Omega)$ defines asymptotic tracking of certain inputs, namely $d, y \in \mathcal{S}^m$, $\mathcal{D} \subseteq \mathcal{S}^m$, and

$$\Omega = \{(d(\cdot), y(\cdot)) : \lim_{k \to \infty} \|d(k) - y(k)\| = 0\}$$

The second, $\mathcal{C}' = (\mathcal{D}, \Omega')$, defines exponential convergence, i.e., we take some $\lambda \in (0, 1)$ and define:

$$\Omega' = \{(d, y) : \|d(k) - y(k)\| \le \lambda \|d(k-1) - y(k-1)\|\}.$$

By definition, we have $\mathcal{C}' \preccurlyeq \mathcal{C}$.

Refinement provides a partial ordering of assume/guarantee contracts, and it is "harder" to satisfy refined contracts:

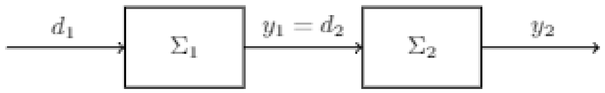


Fig. 1. Cascade of contracts

Proposition 1. Let $\mathcal{C}_i = (\mathcal{D}_i, \Omega_i)$ be assume/guarantee contracts for $i = 1,2,3$ and Σ be a system. Then, the following statements hold:

- $\mathcal{C}_1 \preccurlyeq \mathcal{C}_1$.
- If $\mathcal{C}_1 \preccurlyeq \mathcal{C}_2$ and $\mathcal{C}_2 \preccurlyeq \mathcal{C}_3$ then $\mathcal{C}_1 \preccurlyeq \mathcal{C}_3$.
- If $\mathcal{C}_1 \preccurlyeq \mathcal{C}_2$ and $\Sigma \vDash \mathcal{C}_1$, then $\Sigma \vDash \mathcal{C}_2$.

Proof. See Sharf et al. (2020).

Proposition 1 is important in contract theory, as it shows two key properties. First, if we have an original contract $\mathcal{C}$ and a refined contract $\mathcal{C}'$, any system satisfying $\mathcal{C}'$ also satisfies $\mathcal{C}$. Second, if we have an original contract $\mathcal{C}$ and a refined contract $\mathcal{C}'$, any refinement of $\mathcal{C}'$ is also a refinement of $\mathcal{C}$. These properties allow us to refine a contract on a composite system by multiple contracts on the individual subsystems, which can be further refined by a plethora of contracts on the individual components in the system. If each component satisfies its corresponding contract, then the composite system will satisfy the original contract.

We now move to cascaded composition. Consider the block diagram in Fig. 1. Define the cascaded composition of $\mathcal{C}_1$ and $\mathcal{C}_2$ such that if $\Sigma_1 \vDash \mathcal{C}_1$ and $\Sigma_2 \vDash \mathcal{C}_2$, the cascade of Σ_1 and Σ_2 satisfies the composition $\mathcal{C}_1 \otimes \mathcal{C}_2$. First, let us define the cascade of systems:

Definition 5. Let $\Sigma_i = (\mathcal{X}_i, A_i, B_i, C_i, D_i)$ be systems for $i = 1,2$. The cascade $\Sigma_1 \otimes \Sigma_2 = (\mathcal{X}_\otimes, A_\otimes, B_\otimes, C_\otimes, D_\otimes)$ has input $d_\otimes = d_1$, output $y_\otimes = y_2$, state $x_\otimes = [x_1^\top, x_2^\top]^\top$, allowable initial states $\mathcal{X}_\otimes = \mathcal{X}_1 \times \mathcal{X}_2$, and matrices $A_\otimes = \left[\begin{smallmatrix} A_1 & 0 \\ B_2C_1 & A_2 \end{smallmatrix}\right]$, $B_\otimes = \left[\begin{smallmatrix} B_1 \\ B_2D_1 \end{smallmatrix}\right]$, $C_\otimes = \left[\begin{smallmatrix} D_2C_1 & C_2 \end{smallmatrix}\right]$, $D_\otimes = D_2D_1$.

Consider two contracts $\mathcal{C}_i = (\mathcal{D}_i, \Omega_i)$ as in Fig. 1. When defining a contract that is satisfied by the composition, we at least need $d_1 \in \mathcal{D}_1$ and $(d_2, y_2) \in \Omega_2$. The latter also requires $d_2 \in \mathcal{D}_2$, while the former only implies $(d_1, y_1) \in \Omega_1$. This motivates the following definition:

Definition 6. For two contracts $\mathcal{C}_i = (\mathcal{D}_i, \Omega_i)$, the cascaded composition is $\mathcal{C}_1 \otimes \mathcal{C}_2 = (\mathcal{D}_\otimes, \Omega_\otimes)$ with input $d_\otimes = d_1$, output $y_\otimes = y_2$,

$$\mathcal{D}_\otimes = \{d_\otimes : \ d_\otimes \in \mathcal{D}_1, ((d_\otimes, y_1) \in \Omega_1 \implies y_1 \in \mathcal{D}_2)\},$$
$$\Omega_\otimes = \{(d_\otimes, y_\otimes) : \exists d_2 = y_1, (d_\otimes, y_1) \in \Omega_1, (d_2, y_\otimes) \in \Omega_2\}.$$

We now prove our main claim about contract composition:

Proposition 2. Let $\mathcal{C}_1, \mathcal{C}_2$ and Σ_1, Σ_2 be contracts and systems with inputs d_1, d_2 and outputs y_1, y_2. If $\Sigma_1 \vDash \mathcal{C}_1$ and $\Sigma_2 \vDash \mathcal{C}_2$, then $\Sigma_1 \otimes \Sigma_2 \vDash \mathcal{C}_1 \otimes \mathcal{C}_2$.

Proof. See Sharf et al. (2020)

3. COMPUTATIONAL TOOLS FOR VERIFICATION

The previous section presented abstract assume/guarantee contracts for discrete-time dynamical systems, as well the notions of satisfaction, refinement and cascaded composition. In this section, we present computational tools for verifying satisfaction and refinement, relying on mathematical induction and linear programming. We rely on linearity of both the systems and specifications. More precisely, we present computational tools for assumptions of the form $A^1 d(k+1) + A^0 d(k) \le a^0$ for all k, and guarantees of the form $G^1 \left[\begin{smallmatrix} d(k+1) \\ y(k+1) \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d(k) \\ y(k) \end{smallmatrix}\right] \le g^0$ for all k, where A^0, A^1, G^0, G^1 are matrices and a^0, g^0 are vectors of appropriate dimensions. Specifications of this form include general bounded signals, as well as outputs of dynamical systems (e.g., the input d is the output of a given first-order system). In Section 4, we use specifications of this form to model a contract where the input is assumed to be a (constrained) trajectory of a dynamical system, and the guarantee is a linear inequality defining safe behaviour.

3.1 *Verifying Satisfaction*

Consider a contract $\mathcal{C} = (\mathcal{D}, \Omega)$ where

$$\mathcal{D} = \{d(\cdot) : A^1 d(k+1) + A^0 d(k) \le a^0, \ \forall k\}, \tag{2}$$
$$\Omega = \left\{(d(\cdot), y(\cdot)) : G^1 \left[\begin{smallmatrix} d(k+1) \\ y(k+1) \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d(k) \\ y(k) \end{smallmatrix}\right] \le g^0, \ \forall k\right\} \tag{3}$$

for given $A^0, A^1, G^0, G^1, a^0, g^0$.

Example 4. Suppose $d(\cdot) \in \mathcal{S}^2$ is the position of a robot in a field, parameterized by $\mathcal{F} = \{p \in \mathbb{R}^2 : L_1 \le p_1 \le U_1, L_2 \le p_2 \le U_2\}$ for constants L_1, L_2, U_1, U_2. Assume that for all $k \in \mathbb{N}$, $d(k) \in \mathcal{F}$ and $d(k+1) = d(k) + v(k)$ for some velocity $v(k)$ bounded by $V_{\max}$, i.e. that $-[V_{\max}, V_{\max}]^\top \le d(k+1) - d(k) \le [V_{\max}, V_{\max}]^\top$. The assumptions are of the form (2) where:

$$A^0 = \left[\begin{smallmatrix} 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 \end{smallmatrix}\right]^\top, \ A^1 = \left[\begin{smallmatrix} 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{smallmatrix}\right]^\top$$

and $a^0 = \left[\begin{smallmatrix} U_1 & L_1 & U_2 & L_2 & -V_{\max} & V_{\max} & -V_{\max} & V_{\max} \end{smallmatrix}\right]^\top$.

Let us make the following assumption on (2),(3):

Definition 7. Given two matrices V^1, V^0 and a vector v^0, we say (V^1, V^0, v^0) is *extendable* if for any two vectors u_0, u_1 and $V^1 u_1 + V^0 u_0 \le v^0$, there exists some vector u_2 such that $V^1 u_2 + V^0 u_1 \le v^0$.

Assuming (A^1, A^0, a^0) (or (G^1, G^0, g^0)) is extendable is not very restrictive. It is equivalent to assuming that any signal $v(\cdot)$ adhering to the assumption, and defined for times $k = 0, \ldots, n$, can be extended to a signal defined for all times $k \in \mathbb{N}$ while satisfying the assumption.

Theorem 3. Let $\mathcal{C} = (\mathcal{D}, \Omega)$ be a contract with (2) and (3), and let $\Sigma = (\mathcal{X}_0, A, B, C, D)$ be a system with $x \in \mathcal{S}^n$. Assume that (A^1, A^0, a^0) is extendable. Then $\Sigma \vDash \mathcal{C}$ if and only if for any $n \in \mathbb{N}$, the following condition holds: for any $d_0, x_0, y_0, \ldots, d_{n+1}, x_{n+1}, y_{n+1}$, the condition:

$$\begin{cases} x_0 \in \mathcal{X}_0, & \\ G^1 \left[\begin{smallmatrix} d_{k+1} \\ y_{k+1} \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d_k \\ y_k \end{smallmatrix}\right] \le g^0, & \forall k = 0, \ldots, n-1, \\ A^1 d_{k+1} + A^0 d_k \le a^0, & \forall k = 0, \ldots, n, \\ x_{k+1} = A x_k + B d_k, & \forall k = 0, \ldots, n, \\ y_k = C x_k + D d_k, & \forall k = 0, \ldots, n+1, \end{cases} \tag{4}$$

implies $G^1 \left[\begin{smallmatrix} d_{n+1} \\ y_{n+1} \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d_n \\ y_n \end{smallmatrix}\right] \le g^0$.

Proof. See Sharf et al. (2020).

The previous theorem allows one to verify that a given system satisfies a given contract by proving a sequence of (infinitely many) implications of the form (4). Roughly speaking, this implication guarantees that if the system implements the contract up to time n, then it implements the contract up to time $n+1$. Even though this formulation requires infinitely many steps in general, we will soon see that only finitely many implications of the form (4) needs to be verified. Importantly, the implication (4) can be cast as an optimization problem. For any $n, p \in \mathbb{N}$ such that $n \geq p$, we consider the following optimization problem:

$$\begin{aligned}
\max \quad & \max_i \ \left[\mathrm{e}_i^\top \left(G^1 \left[\begin{smallmatrix} d_{n+1} \\ y_{n+1} \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d_n \\ y_n \end{smallmatrix}\right] - g^0\right)\right] && (5)\\
\text{s.t.} \quad & G^1 \left[\begin{smallmatrix} d_{k+1} \\ y_{k+1} \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d_k \\ y_k \end{smallmatrix}\right] \leq g^0 && , \forall k = p, \ldots, n-1,\\
& A^1 d_{k+1} + A^0 d_k \leq a^0 && , \forall k = p, \ldots, n,\\
& x_{k+1} = A x_k + B d_k && , \forall k = p, \ldots, n,\\
& y_k = C x_k + D d_k && , \forall k = p, \ldots, n+1,\\
& x_p \in \mathcal{X}_p, &&\\
& d_k \in \mathbb{R}^{n_d}, x_k \in \mathbb{R}^{n_x}, y_k \in \mathbb{R}^{n_y}, \forall k = p, \ldots, n+1, &&
\end{aligned}$$

where $\mathrm{e_i}$ are the standard basis elements, and $\mathcal{X}_p$ for $p = 1, 2, \ldots, n$ are sets to be defined later. We denote this problem as $V_{n,n-p}$ and let $\theta_{n,n-p}$ be its value. Here, n is the last time at which the we know the guarantee holds, p is the first time we consider, and $\ell = n - p$ is the length of history we consider. When taking $p = 0$, the problem (5) computes the "worst-case violation" of the guarantee at time $n + 1$, given that the guarantees hold up to time n. For that reason, Theorem 3 can be restated as:

Corollary 4. Under the assumptions of Theorem 3, $\Sigma \vDash \mathcal{C}$ if and only if $\theta_{n,n} \leq 0$ for all $n \in \mathbb{N}$.

Proof. $\theta_{n,n} \leq 0$ if and only if whenever (4) holds, $G^1 \left[\begin{smallmatrix} d_{n+1} \\ y_{n+1} \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d_n \\ y_n \end{smallmatrix}\right] - g^0 \leq 0$ also holds, which is equivalent to $\Sigma \vDash \mathcal{C}$ by Theorem 3. □

The corollary implies that it suffices to compute $\theta_{n,n}$ for all $n \in \mathbb{N}$ in order to verify $\Sigma \vDash \mathcal{C}$. We however prefer to compute $\theta_{n,\ell}$ for small $\ell = n - p$, as this leads to a simpler problem that can be solved more efficiently with existing numerical methods. The main difficulty in reducing the verification to problems $V_{n,\ell}$ for small ℓ is that it requires knowledge of the state trajectory $x(\cdot)$ at time $p = n - l$, captured in (5) via the constraint $x_p \in \mathcal{X}_p$. This simply reduces to the initial value $x_0 \in \mathcal{X}_0$ for problems $V_{n,n}$.

An efficient solution of $V_{n,\ell}$ for small ℓ requires a characterization of $\mathcal{X}_p$ satisfying the following criteria. First, it is desirable that $\mathcal{X}_p$ is a polyhedral set [1] , as (5) reduces to a linear problem for which efficient solvers are available, e.g., Yalmip (Löfberg (2004)). Second, we would like $\mathcal{X}_p$ to be *independent* of p, as this will imply verification of contract satisfaction can be done by solving a finite number of optimization problems (thus not requiring the computation of all $\theta_{n,n}$ as in Corollary 4). Third, $V_{n,\ell}$ is equivalent to $V_{n+1,\ell}$ where $\mathcal{X}_{p+1}$ is the image of $\mathcal{X}_p$ under the dynamics $x_{p+1} = A x_p + B d_p$. Combining the last two points, we search for $\mathcal{X}_p$ which is a robust invariant set.

However, these criteria might be contradictory. The last two dictate choosing $\mathcal{X}_p$ as smallest robust invariant set containing $\mathcal{X}_0$, but this set might not be polyhedral even if $\mathcal{X}_0 = \{0\}$ (Fisher and Gayek (1988)). In fact, the question of whether the minimal robust invariant set containing $\mathcal{X}_0 = \{0\}$ is polyhedral is related to the rationality of the eigenvalues of the matrix A of Σ. We can try and find some polyhedral robust invariant set containing $\mathcal{X}_0$, not necessarily the smallest one. Rakovic et al. (2005) offer a very partial solution for $\mathcal{X}_0 = \{0\}$, but a general solution is not known to the authors.

To avoid these difficulties, we simply set $\mathcal{X}_p = \mathbb{R}^{n_x}$, at the cost of a more conservative test for contract implementation. Namely, a choice of $\mathcal{X}_p$ that is larger than necessary (i.e., larger than the smallest robust invariant set containing $\mathcal{X}_0$) will make the demand $\theta_{n,\ell} \leq 0$ stricter. The following theorem formalizes this case.

Theorem 5. Let $\Sigma = (\mathcal{X}_0, A, B, C, D)$ be a system, and let ν be its observability index. Take a contract $\mathcal{C} = (\mathcal{D}, \Omega)$ such that (2) and (3) hold. Define $\mathcal{X}_p = \mathbb{R}^{n_x}$ for all $p \neq 0$. The following claims hold:

- For any $n \in \mathbb{N}, \theta_{n,n} \leq \theta_{n,n-1} \leq \theta_{n,n-2} \leq \ldots \leq \theta_{n,0}$. Moreover, for any $\ell \geq 0$, we have $\theta_{\ell,\ell} \leq \theta_{\ell+1,\ell} = \theta_{\ell+2,\ell} = \theta_{\ell+3,\ell} = \cdots$.
- Suppose $\mathcal{D}_\star = \{(d_0, d_1) : A^1 d_1 + A^0 d_0 \leq a^0\}$ is bounded, and that for any bounded set $E \subseteq \mathbb{R}^{2n_d}$, the intersection of $E \times \mathbb{R}^{2n_y}$ with $\Omega_\star = \{(d_0, d_1, y_0, y_1) : G^1 \left[\begin{smallmatrix} d_1 \\ y_1 \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d_0 \\ y_0 \end{smallmatrix}\right] \leq g^0\}$ is bounded. Then $\theta_{n,\ell} < \infty$ for $n \geq \ell \geq \nu - 1$, and $\theta_{n,\ell} = \infty$ if $n, \nu - 1 > \ell$.
- Given $\ell \geq 0$, if $\theta_{n,n} \leq 0$ for any $0 \leq n < \ell$ and $\theta_{\ell+1,\ell} \leq 0$, then $\Sigma \vDash \mathcal{C}$.

Proof. See Sharf et al. (2020).

Remark 2. Theorem 5 shows that $\theta_{\ell+1,\ell} = \infty$ if $\ell \leq \nu - 2$. Thus, we will use the third part of Theorem 5 for $\ell = \nu - 1$ to verify implementation.

Remark 3. Consider $V_{n,p}$ for $\mathcal{X}_p = \mathbb{R}^{n_x}$. By using the transfer function associated with the state-space system (A, B, C, D), we can find matrices $E_1, \ldots, E_m$ and $F_0, \ldots, F_m$ such that the state-space representation is equivalent to the recursive equation $y(k) = \sum_{r=1}^m E_r y(k - r) + \sum_{r=0}^m F_r d(k - r)$. Thus, $V_{n,\ell}$ for $\ell \geq m$, $p = n - \ell$ and $\mathcal{X}_p = \mathbb{R}^{n_x}$ can be recast as:

$$\begin{aligned}
\max \quad & \max_i \left[\mathrm{e_i}^\top \left(G^1 \left[\begin{smallmatrix} d_{n+1} \\ y_{n+1} \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d_n \\ y_n \end{smallmatrix}\right] - g^0\right)\right] && (6)\\
\text{s.t.} \quad & G^1 \left[\begin{smallmatrix} d_{k+1} \\ y_{k+1} \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d_k \\ y_k \end{smallmatrix}\right] \leq g^0 && , \forall k = p, \ldots, n-1,\\
& A^1 d_{k+1} + A^0 d_k \leq a^0 && , \forall k = p, \ldots, n,\\
& y_k = \sum_{r=1}^m E_r y_{k-r} + \sum_{r=0}^m F_r d_{k-r} &&\\
& && , \forall k = p + m, \ldots, n+1,\\
& d_k \in \mathbb{R}^{n_d}, y_k \in \mathbb{R}^{n_y} && , \forall k = p, \ldots, n+1.
\end{aligned}$$

This reformulation of (5) is more computationally efficient, as it removes a large number of constraints and variables.

To conclude this section, we showed one can verify a system Σ satisfies a contract $\mathcal{C}$ by solving $\nu + 1$ linear programs, where ν is the observability index of the system. The first ν problems assert that $\theta_{n,n} \leq 0$ for $n = 0, \ldots, \nu - 1$, and the last asserts that $\theta_{\nu+1,\nu} \leq 0$. The first ν problems deal with the initial conditions of the system, and the last problem deals with the long-term behaviour of the system.

3.2 Verifying Refinement

In this section, we prescribe computational tools for verifying refinement between contracts defined by linear inequalities. These tools are similar to the ones presented in the work of Sankaranarayanan et al. (2005).

[1] i.e., it is of the form $\{x : Fx \leq f\}$ for a matrix F and a vector f.

Consider now two contracts $\mathcal{C}_1 = (\mathcal{D}_1, \Omega_1)$ and $\mathcal{C}_2 = (\mathcal{D}_2, \Omega_2)$ of the form (2) and (3), i.e.:

$$\begin{aligned} \mathcal{D}_1 &= \{d(\cdot) : A^1 d(k+1) + A^0 d(k) \le a^0,\ \forall k\}, \qquad (7)\\ \Omega_1 &= \{(d(\cdot), y(\cdot) : G^1 \left[\begin{smallmatrix} d(k+1) \\ y(k+1) \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d(k) \\ y(k) \end{smallmatrix}\right] \le g^0,\ \forall k\},\\ \mathcal{D}_2 &= \{d(\cdot) : B^1 d(k+1) + B^0 d(k) \le b^0,\ \forall k\},\\ \Omega_2 &= \{(d(\cdot), y(\cdot) : H^1 \left[\begin{smallmatrix} d(k+1) \\ y(k+1) \end{smallmatrix}\right] + H^0 \left[\begin{smallmatrix} d(k) \\ y(k) \end{smallmatrix}\right] \le h^0,\ \forall k\}, \end{aligned}$$

for some $A^1, A^0, G^1, G^0, B^1, B^0, H^1, H^0, a^0, g^0, b^0, h^0$. We search for a computationally viable way to verify that $\mathcal{C}_1 \preccurlyeq \mathcal{C}_2$. It suffices to show that any $d \in \mathcal{D}_2$ satisfies $d \in \mathcal{D}_1$, and that if $(d, y) \in \Omega_1$ and $d \in \mathcal{D}_2$ then $(d, y) \in \Omega_2$. As before, we can use inductive reasoning:

Proposition 6. Let $\mathcal{C}_1, \mathcal{C}_2$ be contracts as in (7), where $G^1 = [G^1_d, G^1_y]$ and $G^0 = [G^0_d, G^0_y]$, and assume both $\left(\left[\begin{smallmatrix} B^1 & 0 \\ G^1_d & G^1_y \end{smallmatrix}\right], \left[\begin{smallmatrix} B^0 & 0 \\ G^0_d & G^0_y \end{smallmatrix}\right], \left[\begin{smallmatrix} b^0 \\ g^0 \end{smallmatrix}\right]\right)$ and (B^1, B^0, b^0) are extendable. $\mathcal{C}_1 \preccurlyeq \mathcal{C}_2$ if and only if the following two implications hold for any d_0, d_1, y_0, y_1:

- If $B^1 d_1 + B^0 d_0 \le b^0$, then $A^1 d_1 + A^0 d_0 \le a^0$.
- If $B^1 d_1 + B^0 d_0 \le b^0$ and $G^1 \left[\begin{smallmatrix} d_1 \\ y_1 \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d_0 \\ y_0 \end{smallmatrix}\right] \le g^0$, then $H^1 \left[\begin{smallmatrix} d_1 \\ y_1 \end{smallmatrix}\right] + H^0 \left[\begin{smallmatrix} d_0 \\ y_0 \end{smallmatrix}\right] \le h^0$.

Proof. See Sharf et al. (2020).

Similarly to the previous subsection, we can verify these implications using linear optimization problems:

Theorem 7. Suppose the assumptions of Proposition 6 hold. $\mathcal{C}_1 \preccurlyeq \mathcal{C}_2$ if and only if $\psi_{\mathcal{D}}$ and ψ_Ω, the optimal values of the problems below, are non-positive:

$$\begin{aligned} \psi_{\mathcal{D}} = \max \quad & \max_i \left[\mathrm{e_i}^\top \left(A^1 d_1 + A^0 d_0 - a^0 \right) \right] \\ \text{s.t.} \quad & B^1 d_1 + B^0 d_0 \le b^0, \quad d_0, d_1 \in \mathbb{R}^{n_d} \\ \psi_\Omega = \max \quad & \max_i \left[\mathrm{e_i}^\top \left(H^1 \left[\begin{smallmatrix} d_1 \\ y_1 \end{smallmatrix}\right] + H^0 \left[\begin{smallmatrix} d_0 \\ y_0 \end{smallmatrix}\right] - h^0 \right) \right] \\ \text{s.t.} \quad & G^1 \left[\begin{smallmatrix} d_1 \\ y_1 \end{smallmatrix}\right] + G^0 \left[\begin{smallmatrix} d_0 \\ y_0 \end{smallmatrix}\right] \le g^0, \quad B^1 d_1 + B^0 d_0 \le b^0 \\ & d_0, d_1 \in \mathbb{R}^{n_d},\ y_0, y_1 \in \mathbb{R}^{n_y} \end{aligned}$$

Proof. Follows from Proposition 6, as the implications hold if and only if $\psi_{\mathcal{D}}$ and ψ_Ω are non-positive. ■

To conclude this section, we showed that for contracts defined by time-independent linear inequalities, satisfaction and refinement can be verified using linear programming.

4. SIMULATION EXAMPLE

We exemplify the tools prescribed in Section 3.1 using a simulation example. An application example of the tools in Section 3.2 is available in Sharf et al. (2020).

Consider two vehicles driving along a single-lane highway, as in Fig. 2. We are given a headway $h > 0$, and our goal is to verify that the follower keeps at least the given headway from the leader. Denoting the position and velocity of the follower as $p_1(k)$, $v_1(k)$, and the position and velocity of the leader as $p_2(k), v_2(k)$, we want to show that $p_2(k) - p_1(k) - hv_1(k) \ge 0$ holds at any time $k \in \mathbb{N}$. We address this problem using assume/guarantee contracts.

The input signal to the follower $d(\cdot)$ is $d(k) = [p_2(k), v_2(k)]$. It is reasonable to assume the leader vehicle follows the kinematic laws, i.e.,

$$\begin{aligned} & p_2(k+1) = p_2(k) + \Delta t v_2(k),\ v_2(k+1) = v_2(k) + \Delta t a_2(k), \\ & a_2(k) \in [-a_{\min}, a_{\max}] \end{aligned}$$

where $a_2(k)$ is the acceleration to the leading vehicle and $\Delta t > 0$ is the length of a discrete time step. As for guarantees, we want to assure that $p_2(k) - p_1(k) - hv_1(k) \ge 0$ holds for any $k \in \mathbb{N}$. It is clear that these assumptions and guarantees are given by linear inequalities, meaning that the methods of Section 3 can be applied. Explicitly, the set of assumptions is of the form (2) and the set of guarantees is of the form (3), for:

$$\begin{aligned} A^1 &= \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad A^0 = \begin{bmatrix} -1 & -\Delta t \\ 1 & \Delta t \\ 0 & -1 \\ 0 & 1 \end{bmatrix} \quad a^0 = \begin{bmatrix} 0 \\ 0 \\ \Delta t a_{\max} \\ \Delta t a_{\min} \end{bmatrix}, \\ G^1 &= [\, 0\ 0\ 0\ 0 \,], \quad G^0 = [\, -1\ 0\ 1\ h \,], \quad g^0 = [0]. \end{aligned}$$

We must also specify the system. We assume the follower vehicle also satisfies the kinematic laws, with an acceleration dictated by an affine control law:

$$\begin{aligned} & p_1(k+1) = p_1(k) + \Delta t v_1(k),\ v_1(k+1) = v_1(k) + \Delta t a_1(k), \\ & a_1(k) = \frac{p_2(k) - p_1(k)}{h \Delta t} - \left(\frac{1}{h} + \frac{1}{\Delta t} \right) v_1(k) + \frac{v_2(k)}{h} - 1_{\mathrm{m/s^2}}, \end{aligned}$$

In other words, the follower can be modeled by a system Σ defined by the equations $x(k+1) = Ax(k) + Bd(k) + w$, $y(k) = Cx(k) + Dd(k)$, where $x = y = [p_1, v_1]^\top$, $d = [p_2, v_2]^\top$, $\mathcal{X}_0$ depends on $d(0)$ as we assume the initial state satisfies $p_2(0) - p_1(0) - hv_1(0) \ge 0$ (see Remark 1), and the dynamics are given by the matrices:

$$A = \begin{bmatrix} 1 & \Delta t \\ -\frac{1}{h} & -\frac{\Delta t}{h} \end{bmatrix},\ B = \begin{bmatrix} 0 & 0 \\ \frac{1}{h} & \frac{\Delta t}{h} \end{bmatrix},\ C = I,\ D = 0,\ w = \begin{bmatrix} 0 \\ -\Delta t \end{bmatrix}$$

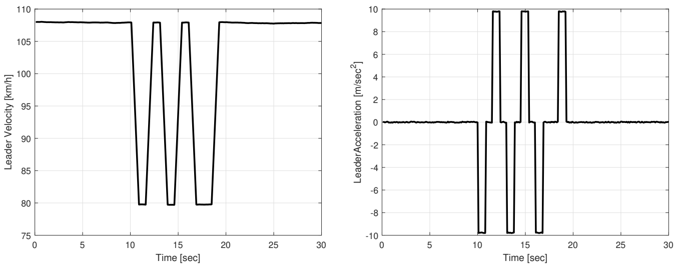
We want to prove that $\Sigma \vDash \mathcal{C}$, and we do so using Theorem 5. The system Σ is observable, and its observability index is $\nu = 1$. Thus, it suffices to prove $\theta_{0,0}, \theta_{2,1} \le 0$, where:

$$\begin{aligned} \theta_{0,0} = \max \quad & -(p_2(0) - p_1(0) - hv_1(0)) \\ \text{s.t.} \quad & p_2(0) - p_1(0) - hv_1(0) \ge 0 \\ & p_1(0), p_2(0), v_1(0), v_2(0) \in \mathbb{R} \\ \theta_{2,1} = \max \quad & -(p_2^+ - p_1^+ - hv_1^+) \\ \text{s.t.} \quad & p_2 - p_1 - hv_1 \ge 0 \\ & p_2^+ = p_2 + \Delta t v_2, \quad v_2^+ = v_2 + \Delta t a_2 \\ & a_2 \in [-a_{\min}, a_{\max}] \\ & p_1^+ = p_1 + \Delta t v_1, \quad v_1^+ = v_1 + \Delta t a_1 \\ & a_1 = \frac{p_2 - p_1}{h \Delta t} - \left(\frac{1}{h} + \frac{1}{\Delta t} \right) v_1 + \frac{v_2}{h} - 1 \\ & p_2^+, p_1^+, v_2^+, v_1^+, a_2^+, a_1^+, p_2, p_1, v_2, v_1, a_2, a_1 \in \mathbb{R}. \end{aligned}$$

In the problem defining $\theta_{2,1}$, the parameters with "+" correspond to time $k = 2$, and the ones without "+" correspond to time $k = 1$. We choose parameters $a_{\min} = a_{\max} = 9.8\mathrm{m/s^2}$, $\Delta t = 0.1$sec, $h = 2$sec, and solve both LP problems using Yalmip (Löfberg (2004)), computing $\theta_{0,0} = 0, \theta_{2,1} = -0.2$, meaning that $\Sigma \vDash \mathcal{C}$ as $\theta_{0,0}, \theta_{2,1} \le 0$.

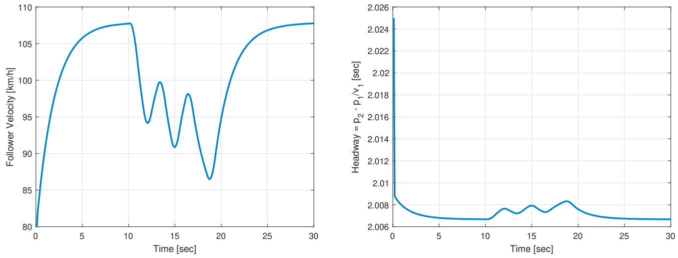
We exemplify that $\Sigma \vDash \mathcal{C}$ through simulation. We consider the following trajectory of the leader - its initial speed is about 110km/h, which is roughly kept for 10 seconds. It then starts to sway wildly for 10 seconds between 80km/h and 110km/h, braking and accelerating as hard as possible.

Fig. 2. The two-vehicle scenario of Section 4



(a) Velocity of leader (b) Acceleration of leader

Fig. 3. Leader vehicle in simulation.



(a) Velocity of follower vehicle (b) Headway between vehicles

Fig. 4. Velocity of follower vehicle and headway.

Finally, it stops swaying and keeps its velocity for 10 more seconds. The velocity and acceleration of the leader can be seen in Fig. 3. The follower starts 45m behind the leader, so the headway is kept at time 0. We run the simulation for both vehicles, and plot the headway $\frac{p_2(k)-p_1(k)}{v_1(k)}$ and the velocity of the follower in Fig. 4. It can be seen that the headway is kept throughout the run, so the guarantees are satisfied, as predicted by our analysis.

5. CONCLUSIONS AND FUTURE RESEARCH

We presented an assume/guarantee contract framework for discrete-time dynamical systems. The framework puts assumptions on the input signal to the system, and prescribes guarantees on the output relative the the input. In particular, as the guarantees do not include the state, systems of different orders can satisfy the same contract. We also defined corresponding fundamental notions such as satisfaction, refinement, and cascaded composition. Perhaps more importantly, we showed that for contracts defined using linear inequalities, satisfaction and refinement can be verified using linear programming, which can be solved efficiently using off-the-shelf optimization software. Finally, we exemplified our methods using a case study on a 2-vehicle leader-follower scenario, where the goal was to obey a certain headway. Future research can extend our results by extending the methods presented in this work for nonlinear, uncertain, or hybrid systems, as well as for verifying compositional refinement, i.e. that a composition of multiple contracts on individual components or subsystems refines a contract on the composite system.

REFERENCES

Belta, C., Yordanov, B., and Gol, E.A. (2017). *Formal methods for discrete-time dynamical systems*, volume 89. Springer.

Benveniste, A., Caillaud, B., Nickovic, D., Passerone, R., Raclet, J.B., Reinkemeier, P., Sangiovanni-Vincentelli, A.L., Damm, W., Henzinger, T.A., Larsen, K.G., et al. (2018). Contracts for system design. *Foundations and Trends in Electronic Design Automation*, 12(2-3), 124–400.

Besselink, B., Johansson, K.H., and Van Der Schaft, A. (2019). Contracts as specifications for dynamical systems in driving variable form. In *Proceedings of the 18th European Control Conference (ECC)*, 263–268.

Donaldson, A.F., Haller, L., Kroening, D., and Rümmer, P. (2011). Software verification using k-induction. In *International Static Analysis Symposium*, 351–368. Springer.

Fisher, M.E. and Gayek, J. (1988). Estimating reachable sets for two-dimensional linear discrete systems. *Journal of Optimization Theory and Applications*, 56(1), 67–88.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in matlab. In *Proceedings of the CACSD Conference*. Taipei, Taiwan.

Meyer, B. (1992). Applying 'design by contract'. *Computer*, 25(10), 40–51.

Nuzzo, P., Sangiovanni-Vincentelli, A.L., Bresolin, D., Geretti, L., and Villa, T. (2015). A platform-based design methodology with contracts and related tools for the design of cyber-physical systems. *Proceedings of the IEEE*, 103(11), 2104–2132.

Nuzzo, P., Xu, H., Ozay, N., Finn, J.B., Sangiovanni-Vincentelli, A.L., Murray, R.M., Donzé, A., and Seshia, S.A. (2014). A contract-based methodology for aircraft electric power system design. *IEEE Access*, 2, 1–25.

Phan-Minh, T., Cai, K.X., and Murray, R.M. (2019). Towards assume-guarantee profiles for autonomous vehicles. In *Proceedings of the IEEE 58th Conference on Decision and Control (CDC)*, 2788–2795. IEEE.

Rakovic, S.V., Kerrigan, E.C., Kouramas, K.I., and Mayne, D.Q. (2005). Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3), 406–410.

Sankaranarayanan, S., Sipma, H.B., and Manna, Z. (2005). Scalable analysis of linear systems using mathematical programming. In *International Workshop on Verification, Model Checking, and Abstract Interpretation*, 25–41. Springer.

Saoud, A., Girard, A., and Fribourg, L. (2018). On the composition of discrete and continuous-time assume-guarantee contracts for invariance. In *Proceedings of the European Control Conference (ECC)*, 435–440. IEEE.

Saoud, A., Girard, A., and Fribourg, L. (2019). Assume-guarantee contracts for discrete and continuous-time systems.

Sharf, M., Besselink, B., Molin, A., Zhao, Q., and Johansson, K.H. (2020). Assume/guarantee contracts for dynamical systems: Theory and computational tools. *arXiv preprint arXiv:2012.12657*.

Tabuada, P. (2009). *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media.

Wongpiromsarn, T., Topcu, U., and Murray, R.M. (2010). Receding horizon control for temporal logic specifications. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, 101–110.