53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

Feedback Refinement Relations for Symbolic Controller Synthesis

Gunther Reissig and Matthias Rungger

***Abstract*— A common issue with existing approaches to symbolic controller synthesis lies in the huge complexity of the resulting controllers. In particular, the controllers usually need full plant state information and contain an abstraction of the plant as a building block. In this note, we present an extension which helps reduce that complexity. Our technique is based on the novel concept of feedback refinement relations to compare plants with their finite-state approximations. As an additional feature, our approach builds on infinitary completed trace semantics and allows for the synthesis of controllers for arbitrary, not necessarily prefix-closed specifications. We also reveal if and how existing symbolic controller synthesis procedures should be extended to benefit from the advantages of our technique.**

I. Introduction

The use of finite-state approximations ("discrete abstractions") of typically infinite-state plants is a relatively recent approach which allows for systematic and fully automated synthesis of feedback controllers to enforce complex specifications. The controllers may be, however, complex and costly to implement, which is one of the issues that have so far prevented the technique from being routinely applied in practice. In this note we present an extension which results in controllers that are considerably less complex.

Given a control problem represented by a plant and a specification, the technique of abstraction-based controller synthesis comprises three steps [1]: First, an abstraction of the plant is computed, usually in the form of a finite automaton whose states ("abstract states") have resulted from quantizing plant states. Second, employing standard algorithms for finite systems, an auxiliary control problem is solved for the abstraction in place of the plant, which results in a finite-state feedback controller ("abstract controller") for the abstraction. Finally, by supplementing it with a suitable interface, the abstract controller is refined into a controller for the actual plant. This way the original control problem is reduced to the finite problem of synthesizing the abstract controller.

A common issue with existing approaches to abstraction-based controller synthesis lies in the huge complexity added to the abstract controller in the course of its refinement. As an example, consider the structure of the actual closed loop shown in Fig. 1(a), which has resulted from one particular refinement method that builds on the existence of *alternating simulation relations* between plants and abstractions [1]. We first notice that the input converter directly depends on the state of the plant. Hence, in contrast to a symbolic controller, which would solely rely on abstract state information, the refined controller in Fig. 1(a) requires full plant state information. This fact adds to the complexity of the closed loop, and in turn, to the cost of its implementation. Furthermore, the abstraction appears as a building block of the refined controller in Fig. 1(a). Given the fact that the abstraction may very well comprise millions of states and billions of transitions [2], an implementation of the refined controller will often be too expensive to be practical.

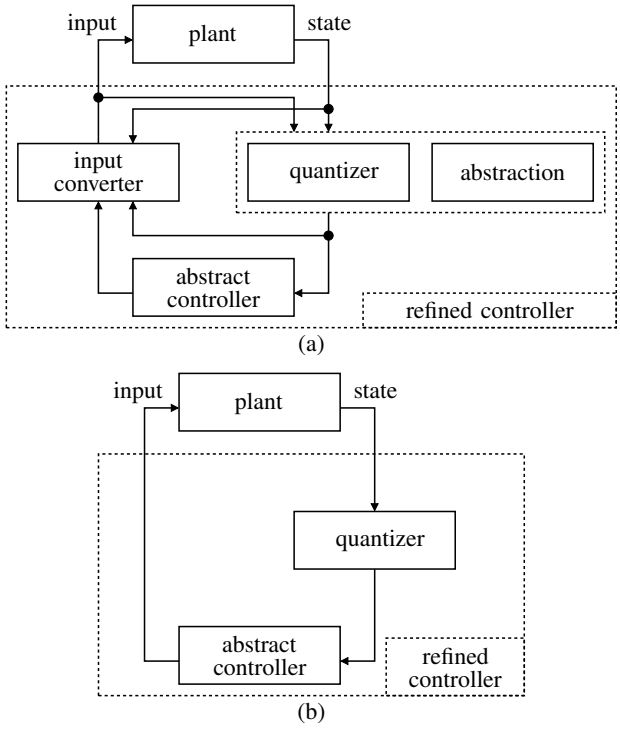

Fig. 1. (a) Actual closed loop resulting from the use of alternating simulation relations [1]. (b) Actual closed loop resulting from the use of feedback refinement relations, as proposed in this note. As our example in Section IV-A shows, the structure of this closed loop is not, in general, compatible with the use of alternating simulation relations.

Recently, methods to obtain refined controllers that are both static and symbolic have been proposed for safety problems [3], [4] and reachability problems [3]. These results require the input alphabet of the plant to be finite and the quantizer to be deterministic with non-overlapping, half-open hyper-rectangles as level sets. The latter condition

G. Reissig is with the University of the Federal Armed Forces Munich, Dept. Aerospace Eng., Chair of Control Eng. (LRT-15), D-85577 Neubiberg (Munich), Germany, http://www.reiszig.de/gunther/

M. Rungger is with the Hybrid Control Systems Group at the Department of Electrical Engineering and Information Technology at the Technische Universitaet Muenchen, 80333 Munich, Germany.

This work has been supported by the German Research Foundation (DFG) under grant no. RE 1249/3-2.

mandates that measurements be precise; measurement errors are not considered. In [3], the plant is additionally assumed to satisfy some stability condition. Further results in [4] do apply in the presence of measurement errors but rely on the use of observers, which reintroduces dynamics into the refined controller. Moreover, abstractions are assumed to be exact in this case, i.e., they must not permit any spurious transition. The computation of such abstractions is, in general, infeasible even for sampled linear time-invariant plants. To conclude, the complexity issue with refined controllers persists whenever measurement errors need to be taken into account or control problems more general than pure safety and reachability problems are to be solved. Moreover, as mentioned already above, the approach to tackle the complexity issue presented in [3] is based on certain stability properties of the plant and fails if the plant is unstable or blocking.

In this note, we propose to require the existence of *feedback refinement relations*, a novel concept to be introduced in Section III, between plants and abstractions. As it turns out, the abstract controller can then be connected to the plant using a quantizer as the only interface. See Fig. 1(b) for the resulting actual closed loop. We prove that the behavior of the latter, in the sense of infinitary completed trace semantics, is contained in the behavior of the abstract closed loop. This shows that the approach is suitable for the abstraction-based synthesis of controllers for arbitrary, not necessarily prefix-closed specifications, including liveness specifications [5]. Moreover, the approach permits to model measurement errors as we allow for non-deterministic quantizers, and it is also independent of any stability properties of the plant. In Section IV we reveal if and how existing symbolic controller synthesis procedures should be extended to benefit from the advantages of the technique we propose.

We would like to emphasize that the focus of this paper is on the relationship between plants and abstractions and its consequences for the refinement of controllers. While the definition of feedback refinement relations also suggests numerical methods to actually compute abstractions, we do not dwell into this issue here. Likewise, the auxiliary control problems for the abstractions can be solved employing standard algorithms for finite systems, e.g. [6], [7], and is not investigated here either.

II. Preliminaries

In this section, we introduce the most basic notation which is used throughout this note and provide the notions to uniformly represent plants, controllers and closed loops as dynamical systems.

A. Notation

$\mathbb{R}$, $\mathbb{Z}$ and $\mathbb{Z}_+$ denote the sets of real numbers, integers and non-negative integers, respectively, and $\mathbb{N} = \mathbb{Z}_+ \setminus \{0\}$. $[a, b]$, $]a, b[$, $[a, b[$, and $]a, b]$ denote closed, open and half-open, respectively, intervals with end points a and b. $[a; b]$, $]a; b[$, $[a; b[$, and $]a; b]$ stand for discrete intervals, e.g. $[a; b] = [a, b] \cap \mathbb{Z}$ and $[1; 4[= \{1, 2, 3\}$.

$f\colon A \rightrightarrows B$ denotes a *set-valued map* of A into B, whereas $f\colon A \to B$ denotes an ordinary map; see [8]. If f is set-valued, then f is *strict* and *single-valued* if $f(a) \neq \emptyset$ and $f(a)$ is a singleton, respectively, for every a. We identify set-valued maps $f\colon A \rightrightarrows B$ with binary relations on $A \times B$, i.e., $(a, b) \in f$ iff $b \in f(a)$. Moreover, if f is single-valued, it is identified with an ordinary map $f\colon A \to B$. The inverse mapping $f^{-1}\colon B \rightrightarrows A$ is defined by $f^{-1}(b) = \{\, a \in A \mid b \in f(a) \,\}$, and $f \circ g$ denotes the composition of f and g, $(f \circ g)(x) = f(g(x))$.

The set of maps $A \to B$ is denoted B^A, and the set of all signals that take their values in B and are defined on intervals of the form $[0; T[$ is denoted B^∞, $B^\infty = \bigcup_{T \in \mathbb{Z}_+ \cup \{\infty\}} B^{[0;T[}$.

B. Plants, controllers, and closed loops

We consider dynamical systems of the form

$$x(t+1) \in F(x(t), u(t), y(t)), \tag{1a}$$

$$y(t) \in H(x(t), u(t)), \tag{1b}$$

$$x(0) \in X_0, \tag{1c}$$

as formalized below.

II.1 Definition. *A system is a sextuple*

$$(X, X_0, U, F, Y, H), \tag{2}$$

where X, X_0, U and Y are nonempty sets, $X_0 \subseteq X$, $H\colon X \times U \rightrightarrows Y$ is strict, and $F\colon X \times U \times Y \rightrightarrows X$.
A triple $(u, x, y) \in U^{[0;T[} \times X^{[0;T[} \times Y^{[0;T[}$ is a solution of the system (2) (on $[0; T[$, starting at $x(0)$) if $T \in \mathbb{N} \cup \{\infty\}$, (1a) holds for all $t \in [0; T-1[$, (1b) holds for all $t \in [0; T[$, and (1c) holds.

We call the sets X, U, and Y the *state*, *input*, and *output alphabet*, respectively, of the system (2), X_0 is the *set of initial states*, and F and H is the *transition function* and the *output function*, respectively, of (2). The system (2) is *autonomous* if its input alphabet is a singleton, and it is *static* if its state alphabet is a singleton and its transition function is strict. We drop arguments that are elements of singleton sets and write e.g. $H(u)$ for $H(x, u)$ if the system is static.

We would like to include some remarks on our definition of a system. Firstly, the transition function F and the output function H are chosen to be set-valued to facilitate the representation of disturbances and other kinds of non-determinism. The map H is however assumed to be strict so that blocking can only be caused by the dynamic part (1a) of any system. Furthermore, we let H depend on the input to cover the important case of static feedback controllers. Otherwise, if the map H does not depend on its second argument, we say that the system (2) is *Moore*, in which case we write $H(x)$ for $H(x, u)$ in (1b). The system is *Moore with state output* if $X = Y$ and H takes the form $H(x, u) = \{x\}$. Finally, we let the transition function F depend on the output to be able to represent closed loops and refined controllers. See the definition below as well as

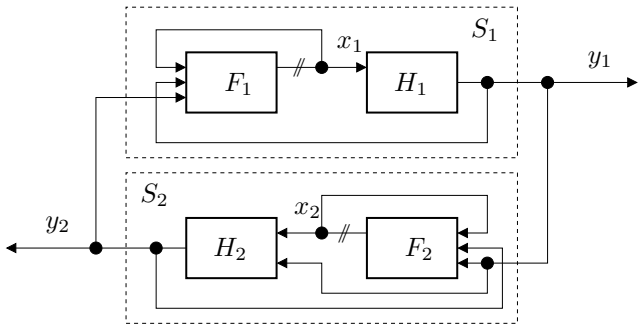

Fig. 2. Closed loop $S_1 \times S_2$ of systems S_1 and S_2 according to Definition II.2. The system S_1 is required to be Moore. The symbol // denotes a delay.

Section III-C. If the map F does not depend on its third argument, we say that the dynamics of (2) is *separated* (from the output of the system). In this case, (1a) takes the form $x(t+1) \in F(x(t), u(t))$.

II.2 Definition. *Let $S_i = (X_i, X_{i,0}, U_i, F_i, Y_i, H_i)$ be systems, $i \in \{1, 2\}$, and assume that S_1 is Moore, $Y_2 \subseteq U_1$, $Y_1 \subseteq U_2$, and that the following condition holds:*

$$y_1 \in H_1(x_1) \wedge y_2 \in H_2(x_2, y_1) \wedge F_1(x_1, y_2, y_1) = \emptyset \\ \Longrightarrow F_2(x_2, y_1, y_2) = \emptyset. \quad (3)$$

Then S_2 is feedback composable *with S_1, and the* closed loop *composed of S_1 and S_2, denoted $S_1 \times S_2$, is the sextuple*

$$(X_{12}, X_{12,0}, \{0\}, F_{12}, Y_{12}, H_{12}), \quad (4)$$

where $X_{12} = X_1 \times X_2$, $X_{12,0} = X_{1,0} \times X_{2,0}$, $Y_{12} = Y_1 \times Y_2$, and $F_{12} \colon X_{12} \times Y_{12} \rightrightarrows X_{12}$ and $H_{12} \colon X_{12} \rightrightarrows Y_{12}$ satisfy

$$F_{12}(x, y) = F_1(x_1, y_2, y_1) \times F_2(x_2, y_1, y_2), \\ H_{12}(x) = \{\, y \,|\, y_1 \in H_1(x_1) \wedge y_2 \in H_2(x_2, y_1) \,\}.$$

The closed loop $S_1 \times S_2$ defined above is obviously autonomous. However, even if S_1 and S_2 both have separated dynamics, this property does not carry over to the closed loop unless the output functions H_1 and H_2 are both single valued. See also Fig. 2. The assumption that S_1 is additionally Moore is common and ensures that $S_1 \times S_2$ does not contain a delay free cycle [9]. Note that due to requirement (3), the composition of two systems into a closed loop is not symmetric. For example, if S_1 is blocking then (3) requires S_2 to be blocking as well. However, if S_2 is blocking, this does not impose any conditions on S_1. The requirement (3), which has its analog in the theory developed in [1], is particularly important and will be needed later to ensure that if the abstract closed loop is non-blocking, then so is the actual closed loop.

III. Symbolic controller synthesis using Feedback Refinement Relations

In this section, we introduce feedback refinement relations as a novel means to compare systems. Given that the system and its abstraction is related by this novel notion, as we shall prove, any abstract controller can be connected to the plant using a simple quantizer as the only interface, such that the abstract closed loop is able to reproduce the full behavior of the actual closed loop. This implies that the refined controller, which is composed of the abstract controller and the quantizer, solves a predefined control problem for the plant provided that the abstract controller solves a suitable auxiliary problem for the abstraction.

A. Behaviors, specifications, and control problems

In the most general of terms, a control problem is a pair of a plant and a specification for which a controller is to be synthesized that is feedback composable with the plant and ensures that the closed loop satisfies the specification, e.g. [10]. Actual approaches vary in how they define the terms behavior, feedback composability, and satisfaction of a specification, and the one we follow builds on *infinitary completed trace semantics* [11].

III.1 Definition. *Let S denote the system (2). The* behavior of *S, $\mathcal{B}(S)$, is defined by*

$$\mathcal{B}(S) = \{(u, y) | \exists_x (u, x, y) \text{ is a solution of } S \text{ on } [0; T[, \text{ and if } T < \infty, \text{ then } F(x(T-1), u(T-1), y(T-1)) = \emptyset\}. \quad (5)$$

Given a set Z, any subset $\Sigma \subseteq Z^\infty$ is called a specification on *Z. The system S is said to* satisfy *a specification Σ on $U \times Y$ if $\mathcal{B}(S) \subseteq \Sigma$. Given a specification Σ on $Y \times U$, a system C* solves the control problem *(S, Σ) if C is feedback composable with S and the closed loop $S \times C$ satisfies Σ.*

In words, the behavior of a system S contains all non-continuable signals that could possibly be observed at the input and output terminals of S. A specification Σ, on the other hand, contains all signals that are considered acceptable at those terminals, and S satisfies Σ iff the full behavior of S is acceptable. Otherwise, a controller C is to be designed that eliminates any unacceptable signals from the closed loop $S \times C$. The requirement that signals be non-continuable is important as it allows for behaviors and specifications that are not prefix-closed and so enables us to cover e.g. liveness specifications [5].

B. Feedback Refinement Relations

The key point of the technique of abstraction-based controller synthesis is that the behavior of the abstract closed loop contains, in a suitable sense, the full behavior of the actual closed loop. We aim to guarantee this property, to a large extent, by computing suitable abstractions. However, in the course of these computations, the abstract controller is not yet known, and thus, the challenge is to guarantee the property without making reference to any closed loop behaviors. This is precisely what feedback refinement relations achieve by requiring appropriate relationships between the transition functions of the plant and the abstraction. In our definition below, we will use state dependent sets of admissible inputs: For any system (2) with separated dynamics and input alphabet U and any $x \in X$, we write

$$U(x) = \{\, u \in U \,|\, F(x, u) \neq \emptyset \,\}.$$

III.2 Definition. *Let $S_i = (X_i, X_{i,0}, U_i, F_i, Y_i, H_i)$ be Moore systems with state output and separated dynamics, $i \in \{1,2\}$, and assume $U_2 \subseteq U_1$.*
A strict relation $Q \subseteq X_1 \times X_2$ is a feedback refinement relation from S_1 to S_2 if the following holds for every pair $(x_1, x_2) \in Q$.

(i) $x_1 \in X_{1,0}$ *implies* $x_2 \in X_{2,0}$.
(ii) $U_2(x_2) \subseteq U_1(x_1)$.
(iii) If $u \in U_2(x_2)$ *and* $x_1' \in F_1(x_1, u)$, *then* $Q(x_1') \subseteq F_2(x_2, u)$.

Intuitively, and similarly to other kinds of simulation relations, a feedback refinement relation from the system S_1 to the system S_2 associates, with every state of S_1, one or more states of S_2, and imposes certain relations between the local dynamics of the systems in associated states. However, while e.g. alternating simulation relations only require that for each input u_2 admissible for S_2 there exists an associated input u_1 admissible for S_1 [1], our definition above additionally mandates that $u_1 = u_2$. Moreover, the definition of alternating simulation relation requires that for each transition from x_1 to x_1' in S_1 *there exists* a state x_2' associated with x_1' and a transition from x_2 to x_2' in S_2. In contrast, feedback refinement relations require the existence of the latter transition for *every* state x_2' associated with x_1'.

C. Behavioral inclusion

We are now going to exploit the stronger conditions that feedback refinement relations impose on the relation between plant and abstraction dynamics in order to obtain refined controllers that are less complex. In particular, the latter should be symbolic which suggests the use of feedback refinement relations as measurement maps. This idea turns out to be feasible, and to formulate our related, behavioral inclusion type result below, we need the notion of *series connection*. To this end, let S denote the system (2). If the dynamics of S is separated and $Q\colon Y \rightrightarrows Y'$ is strict, then the series connection of S and Q, denoted $Q \circ S$, is the sextuple

$$Q \circ S = (X, X_0, U, F, Y', Q \circ H),$$

which is obviously a system with separated dynamics. Intuitively, Q is a measurement map that yields a quantized version of the output of the system S. Alternatively, if U' is a non-empty set and $Q\colon U' \rightrightarrows U$ is strict, then the series connection of Q and S, denoted $S \circ Q$, is the sextuple

$$S \circ Q = (X, X_0, U', F', Y, H'),$$

where the maps $H'\colon X \times U' \rightrightarrows Y$ and $F'\colon X \times U' \times Y \rightrightarrows X$ are given by

$$H'(x,u) = H(x, Q(u)), \tag{6}$$
$$F'(x,u,y) = F(x, Q(u) \cap H(x,\cdot)^{-1}(y), y). \tag{7}$$

Obviously, $S \circ Q$ is a system, and Q supplies the system S with a quantized version of inputs.

III.3 Theorem. *Let Q be a feedback refinement relation from the system S_1 to the system S_2,*

$$S_i = (X_i, X_{i,0}, U_i, F_i, X_i, \mathrm{id}) \tag{8}$$

for $i \in \{1,2\}$, where $\mathrm{id}\colon X_i \to X_i$ denotes the identity map, and assume that the system C has separated dynamics and is feedback composable with S_2. Then the following holds.

(i) C is feedback composable with the series connection $Q \circ S_1$, and the series connection $C \circ Q$ is feedback composable with S_1.
(ii) $\mathcal{B}((Q \circ S_1) \times C) \subseteq \mathcal{B}(S_2 \times C)$.
(iii) For every $(0, (x_1, u)) \in \mathcal{B}(S_1 \times (C \circ Q))$ there exists a map x_2 such that $(0, (x_2, u)) \in \mathcal{B}(S_2 \times C)$ and $(x_1(t), x_2(t)) \in Q$ for all t in the domain of x_1.

D. Solution of control problems

We are now going so solve control problems using Theorem III.3. To this end, let S_1 and S_2 be systems of the form (8), in which S_1 is seen as the plant and S_2 is an abstraction of S_1. Let further Q be a feedback refinement relation from S_1 to S_2, which will appear as the quantizer in the resulting actual closed loop. See Fig. 1(b). In addition, assume that the control problem to be solved for the plant S_1 ("actual control problem") takes the form (S_1, Σ_1), where Σ_1 is a specification on $X_1 \times U_1$. (Here and in the sequel, we always identify spaces of the form $\{0\} \times V$ with V.)

III.4 Definition. *Let the systems S_1 and S_2 take the form (8), let Σ_1 be a specification on $X_1 \times U_1$, and let $Q \subseteq X_1 \times X_2$ be a strict relation. A specification Σ_2 on $X_2 \times U_2$ is called an abstract specification associated with S_1, S_2, Q and Σ_1, if the following condition holds.*
If $(0, (x_2, u)) \in \Sigma_2$, where x_2 and u are defined on $[0; T[$ for some $T \in \mathbb{N} \cup \{\infty\}$, and if $x_1\colon [0; T[\to X_1$ satisfies $(x_1(t), x_2(t)) \in Q$ for all $t \in [0; T[$, then $(0, (x_1, u)) \in \Sigma_1$.

The result presented below shows that the technique of abstraction-based controller synthesis using feedback refinement relations is feasible and leads to the actual closed loop shown in Fig. 1(b).

III.5 Theorem. *Let Q be a feedback refinement relation from the system S_1 to the system S_2, where S_1 and S_2 take the form (8). Let further Σ_1 be a specification on $X_1 \times U_1$, and assume that Σ_2 is an abstract specification associated with S_1, S_2, Q and Σ_1. Then the following holds.*
If the abstract controller C has separated dynamics and solves the abstract control problem (S_2, Σ_2), then the refined controller $C \circ Q$ solves the actual control problem (S_1, Σ_1).

The above result shows how to use a solution of an auxiliary, abstract control problem to arrive at the actual closed loop in Fig. 1(b), which solves the actual control problem. Abstractions and abstract controllers can actually be computed efficiently in many practical cases, e.g. [1]–[3], [6], [7], [12]–[17]. Details are beyond the scope of this paper. However, we will touch upon the issue of how to compute abstractions and abstract specifications in Section IV.

IV. Suitability of and extensions to previous abstraction techniques

In this section, we review existing symbolic controller synthesis procedures to see if and how they should be extended to benefit from the advantages of feedback refinement relations. In doing so, we will use the following control problem to illustrate our findings: Let us assume that we seek to steer the state of the continuous-time system

$$\dot{x} = \begin{pmatrix} -1 & -\pi \\ \pi & -1 \end{pmatrix} x + \begin{pmatrix} 2 \\ -2 \end{pmatrix} u, \quad u \in U = [-3; 3], \tag{9}$$

into the target set

$$Z_1 = [7/4, \infty[\times \mathbb{R},$$

in which we can actually measure the state and control the input of (9) at integer-valued instances of time only. This leads us to the control problem (S_1, Σ_1), where the plant S_1 is given by

$$S_1 = (\mathbb{R}^2, \mathbb{R}^2, U, F_1, \mathbb{R}^2, \mathrm{id}), \tag{10a}$$

$$F_1(x,u) = -\frac{x}{\mathrm{e}} + \frac{2(1+\mathrm{e})}{\mathrm{e}\,(1+\pi^2)} \begin{pmatrix} \pi+1 \\ \pi-1 \end{pmatrix} u, \tag{10b}$$

and the specification Σ_1 is given by

$$\Sigma_1 = \left\{ (0,(x,u)) \in \left(\{0\} \times \mathbb{R}^2 \times U\right)^\infty \,\middle|\, \exists_t\, x(t) \in Z_1 \right\}.$$

This example has been tailored for the purpose of illustrating certain properties of the abstraction-based controller synthesis procedures we are going to review, resulting in somewhat unusual choices for the problem parameters. We emphasize, however, that all relevant phenomena exhibited by this control problem persist under sufficiently small perturbations of the parameter values.

We also note that we have chosen, for the sake of simplicity, to consider abstractions with countable rather than finite state space. Finite-state abstractions could be obtained by restricting the dynamics of the plant to a positively invariant, compact subset of its state space.

A. Approach based on alternating simulation relations [1]

We shall demonstrate first that, contrary to common belief [18, Section 4.2], the structure of the closed loop in Fig. 1(b) is not, in general, compatible with the use of alternating simulation relations. To this end, we define the system S_2 and the relation Q by

$$S_2 = (\mathbb{Z}^2, \mathbb{Z}^2, U, F_2, \mathbb{Z}^2, \mathrm{id}), \tag{11a}$$

$$F_2(x,u) = \left\{ x' \in \mathbb{Z}^2 \,\middle|\, \|x' - F_1(x,u)\|_2 \le \eta \right\}, \tag{11b}$$

$$Q \colon \mathbb{R}^2 \rightrightarrows \mathbb{Z}^2 \colon x \mapsto \bar{B}_2(x,\varepsilon) \cap \mathbb{Z}^2, \tag{11c}$$

$$\eta = \sqrt{2}/2, \quad \varepsilon = 9/8, \tag{11d}$$

where $\|\cdot\|_p$ denotes the p-Norm and $\bar{B}_p(c,r)$ denotes the closed ball with radius r centered at c, with respect to $\|\cdot\|_p$. Then Q is an (ε-approximate) alternating simulation relation from S_1 to S_2 by [1, Th. 11.12] . According to the Definition III.4, the abstract specification Σ_2 associated with S_1, S_2, Q and Σ_1 is then given by

$$Z_2 = \left\{ x \in \mathbb{Z}^2 \,\middle|\, \bar{B}_2(x,\varepsilon) \subseteq Z_1 \right\} = [3; \infty[\times \mathbb{Z},$$

$$\Sigma_2 = \left\{ (0,(x,u)) \in \left(\{0\} \times \mathbb{Z}^2 \times U\right)^\infty \,\middle|\, \exists_t\, x(t) \in Z_2 \right\}.$$

To solve the abstract control problem (S_2, Σ_2), define

$$p = (2,1), \quad q = (1,1), \quad \text{and} \quad r = (3,1) \tag{12}$$

and observe that $\{q\} = F_2(p,2)$ and $\{r\} = F_2(q,3)$. In particular, $\|p' - p\|_2 > \eta$ for $p' = F_1(p,2)$, and hence, $p \notin F_2(p,2)$ by (11b). See also Fig. 3(a). It follows that for any static system C satisfying

$$C = (\{0\}, \{0\}, \mathbb{Z}^2, F_c, U, H_c), \tag{13a}$$

$$F_c(0,x) = 0, \tag{13b}$$

$$H_c(0,p) = 2, \quad H_c(0,q) = 3, \tag{13c}$$

solutions of the closed loop $S_2 \times C$ starting from the states p and q are transferred into the abstract target state r. Moreover, it is possible to choose the output function H_c on $\mathbb{Z}^2 \setminus \{p,q\}$ such that closed-loop solutions starting anywhere in $\mathbb{Z}^2$ are forced into the target set Z_2. Then the system C solves the abstract problem (S_2, Σ_2).

It is well known that the abstract controller C can be refined to solve the actual control problem [1, Section 11.3]. However, the refined controller in [1] is rather complex, see Fig. 1(a), and is not simply given by the abstract controller with the alternating simulation relation acting as an interface. We would like to emphasize that this complexity issue is

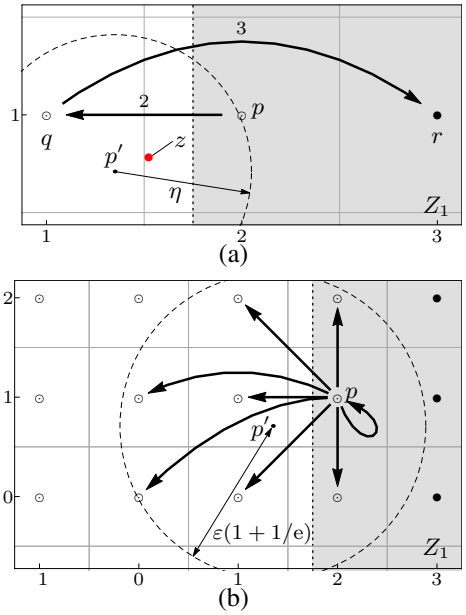

Fig. 3. Illustration of the example in Section IV; • and ⊙ denote states in the target set Z_2 and other abstract states, respectively. (a) Transitions emanating from the abstract states p and q in the closed loop $S_2 \times C$ in Section IV-A. Under the control 2, z is an equilibrium of S_1, whereas there is no loop transition in S_2 at the abstract state p which is associated with z. (b) Transitions emanating from p under the control 2 in the abstraction S_2 in Section IV-B, which is based on a feedback refinement relation. The loop transition at p ensures that the abstract controller would not enable the control 2 at p.

not merely a result of the particular controller refinement procedure, but is intrinsic to alternating simulation relations. To this end, we show that the refined controller $C \circ Q$ fails to solve the actual control problem (S_1, Σ_1). Even worse, using the strictly finer, single-valued quantizer $\widetilde{Q}$ given by

$$\widetilde{Q} \colon \mathbb{R}^2 \rightrightarrows \mathbb{Z}^2 \colon x \mapsto \left(x +]-1/2, 1/2]^2\right) \cap \mathbb{Z}^2 \tag{14}$$

does not resolve the issue. To see this, let $z = \frac{4}{1+\pi^2}\left(\begin{smallmatrix}1+\pi\\ \pi-1\end{smallmatrix}\right)$. Then $z \notin Z_1$, $\widetilde{Q}(z) = \{p\}$, and $F_1(z, H_c(0,p)) = z$. Hence, z is an equilibrium of the closed loop $S_1 \times (C \circ \widetilde{Q})$, and $(0, (z, z, \dots), (2, 2, \dots))$ is in $\mathcal{B}(S_1 \times (C \circ \widetilde{Q}))$ but not in Σ_1. The crucial point with this example is that, under the control 2, there is no loop transition in S_2 emanating from the abstract state p, whereas the actual state z associated with p by the alternating simulation relation Q is an equilibrium of the plant S_1.

B. Extension to feedback refinement relations

We now extend [1, Th. 11.12] to obtain abstractions in the sense of feedback refinement relations, and in turn, to arrive at a refined controller that is both static and symbolic. To this end, we consider the affine control system

$$\dot{x} = Ax + Bu + Cw + h, \tag{15}$$

where A, B and C are real $n \times n$, $n \times r$ and $n \times l$ matrices, respectively, and $h \in \mathbb{R}^n$. Here, u represents the input of the system which is amenable to control, and w represents a disturbance which cannot be controlled. See [1, Sec. 11.1].

The signals u and w are defined on any closed interval. We also require u and w to be measurable and locally essentially bounded (which includes piecewise continuous signals) and consider absolutely continuous solutions of (15), e.g. [19].

We aim at computing abstractions for a sampled version of the affine control system (15) using a sampling time $\tau > 0$. Formally, we consider the system

$$S_1 = (\mathbb{R}^n, \mathbb{R}^n, \mathcal{U}, F_1, \mathbb{R}^n, \mathrm{id}), \tag{16a}$$

$$\begin{aligned} F_1(\xi, u) = \{\xi' \in \mathbb{R}^n \mid \exists_{w \in \mathcal{W}} \exists_{x\colon \text{ solution of (15) on } [0,\,\tau]} \\ x(0) = \xi, x(\tau) = \xi'\}, \end{aligned} \tag{16b}$$

where $\mathcal{U}$ and $\mathcal{W}$ denote sets of admissible controls and disturbances; these sets are non-empty sets of measurable and locally essentially bounded signals $[0,\tau] \to \mathbb{R}^r$ and $[0,\tau] \to \mathbb{R}^l$, respectively.

IV.1 Theorem. *Consider the system S_1 in (16a) and define the function V by $V(x) = \langle x | Px \rangle$, where P is a real, symmetric, positive definite $n \times n$ matrix satisfying*

$$\underline{\alpha}^2 \,\mathrm{id} \le P \le \overline{\alpha}^2 \,\mathrm{id}, \tag{17}$$

$$\lambda \ge \max \sigma(A^*P + PA)/2/\underline{\alpha}^2 \tag{18}$$

for constants $\lambda \in \mathbb{R}$ and $\underline{\alpha}, \overline{\alpha} > 0$. Here, $\langle \cdot | \cdot \rangle$ denotes the Euclidean inner product, A^* is the transpose of A, and $\max \sigma(M)$ is the maximum eigenvalue of the matrix M.

Let the system S_2 be given by

$$\begin{aligned} S_2 = (X_2, X_2, \mathcal{U}, F_2, X_2, \mathrm{id}), \quad X_2 = 2\eta\mathbb{Z}^n/\sqrt{n}, \\ F_2(\xi, u) = \left\{ \xi' \in X_2 \,\middle|\, \exists_{q \in F_1(\xi,u)} V(\xi' - q) \le \underline{\alpha}\varepsilon(1 + \mathrm{e}^{\lambda\tau}) \right\}, \\ 0 < \eta \le \varepsilon\underline{\alpha}/\overline{\alpha}, \end{aligned} \tag{19}$$

for some constants η, ε satisfying (19). Then the relation $Q \subseteq \mathbb{R}^n \times X_2$ given by

$$Q(\xi_1) = \{\, \xi_2 \in X_2 \,|\, V(\xi_1 - \xi_2) \le \underline{\alpha}\varepsilon \,\}$$

is a feedback refinement relation from S_1 to S_2.

We now come back to the special case (10) of the affine control system of the present section in order to demonstrate that abstractions based on feedback refinement relations resolve the issue discussed Section IV-A. The problem was that the equilibrium state z in S_1 is associated with the non-equilibrium abstract state p. The abstraction S_2 in Theorem IV.1, with η and ε as in (11d), allows for additional transitions not present in abstractions based on alternating simulation relations. In particular, the abstract state p is now an equilibrium. See Fig. 3(b). Hence, a controller solving (S_2, Σ_2) cannot assign the control symbol 2 to the abstract state p anymore.

Finally, it is easy to see that the controller (13a), (13b) given by $H_c(0,x) = \{3\}$ if $x_1 \le -2$, and $H_c(0,x) = \{-3\}$, otherwise, solves the abstract control problem (S_2, Σ_2). Thus, by Theorems IV.1 and III.5, $C \circ Q$ solves (S_1, Σ_1) for the quantizer Q given in (11c).

C. Approach based on bisimulation relations [3]

Consider the quantizer (14) and the system (11a) given by

$$F_2(x, u) = \widetilde{Q}(F_1(x, u)).$$

Then, under suitable hypotheses, there exists an (ε-approximate) bisimulation relation between S_1 and S_2 [3, Th. 1]. Moreover, if C solves the abstract control problem (S_2, Σ_2) and is additionally a minimum-time controller with respect to the target set Z_2, then $C \circ \widetilde{Q}$ solves the actual control problem (S_1, Σ_1) and is a minimum-time controller with respect to Z_1 [3, Th. 3].

The specific parameters given in (11d), however, do not satisfy the assumptions in [3]. Indeed, it can be easily verified that given (13c), the map H_c can be extended to the whole of the abstract state space such that (13a) is a minimum time controller. Using the same arguments about the equilibrium point z as in Section IV-A we see that $C \circ \widetilde{Q}$ does not solve the actual control problem (S_1, Z_1). This shows that the hypotheses of [3, Th. 3] are not satisfied. To actually satisfy them, the parameter η could be decreased, which leads to finer state space quantization, and hence, to controllers that are more complex. Alternatively, the approximation error ε could be increased. In any case, the quantizer must be single-valued, which is an essential requirement in [3] and does not allow for modeling any measurement errors. In addition, the results in [3] only cover pure reachability problems and pure safety problems, and additionally require plant stability. We therefore recommend to consider using feedback refinement relations instead.

D. Other approaches [13], [20]

There are several successful approaches to abstraction-based controller synthesis that do not make explicit reference to any kind of simulation relation; see [13], [14] and the references given there. In particular, the quantizers considered in [13] take the form

$$Q\colon \mathbb{R}^n \rightrightarrows V\colon x \mapsto \{\, \Omega \in V \mid x \in \Omega \,\},$$

where V is a covering of $\mathbb{R}^n$ by non-empty sets. In the special case when the memory span equals 1, the abstractions S_2 computed in [13] take the form $S_2 = (V, V, U, F_2, V, \mathrm{id})$ and satisfy the following condition on the map F_2:

$$\{\, \Omega' \in V \mid \Omega' \cap F_1(\Omega, u) \neq \emptyset \,\} \subseteq F_2(\Omega, u). \qquad (20)$$

In other words, $\Omega' \in F_2(\Omega, u)$ whenever $u \in U$, $x \in \Omega \in V$, $x' = F_1(\Omega, u)$ and $x' \in \Omega'$. Consequently, the relation $\in$ considered as a subset of $\mathbb{R}^n \times V$ is a feedback refinement relation from S_1 to S_2. Therefore, the computed abstractions can be used to synthesize symbolic controllers for arbitrary specifications, including liveness specifications, once the approach in [13] has been extended to build on non-prefix-closed behaviors and to impose the condition (3) on feedback connections.

In contrast, the abstractions computed in [20] satisfy the weaker condition $F_1(\Omega, u) \subseteq \bigcup_{\Omega' \in F_2(\Omega,u)} \Omega'$ whenever $\Omega \in V$ and $u \in U$, in place of (20). Hence, the relation $\in$ is not, in general, a feedback refinement relation, and the approach suffers from the deficits we have already discussed. Therefore, the stronger condition (20) should be adopted.

V. Conclusions

We have presented a novel approach to abstraction-based controller synthesis which builds on the concept of feedback refinement relation introduced in the present paper. While our approach shares the principle of "accepting more inputs, generating fewer outputs" with other theories, e.g. [1], [3], [4], [13], [21] and the references therein, it distinguishes itself from earlier methods in that it yields controllers that are less complex. In particular, the obtained controllers are actually symbolic, i.e., they do not require full plant state information, and in the case of pure reachability and pure safety problems, the controllers are additionally static. Moreover, our approach applies to arbitrary control specifications, including liveness specifications, does not require plant stability, and allows to model measurement errors.

Methods that actually compute abstractions or solve the resulting abstract control problems are beyond the scope of this note. However, we have extended one result from [1] to yield feedback refinement relations for affine control systems subject to disturbances. We are currently working on extensions of that result to more general classes of control systems.

References

[1] P. Tabuada, *Verification and control of hybrid systems*. New York: Springer, 2009.

[2] G. Reißig, "Abstraction based solution of complex attainability problems for decomposable continuous plants," in *Proc. 49th IEEE Conf. Decision and Control (CDC), Atlanta, GA, U.S.A., 15-17 Dec. 2010*. New York: IEEE, 2010, pp. 5911–5917, DOI:10.1109/CDC.2010.5718125, free access: http://www.reiszig.de/gunther/pubs/i10product.abs.html.

[3] A. Girard, "Low-complexity quantized switching controllers using approximate bisimulation," *Nonlinear Anal. Hybrid Syst.*, vol. 10, pp. 34–44, 2013.

[4] E. Dallal, A. Colombo, D. Del Vecchio, and S. Lafortune, "Supervisory control for collision avoidance in vehicular networks with imperfect measurements," in *Proc. 52th IEEE Conf. Decision and Control (CDC), Florence, Italy, 10-13 Dec. 2013*. New York: IEEE, 2013, pp. 6298–6303.

[5] H. Völzer and D. Varacca, "Defining fairness in reactive and concurrent systems," *J. ACM*, vol. 59, no. 3, pp. 13:1–37, 2012.

[6] O. Kupferman and M. Y. Vardi, "Model checking of safety properties," in *Proc. 11th Intl. Conf. Computer Aided Verification (CAV), Trento, Italy, Jul.6-10, 1999*, ser. Lect. Notes Computer Science. Springer, Berlin, 1999, vol. 1633, pp. 172–183.

[7] R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Sa'ar, "Synthesis of reactive(1) designs," *J. Comput. System Sci.*, vol. 78, no. 3, pp. 911–938, 2012.

[8] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, ser. Grundlehren der Mathematischen Wissenschaften. Berlin: Springer-Verlag, 1998, vol. 317, 3rd corr printing 2009.

[9] M. Vidyasagar, *Input-output analysis of large-scale interconnected systems*, ser. Lect. Notes Control Inform. Sciences. Berlin: Springer-Verlag, 1981, vol. 29, decomposition, well-posedness and stability.

[10] J. C. Willems, "On interconnections, control, and feedback," *IEEE Trans. Automat. Control*, vol. 42, no. 3, pp. 326–339, 1997.

[11] R. J. van Glabbeek, "The linear time–branching time spectrum. I. The semantics of concrete, sequential processes," in *Handbook of process algebra*, J. A. Bergstra, A. Ponse, and S. A. Smolka, Eds. Amsterdam: North-Holland, 2001, pp. 3–99.

[12] G. Reißig, "Computation of discrete abstractions of arbitrary memory span for nonlinear sampled systems," in *Proc. 12th Intl. Conf. Hybrid Systems: Computation and Control (HSCC), San Francisco, U.S.A., Apr. 13-15, 2009*, ser. Lect. Notes Computer Science, R. Majumdar and P. Tabuada, Eds., vol. 5469. Springer, 2009, pp. 306–320, DOI:10.1007/978-3-642-00602-9_22, free access: http://www.reiszig.de/gunther/pubs/i09HSCC.abs.html.

[13] G. Reißig, "Computing abstractions of nonlinear systems," *IEEE Trans. Automat. Control*, vol. 56, no. 11, pp. 2583–2598, Nov. 2011, DOI:10.1109/TAC.2011.2118950, arXiv:0910.2187.

[14] M. Rungger and O. Stursberg, "On-the-fly model abstraction for controller synthesis," in *American Control Conference (ACC)*, 2012, pp. 2645–2650.

[15] G. Reißig, "Convexity of reachable sets of nonlinear ordinary differential equations," *Automat. Remote Control*, vol. 68, no. 9, pp. 1527–1543, Sep. 2007, DOI:10.1134/S000511790709007X, arXiv:1211.6080, Russian transl. in Avtomat. i Telemekh., 2007, no. 9, pp. 64-78.

[16] A. Weber and G. Reißig, "Local characterization of strongly convex sets," *J. Math. Anal. Appl.*, vol. 400, no. 2, pp. 743–750, Apr. 2013, DOI:10.1016/j.jmaa.2012.10.071, arXiv:1207.4347.

[17] A. Weber and G. Reissig, "Classical and strong convexity of sublevel sets and application to attainable sets of nonlinear systems," *SIAM J. Control Optim.*, vol. 52, 2014, accepted. arXiv:1311.4989.

[18] M. Mazo, Jr. and P. Tabuada, "Symbolic approximate time-optimal control," *Systems Control Lett.*, vol. 60, no. 4, pp. 256–263, 2011.

[19] D. L. Lukes, *Differential equations*, ser. Mathematics in Science and Engineering. London: Academic Press Inc. [Harcourt Brace Jovanovich Publishers], 1982, vol. 162, classical to controlled.

[20] G. Reissig and M. Rungger, "Abstraction-based solution of optimal stopping problems under uncertainty," in *Proc. IEEE Conf. Decision and Control (CDC), Florence, Italy, 10-13 Dec. 2013*. New York: IEEE, 2013, pp. 3190–3196, DOI:10.1109/CDC.2013.6760370.

[21] S. Tripakis, B. Lickly, T. A. Henzinger, and E. A. Lee, "A theory of synchronous relational interfaces," *ACM Trans. Program. Lang. Syst.*, vol. 33, no. 4, p. 14, 2011.

[22] M. Rungger, M. Mazo, and P. Tabuada, "Specification-guided controller synthesis for linear systems and safe linear-time temporal logic," in *Proc. 16th Intl. Conf. Hybrid Systems: Computation and Control (HSCC)*. ACM, 2013, pp. 333–342.