49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Abstraction based solution of complex attainability problems for decomposable continuous plants

Gunther Reißig

***Abstract*—The focus of the present paper is systems of nonlinear continuous sub-plants that share a common input but are otherwise coupled only through the specification of a control problem, possibly including state constraints. Examples include cart-pole systems and collision avoidance problems involving multiple vehicles. We propose a method that uses finite state models for solving highly complex continuous attainability problems.**

We first prove that finite state models, also called discrete abstractions, of the overall plant may be obtained as products of abstractions of sub-plants. The latter, which we call factors, may be determined quickly and concurrently. We also modify a state-of-the-art algorithm for the discrete, auxiliary attainability problems that arise, to work directly with the set of factors and prove that the asymptotic computational complexity of the modified algorithm matches that of the original one. In practice, the latter will often be much slower since the representation of the abstraction of the overall plant on a computer is likely to require an excessive amount of memory. Practicability of our method is demonstrated by successfully designing discrete controllers that globally stabilize decomposable nonlinear continuous plants whose overall finite state models would include millions of states and billions of transitions. Working with the factors instead, problem data fit into main memory of a customary personal computer, and computations take only minutes.

I. Introduction

In recent years, the idea of using finite state models, also called discrete abstractions, for solving analysis and synthesis problems in continuous and hybrid systems has attracted a lot of interest; see [1], [2] and the references given there. This approach requires to first determine a sufficiently accurate discrete abstraction and then to solve discrete auxiliary problems for the latter, in order to obtain provably correct statements for the original continuous or hybrid system, or alternatively, to obtain discrete controllers that provably enforce a specification for the original system. However, suitable finite state models may be extremely costly to determine, and the auxiliary discrete problems that arise may also be quite complex, which is why practicability of the above approach is currently limited to low-dimensional or otherwise special plant dynamics.

In this paper, we propose a method for solving highly complex attainability problems, which exploits a nonlinear continuous plant's structure by working directly with finite state models of sub-plants. Specifically, we consider discrete-time systems

$$x(k+1) = G(x(k), u(k)) \tag{1}$$

The author is with Universität Kassel, Fachbereich 16 - Elektrotechnik/Informatik, Regelungs- und Systemtheorie, Wilhelmshöher Allee 73, D-34121 Kassel, Germany, http://www.reiszig.de/gunther/

whose right hand side G decomposes into p factors that are coupled through a common input,

$$G(x,u) = (G_1(x_1,u), \ldots, G_p(x_p,u)). \tag{2}$$

Here, x represents an output signal, $x = (x_1, \ldots, x_p) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_p}$, and u is an input signal which is usually assumed to take its values in some finite set. The method we propose also applies if the system (1) arises from a continuous-time system

$$\dot{x} = F(x,u), \tag{3}$$

$$F(x,v) = (F_1(x_1,u), \ldots, F_p(x_p,u)) \tag{4}$$

under sampling, in which case the right hand side G is not explicitly given.

The attainability problems we consider are defined by an initial set X_0, a target set X_1, and a set X_2 of admissible states. We seek to design a controller that forces the state x of (1) into the target set X_1 within finite time, from everywhere in the initial set X_0, and also guarantees that the state constraint $x \in X_2$ is satisfied. Hence, in addition to the coupling through a common input u, the sub-systems

$$x_i(k+1) = G_i(x_i(k), u(k)) \tag{5}$$

are coupled through the specification of a control problem. Note that, in contrast to the right hand side G of (1), the sets X_0, X_1 and X_2 are not assumed to possess any product structure.

To fix ideas, we next explain the approach we follow by means of an example. Consider a system of $p-1$ pendula mounted on a common cart, $p \geq 2$. See Fig. 1(a). Here, $x_{p,1}$ denotes the position of the cart, $x_{i,1}$ is the angle between the ith pendulum and the downward vertical, and the acceleration u of the cart is considered a control. The motion of the ith pendulum may be described by

$$\dot{x}_{i,1} = x_{i,2}, \tag{6a}$$

$$\dot{x}_{i,2} = -\omega_i^2 \sin(x_{i,1}) - u\,\omega_i^2 \cos(x_{i,1}) - 2\gamma_i x_{i,2}, \tag{6b}$$

where ω_i and γ_i are parameters, and the motion of the cart, by

$$\dot{x}_{p,1} = x_{p,2}, \tag{7a}$$

$$\dot{x}_{p,2} = u. \tag{7b}$$

Hence, the right hand side of the system (7)-(6) matches the decomposition (4) with $x_i = (x_{i,1}, x_{i,2})$.

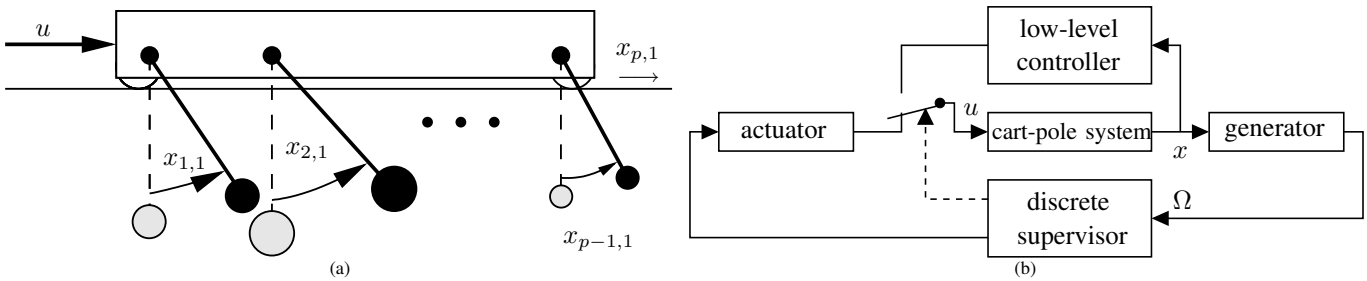

Figure 1. (a) Cart-pole system which decomposes into p sub-systems. The pendula are swung up and the cart is moved to the origin by the hybrid control system (b) [1].

Assume that the pendula should be swung up and the cart should be moved to the origin by means of the hybrid control system shown in Fig. 1(b). Clearly, it is straightforward to design a low-level controller that locally stabilizes the target point $(\pi, 0, \pi, 0, \ldots, \pi, 0, 0, 0) \in \mathbb{R}^{2p}$, provided that the linearization of the overall system is controllable at that point, which would depend on the particular values of the parameters in (6). Thus, it remains to force the state x of the system into the stability region of the target point, after what control can be handed over to the low-level controller. In this instance of the attainability problem, the initial set X_0 could be a neighborhood of the origin in $\mathbb{R}^{2p}$, the target set X_1, a positive invariant subset of the stability region of the target point, and the set $X_2 \subseteq \mathbb{R}^{2p}$ of admissible states could be defined by constraints on the position of the cart. This kind of problem may be solved in the following way [1], [3]: First, the system (6)-(7) is sampled, which results in a discrete-time system of form (1). Then the state space of the sampled system (1) is quantized by supplementing (1) at its output with a static system

$$\Omega(k) \in Q(x(k)). \tag{8}$$

In Fig. 1(b), translation between continuous-time and discrete-time signals as well as quantization is realized by interface devices. Specifically, the open loop composed of actuator, cart-pole system and generator is represented by the sampled and quantized system (1),(8). The latter, in turn, is approximated by a *discrete abstraction*, a term reserved throughout this paper for a superset of the behavior of (1),(8) in the sense of WILLEMS [4] that can be represented by a finite automaton. That is, the automaton is capable of generating every signal that could possibly be generated by (1),(8), but may (and will) additionally generate spurious signals.

Finally, to determine a supervisor, a discrete attainability problem for the abstraction needs to be solved, which reduces to a distance problem in some hypergraph. A supervisor obtained in this way would solve the original, continuous attainability problem.

There are two difficulties with the approach outlined above, namely, computation of an abstraction of the overall $2p$-dimensional plant (1),(8) could be extremely time consuming, and the hypergraph for which distances had to be computed could be excessively large. In contrast, the method we propose in this paper requires determining a total of p factor abstractions of 2-dimensional sampled and quantized sub-systems, each corresponding to one of the p systems (7) and (6), and the overall hypergraph is implicitly represented by much smaller sub-graphs, which correspond to the factor abstractions.

The remaining of this paper is structured as follows. In section II, we review notation and terminology related to systems, behaviors and hypergraphs. In section III, the class of attainability problems we intend to solve is formally given and reduced to a distance problem in a hypergraph representing a finite state model of the plant. In section IV, we derive an analog of the method from section III which takes advantage of the structure of decomposable plants and is capable of solving highly complex, previously intractable attainability problems. Finally, in section V, we demonstrate the application of the method from section IV to the pendulum example that has been discussed earlier in the present section.

II. PRELIMINARIES

A. Basic notation and terminology

$\mathbb{R}$ and $\mathbb{Z}$ denote the sets of real numbers and integers, respectively, and $\mathbb{R}_+$ and $\mathbb{Z}_+$, their subsets of non-negative elements. $[a, b]$, $]a, b[$, $[a, b[$, and $]a, b]$ denote closed, open and half-open, respectively, intervals with end points a and b, e.g. $[0, \infty[= \mathbb{R}_+$. $[a; b]$, $]a; b[$, $[a; b[$, and $]a; b]$ stand for discrete intervals, e.g. $[a; b] = [a, b] \cap \mathbb{Z}$. We endow $\mathbb{R}^n$ with the standard Euclidean product $\langle \cdot | \cdot \rangle$, i.e., $\langle x | y \rangle = \sum_{i=1}^{n} x_i y_i$ for any $x, y \in \mathbb{R}^n$. For any sets A and B, $\mathcal{P}(A)$ is the power set of A, and $f \colon A \to B$ denotes a map of A into B.

B. Behaviors

Here and in the sequel, B^A denotes the set of all maps $A \to B$. Given an arbitrary set W called *signal alphabet*, any subset $B \subseteq W^{\mathbb{Z}_+}$ is a *behavior* on W. Hence, elements of B are infinite sequences $w \colon \mathbb{Z}_+ \to W$, which we call *signals*. We interchangeably use $w(k)$ and w_k to denote the value of the signal w at time k. We often identify the restriction $w|_{[k,k']}$ of a signal $w \colon \mathbb{Z}_+ \to W$ to $[k, k']$ with the tuple $(w(k), w(k+1), \ldots, w(k'))$, which implies we identify $w|_{[k,k']}$ with $(\sigma^k w)|_{[0,k'-k]}$. Here, σ^τ denotes the *backward τ-shift* defined by $(\sigma^\tau w)(k) = w(\tau + k)$. The *restriction* of B to $I \subseteq \mathbb{Z}_+$, $B|_I$, is defined by $B|_I := \{w|_I \mid w \in B\}$. B is *time-invariant* if $\sigma^1 B \subseteq B$. B is *N-complete*, or equivalently, B has *memory span* N, if $N \in \mathbb{Z}_+$ and $B = \left\{ w \in W^{\mathbb{Z}_+} \,\middle|\, \forall_{\tau \in \mathbb{Z}_+} (\sigma^\tau w)|_{[0;N]} \in B|_{[0;N]} \right\}$. A superset B' of a behavior $B \subseteq W^{\mathbb{Z}_+}$ is called an *abstraction* of B, and B' is additionally called *discrete* if it can be realized by

a finite automaton. In particular, an abstraction that has finite memory span is discrete.

C. Discrete-time systems

In (1) with right hand side $G\colon X \times U \to X$, $u\colon \mathbb{Z}_+ \to U$ represents an input signal and $x\colon \mathbb{Z}_+ \to X$, an output signal. A *trajectory* of (1) is a sequence $(x,u)\colon \mathbb{Z}_+ \to X \times U$ for which (1) holds for all $k \in \mathbb{Z}_+$. The collection of such trajectories is called the *behavior* of (1).

D. Hypergraphs

We provide basic terminology from the theory of directed hypergraphs, see [5] and the references given there, where we specialize to what is usually called an F-hypergraph, and extend the terminology to allow for loops, parallel hyperedges, and labels.

A *(directed) hypergraph* is a triple (V,E,U) of a finite set V of *vertices*, a finite set U of *labels*, and a set $E \subseteq U \times V \times (\mathcal{P}(V) \setminus \{\emptyset\})$ of *hyperedges*. (V',E',U') is called the *sub-hypergraph* of (V,E,U) *induced* by the subset $E' \subseteq E$ of hyperedges, if $V' = \bigcup_{(u,v,W)\in E'} \{v\} \cup W$ and $U' = \{u \,|\, (u,v,W) \in E'\}$. (V',E',U') is called the *sub-hypergraph* of (V,E,U) *induced* by the subset $V' \subseteq V$ of vertices, if $E' = \{(u,v,W) \in E \,|\, \{v\} \cup W \subseteq V'\}$ and U' is as before.

Let (V,E,U) be a directed hypergraph. If $h = (u,v,W) \in E$, then v, u, and W are the *initial vertex*, the *label*, and the *head*, resp., of h, and elements of W are called *terminal vertices* of the hyperedge h. We also use the notation $T(h) := v$, $H(h) := W$, and $L(h) := u$. The *size* of (V,E,U) is defined to be $|E| + \sum_{e\in E} |H(e)|$.

In complete analogy to ordinary directed graphs, we call an alternating sequence $P = (v_1, e_1, \ldots, e_q, v_{q+1})$ of vertices v_i and hyperedges e_i a *path* from v_1 to v_{q+1} of *length* q if $v_1 = T(e_1)$, $v_{q+1} \in H(e_q)$, and $v_j = T(e_j) \in H(e_{j-1})$ for all $j \in \{2,\ldots,q+1\}$. If $S,T \subseteq V$, then $v \in V$ is *reachable* from S via T in (V,E,U) if there is a path from some $s \in S$ to v whose vertices are in T. If T is not specified, we assume $T = V$.

The concept of a hyperpath is more subtle: A subset $E' \subseteq E$ is an *F-path* from a vertex $s \in V$ to a set $Z \subseteq V$ of vertices, if the following two conditions hold: (i) There is an ordering $(e_1,\ldots,e_k)$ of the vertices $e_i \in E'$, $k = |E'|$, such that $s = T(e_1)$ and $H(e_i) \subseteq Z \cup \bigcup_{j=i+1}^{k} T(e_j)$. (ii) E' is minimal, i.e., no proper subset of E' satisfies condition (i). Note that by minimality, different hyperedges in an F-path cannot share a common initial vertex.

The *length* of an F-path E' is the maximum length of a path in the sub-hypergraph induced by E', and the *distance* from s to Z, denoted $d(s,Z)$, is the minimum length of an F-path from s to Z. If there is no such path, we set $d(s,Z) = \infty$.

III. Reduction of continuous attainability problems to distance problems in hypergraphs

The attainability problem we seek to solve for the continuous plant (1) is:

III.1 Problem. Let an initial set $X_0 \subseteq X$, a target set $X_1 \subseteq X$, and a set $X_2 \subseteq X$ of admissible states be given. Find a controller $R\colon X \to \mathcal{P}(U) \setminus \{\emptyset\}$ such that the condition

$$x_0 \in X_0 \Longrightarrow \exists_{k\in\mathbb{Z}_+} \left(x_k \in X_1 \text{ and } \forall_{t\in[1;k]} x_t \in X_2\right) \tag{9}$$

holds whenever $x\colon \mathbb{Z}_+ \to X$ is a sequence that fulfills

$$x(k+1) \in G(x(k), R(x(k))) \tag{10}$$

for all $k \in \mathbb{Z}_+$, where $G(x,R(x))$ denotes the set $\{G(x,u) \,|\, u \in R(x)\}$.

In other words, using a controller R, which may in general be non-deterministic, the state of the closed loop (10) should be steered into the target set within finite time, from everywhere in the initial set, while state constraints $x \in X_2$ should be satisfied on the way to the target.

A. Quantization

Our first step towards a solution of Problem III.1 through reduction to a finite problem is quantization of the continuous state space of (1) and formulation of a suitable auxiliary problem for the quantized system. To this end, we first choose an open neighborhood $W \subseteq X_2$ of $X_0 \cup X_1$. We next choose a finite covering C' of $X_0 \cup X_1$ by full-dimensional convex polyhedra ("operating range symbols") whose union is a subset of W, and finally supplement C' with additional full-dimensional convex polyhedra ("overflow symbols") that do not intersect $X_0 \cup X_1$, to obtain a covering C of $\mathbb{R}^n$. It follows from [1, Lemma IV.7, Theorem IV.8] that this construction is possible if X_0 and X_1 are compact and a neighborhood W as chosen above exists, which we assume henceforth. We now define the *quantizer* (8) by its system C of level sets ("cells"),

$$Q\colon \mathbb{R}^n \to \mathcal{P}(C)\colon x \mapsto \{\Omega \in C \,|\, x \in \Omega\}, \tag{11}$$

where $\mathcal{P}(C)$ denotes the power set of C. As C is a covering of $\mathbb{R}^n$, $Q(x) \neq \emptyset$ for every x. Note also that this quantizer is non-deterministic in general. The *behavior* B of the quantized system (1),(8) is the collection of sequences $(u,\Omega)\colon \mathbb{Z}_+ \to U \times C$ for which there exists a signal $x\colon \mathbb{Z}_+ \to \mathbb{R}^n$ such that (1) and (8) hold for all $k \in \mathbb{Z}_+$.

The following observation shows we can obtain a solution of Problem III.1 through solving a suitable attainability problem for the quantized system (1),(8).

III.2 Proposition. *Define*

$$Z_0 = \{\Omega \in C \,|\, \Omega \cap X_0 \setminus X_1 \neq \emptyset\}, \tag{12a}$$

$$Z_1 = \{\Omega \in C \,|\, \Omega \subseteq X_1\}, \tag{12b}$$

$$Z_2 = C', \tag{12c}$$

let $R'\colon C \to \mathcal{P}(U) \setminus \{\emptyset\}$*, and assume that the condition*

$$\Omega_0 \in Z_0 \Longrightarrow \exists_{k\in\mathbb{Z}_+} \left(\Omega_k \in Z_1 \text{ and } \forall_{t\in[1;k]} \Omega_t \in Z_2\right) \tag{13}$$

holds for any sequence $\Omega\colon \mathbb{Z}_+ \to C$ *for which*

$$x(k+1) \in G(x(k), R'(\Omega(k))), \tag{14a}$$

$$\Omega(k) \in Q(x(k)) \tag{14b}$$

is fulfilled for some $x\colon \mathbb{Z}_+ \to X$ and all $k \in \mathbb{Z}_+$. Further assume the controller $R\colon X \to \mathcal{P}(U)$ fulfills

$$\emptyset \neq R(x) \subseteq \bigcup_{x\in\Omega\in C} R'(\Omega)$$

for every $x \in X$, where the union is taken over all $\Omega \in C$ containing x. Then R solves Problem III.1. □

B. Abstraction

The purpose of quantizing the state space of the continuous plant (1) was to obtain a system whose input and output alphabets are finite. A difficulty is that, while (1) is 1-complete, the quantized plant (1),(8) normally is not N-complete for any N, which precludes a finite automata representation of the latter and makes it difficult to solve Problem III.1 directly with the help of (1),(8). We therefore intend to approximate the behavior B of (1),(8) by a superset – an abstraction – that does allow for a finite automaton realization. A natural choice for such a superset would be the smallest 1-complete superset of B, which is realizable and is called the 1-*complete hull of* B [1], [3] and is given by [6], [7]

$$\left\{(u,\Omega)\colon \mathbb{Z}_+ \to U \times C \,\middle|\, \forall_{k\in\mathbb{Z}_+} G(\Omega_k \cap X, u_k) \cap \Omega_{k+1} \neq \emptyset\right\}.$$

While the 1-complete hull can not be determined exactly in general, the point is that other 1-complete abstractions that conservatively and arbitrarily accurately approximate the smallest one can be obtained for a large class of nonlinear systems (1) by overapproximating the set $G(\Omega_k \cap X, u_k) \cap \Omega_{k+1}$. See [1] for an overview and for a recent and particularly efficient and accurate method.

Once a suitable abstraction has been determined, we obtain a solution to Problem III.1 from any solution to an auxiliary problem for the abstraction, e.g. [7].

III.3 Lemma. *Let B' be a 1-complete abstraction of B, let Z_0, Z_1 and Z_2 be defined by (12), and let $R'\colon C \to \mathcal{P}(U)\setminus\{\emptyset\}$ be such that (13) holds whenever $(u,\Omega) \in B'$ and $u_k = R'(\Omega_k)$ for all $k \in \mathbb{Z}_+$. Then R' fulfills the assumptions in Prop. III.2.*

C. Reduction to distance problems in hypergraphs

We next give a realization of 1-complete abstractions of B in the language of hypergraphs, which should be easy to be reinterpreted in terms of automata. We prefer hypergraphs to automata as we aim at the application of efficient algorithms from hypergraph theory.

III.4 Definition. *The hypergraph (C, E, U) realizes a 1-complete abstraction of B if*

$$E = \{(u,\Omega,h(u,\Omega)) \,|\, u \in U, \Omega \in C\}, \tag{15a}$$

$$C \supseteq h(u,\Omega) \supseteq \{\Omega' \in C \,|\, G(\Omega\cap X, u)\cap\Omega' \neq \emptyset\}. \tag{15b}$$

We finally show that the solution of a distance problem in a hypergraph that realizes an abstraction of B provides a solution to the continuous attainability problem III.1. The following theorem is obtained by extending a result on prestabilizability from [8] to allow for the state constraint $x \in X_2$, and combining it with Proposition III.2 and Lemma III.3 as well as with results that guarantee the existence of a suitable quantizer [1, Lemma IV.7, Theorem IV.8].

III.5 Theorem. *Let X_0 and X_1 be compact, and let there exist an open neighborhood $W \subseteq X_2$ of $X_0 \cup X_1$. Define the quantizer Q and the covering C as in section III-A, let Z_0, Z_1 and Z_2 be defined by (12), and let B be the behavior of the quantized system (1),(8). Let further be H a hypergraph that realizes a 1-complete abstraction of B, and H', the sub-hypergraph of H induced by Z_2, and let d be the distance function in H'.*

Then the continuous attainability problem III.1 has a solution if $d(\Omega, Z_1) < \infty$ for all $\Omega \in Z_0$.

It follows that Problem III.1 can be efficiently solved using well-known algorithms for distance problems in hypergraphs [9]. In fact, these algorithms are easily supplemented with just one additional command in order to obtain a controller in addition to the distance function and may be implemented to run in time linear in the size of H [9].

IV. An efficient method for decomposable plants

In this section, we will exploit the structure (5) of the right hand side G of the system (1) in order to improve the efficiency of the general method from section III.

The general procedure of choosing a quantizer (11) will be the same as in section III. However, in order to exploit the product structure of the plant (1), we will need to choose a quantizer that is compatible with the structure of (1). This will allow us to consider completely decoupled quantized versions of the sub-plants (5). As in section III, we first choose an open neighborhood $W \subseteq X_2$ of $X_0 \cup X_1$. We next fix p full-dimensional, compact, convex polyhedra $Y_i \subseteq \mathbb{R}^{n_i}$ in the state spaces of the p sub-plants (5) and set $Y := Y_1 \times \cdots \times Y_p$. In addition to the requirements imposed on C' in section III, we here additionally restrict the cells in C' to be scaled and translated copies of Y, and the overflow symbols in $C \setminus C'$, to possess a product structure. The following result shows that such a construction is always feasible. The point here is that by compactness of X_0 and X_1 and by openness of W one can choose overflow symbols that do not intersect $X_0 \cup X_1$. See also Fig. 2.

IV.1 Proposition. *Let X_0 and X_1 be compact, Y and W be as above, and let $\hat{\lambda} > 0$. Then there exists λ, $0 < \lambda < \hat{\lambda}$, and a finite covering C' of $X_0 \cup X_1$ whose elements are translated copies of λY whose union is contained in W, and C' can be supplemented to a finite cover C of $\mathbb{R}^n$ whose elements are full-dimensional convex polyhedra. Moreover, C and C' can be chosen such that*

$$C = \{\Omega_1 \times \cdots \times \Omega_p \,|\, \Omega_1 \in C_1, \ldots, \Omega_p \in C_p\} \tag{16}$$

and $\Omega \cap (X_0 \cup X_1) = \emptyset$ whenever $\Omega \in C \setminus C'$.

We now define quantizers Q_i for the sub-plants (5) by

$$Q_i(x_i) = \{\Omega_i \in C_i \,|\, x_i \in \Omega_i\}, \tag{17}$$

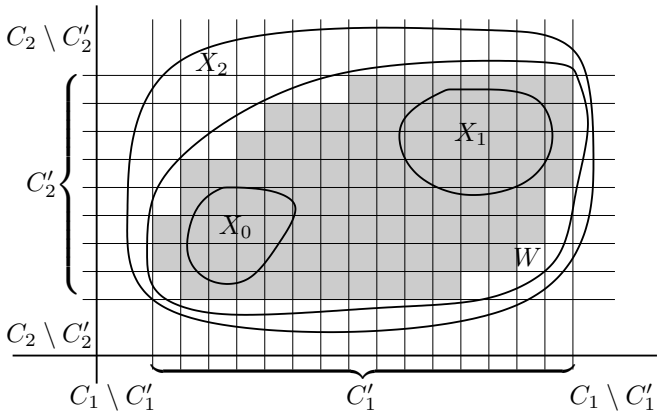

Figure 2. Illustration of the covering constructed in Proposition IV.1 in the case $p = 2$. Shaded cells are operation range symbols in C'.

and the quantizer (11) of the overall plant (1), by

$$Q(x) = (Q_i(x_i), \ldots, Q_p(x_p)).$$

Once that kind of quantizer has been designed, the following result shows that the product of hypergraphs realizing abstractions of B_i realizes an abstraction of the overall plant.

IV.2 Lemma. *For each $i \in \{1, \ldots, p\}$ let the hypergraph $H_i = (C_i, E_i, U)$ realize an 1-complete abstraction of the behavior B_i of the ith quantized sub-plant. Then the product hypergraph $H = (C, E, U)$ realizes an 1-complete abstraction of the behavior B of the overall plant, where E is the set of all $(u, \Omega_1 \times \cdots \times \Omega_p, \{\Omega_1' \times \cdots \times \Omega_p' \,|\, \Omega_1' \in W_1, \ldots, \Omega_p' \in W_p\})$ for which $(u, \Omega_i, W_i) \in E_i$ for all $i \in \{1, \ldots, p\}$.*

It follows that the abstractions of the sub-plants may be determined independently and concurrently. Since the dimension of the overall state space is the product of the dimensions of the state spaces of the sub-plants, a direct determination of an abstraction of the overall plant would typically be slower by several orders of magnitude. In order to devise a method that efficiently solves the continuous attainability problem III.1 for decomposable plants, we now present an algorithms for computing distances in hypergraphs which works directly with the factor abstractions.

IV.3 Theorem. *Let X_0 and X_1 be compact, and let there exist an open neighborhood $W \subseteq X_2$ of $X_0 \cup X_1$. Define the quantizer Q and the covering C as in section IV above, let Z_0, Z_1 and Z_2 be defined by (12), and let B_i denote the behavior of the quantized sub-system (5),(17). Let further be H_i a hypergraph that realizes a 1-complete abstraction of B_i, and H, the product of the H_i as defined in Lemma IV.2, and H', the sub-hypergraph of H induced by Z_2, and let d be the distance function in H'. Then the following holds.*

(i) The distance function $d(\cdot, Z_1)$ is given by the output V of the algorithm in Fig. 3.

(ii) If $d(\Omega, Z_1) < \infty$ for all $\Omega \in Z_0$, then the output R' satisfies the hypotheses of Lemma III.3, and hence, a solution R to the continuous attainability problem III.1 is obtained as in Lemma III.3.

(iii) The running time of the algorithm in Fig. 3 is linear in the size of H.

Input: p, $H_i = (C_i, E_i, U)$ $(i \in \{1, \ldots, p\})$, Z_0, Z_1, Z_2

1: **for all** $i \in \{1, \ldots, p\}$ **do**
2: **for all** $j \in \{1, 2, 3\}$ **do**
3: $Z_{j,i} := \{\Omega_i \,|\, \Omega_1 \times \cdots \times \Omega_p \in Z_j\}$
4: **end for**
5: $(Z_{2,i}, E_i', U_i') :=$ sub-hypergraph of H_i induced by $Z_{2,i}$
6: Compute distances $d_i(\cdot, Z_{1,i})$ in $(Z_{2,i}, E_i', U_i')$
7: $T_i := \{\Omega \in Z_{2,i} \,|\, d_i(\Omega, Z_{1,i}) < \infty\}$
8: $A_i := \{\Omega \in Z_{2,i} \,|\, \Omega$ reachable from $Z_{0,i}$ via T_i in $(Z_{2,i}, E_i', U_i')\}$
9: $(A_i, E_i'', U_i'') :=$ sub-hypergraph of $(Z_{2,i}, E_i', U_i')$ induced by A_i
10: **end for**
11: $Q := Z_1$; $R' :=$ arbitr.; $V(\Omega) := \begin{cases} 0, & \text{if } \Omega \in Z_1, \\ \infty, & \text{if } \Omega \in Z_2 \setminus Z_1 \end{cases}$
12: **while** $Q \neq \emptyset$ **do**
13: pick $\Omega \in \operatorname{argmin} \{V(\Omega') \,|\, \Omega' \in Q\}$
14: $Q := Q \setminus \{\Omega\}$
15: **for all** $e_1 = (u, \Omega_1', W_1) \in E_1''$ with $\Omega_1 \in W_1$ **do**
16: ...
17: **for all** $e_p = (u, \Omega_p', W_p) \in E_p''$ with $\Omega_p \in W_p$ **do**
18: $W = \left\{\widetilde{\Omega}_1 \times \cdots \times \widetilde{\Omega}_p \,\middle|\, \widetilde{\Omega}_1 \in W_1, \ldots, \widetilde{\Omega}_p \in W_p\right\}$
19: $\Omega' = \Omega_1' \times \cdots \times \Omega_p'$
20: **if** $\Omega' \in Z_2$, $W \cap Q = \emptyset$ and $V(\Omega') > 1 + V(\Omega)$ **then**
21: $R'(\Omega') := u$
22: $V(\Omega') := 1 + V(\Omega)$
23: $Q := Q \cup \{\Omega'\}$
24: **end if**
25: **end for**
26: ...
27: **end for**
28: **end while**

Output: V, R'

Figure 3. Algorithm that determines a controller R' for a decomposable plant by computing distances to Z_1 in a sub-hypergraph of H and working directly with the factors H_i.

The algorithm in Fig. 3 is an extension of the one in [9], which works directly with the factor abstractions. First of all, there is a preprocessing step, consisting of lines 1-10, whose execution is not necessary for the correct behavior of the algorithm, nor does it change the asymptotic worst-case complexity of the latter. However, that step implements a heuristic which, in practice, may considerably reduce the size of the distance problem to be solved by the main loop of the algorithm; see section V. Essentially, for each of the factors H_i, overflow symbols are removed, an auxiliary distance problem is solved (line 6), and based on the solution of the latter, each factor is shrunk again without affecting the distance function and the controller to be computed by the main loop. In the main loop (lines 12-28), the for-loop over

the hyperedges in H' is represented by p nested for-loops, which corresponds to the necessity of directly walking through the sets of hyperedges of the factor hypergraphs. In addition, the condition $\Omega' \in Z_2$, whose computational verification is trivial, had to be included on line 20 since Z_2 does not necessarily decompose and the sub-hypergraph H' is never explicitly computed.

V. Example

In this section, we demonstrate the application of our method to the simplest case of the pendulum example from Section I, namely, to the problem of swinging up a single pendulum mounted on a cart, in which case the overall plant is 4-dimensional and decomposes in $p = 2$ sub-plants. Specifically, one copy of (6) describes the motion of the pole, and (7), of the cart, and the overall system (6),(7) with parameters $\omega_1 = 1$ and $\gamma_1 = 0.0125$ models a real experimental setup.

The low-level controller in Fig. 1 is defined by the affine feedback $u = K(x - (\pi, 0, 0, 0))$, $K = (-3, -2.8, 0.15, 1)$, which stabilizes the system at $(\pi, 0, 0, 0)$. Using SOS-programming techniques [10], [11] we have verified that the ellipsoid

$$\Gamma = \left\{ x \in \mathbb{R}^4 \,\middle|\, \langle x|Px\rangle \leq 1 \right\}$$

is a positive invariant subset of the stability region, where P is the symmetric matrix given by

$$P = \begin{pmatrix} 8.674 & 8.297 & -1.25 & -3.809 \\ \cdot & 8.443 & -1.236 & -3.823 \\ \cdot & \cdot & 0.375 & 0.657 \\ \cdot & \cdot & \cdot & 2.303 \end{pmatrix}.$$

The set U of admissible control symbols consists of 11 signals, which are constant on their intervals of definition. Specifically, 7 of those signals are defined on $[0, 1/3]$ with values 0, ± 1.2, ± 0.8, and ± 0.4, and the remaining 4 are defined on $[0, 2/9]$ with values ± 0.9 and ± 0.45. From the continuous-time model (6)-(7) we obtain a discrete-time plant (1) by non-uniform sampling. More precisely, the right hand side G of (1) is defined by $G(x_0, u) = \varphi(T(u), x_0, u)$, where $u \in U$, $\varphi(t, x_0, u)$ is the solution at time t of the initial value problem composed of the ODE (6)-(7) and the initial condition $x(0) = x_0$, and the sampling time $T(u) \in \{1/3, 2/9\}$ is the length of the interval of definition of u. The discrete-time plant (1) is subject to the constraints

$$|x_3|, |x_4| \leq 2.4, \quad |x_2| \leq \pi. \tag{18}$$

Here, we focus on the design of a supervisor that steers the state into the stability region. Specifically, we set $X_2 := \mathbb{R} \times [-\pi, \pi] \times [-2.4, 2.4] \times [-2.4, 2.4]$, which reflects the constraints (18), $X_0 = [-0.05, 0.05]^4$, and $X_1 = \Gamma$. For each of the two sub-plants, we use translated copies of a rectangle, intersected with $[0, 2\pi] \times [-\pi, \pi]$ and $[-2.4, 2.4] \times [-2.4, 2.4]$, respectively, as operating range cells. The coverings C_1' and C_2' thus obtained are then supplemented by the two overflow symbols $\mathbb{R} \times [\pi, \infty[$ and $\mathbb{R} \times]-\infty, -\pi]$ in the case of the

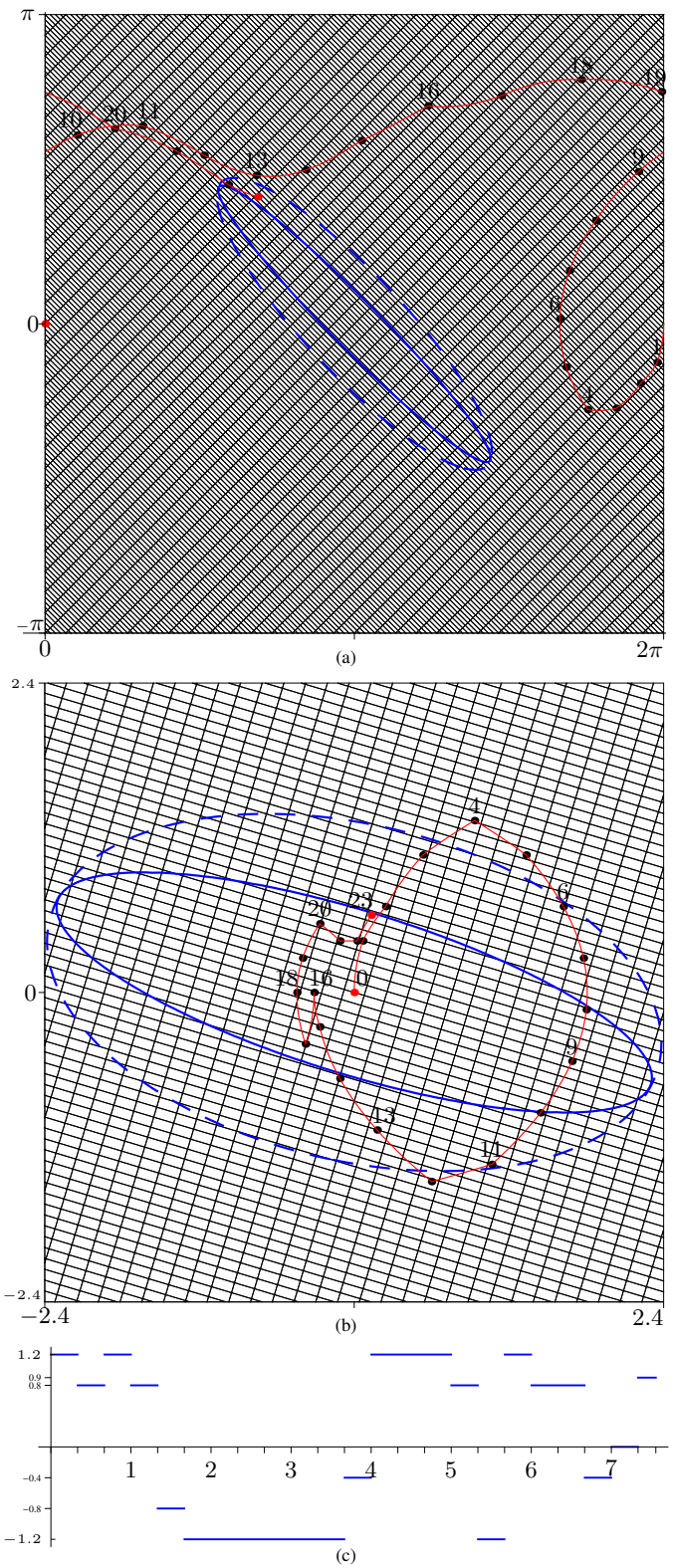

Figure 4. State space of pole (a) and cart (b). The larger (resp., smaller) ellipsoid is the projection of the stability region Γ onto (resp., the intersection of Γ with) the state space of the sub-plant. Γ accounts for less than 1% of the volume of the state space; see Tab. I. The supervisor forces the state into Γ within at most 33 steps, from everywhere in a neighborhood of the origin. One particular trajectory, which starts precisely at the origin, is illustrated in (a), (b), and the corresponding control signal is shown in (c).

Table I
COMPUTATIONAL RESULTS FOR THE EXAMPLE IN SECTION V.

state space X_2':	$[0,2\pi]\times[-\pi,\pi]\times[-2.4,2.4]\times[-2.4,2.4]$		
target region X_1:	Γ	$\Gamma/\sqrt{2}$	$\Gamma/2$
$\mathrm{vol}(X_1)/\mathrm{vol}(X_2')$ in %:	0.819	0.205	0.0512
pole:			
CPU (abstraction)	559	1020	2320
vertices/10^3	7.44	13.1	29.3
size/10^3	452	795	1770
vertices (shrunk)/10^3	7.14	12.6	28.3
size (shrunk)/10^3	440	776	1730
cart:			
CPU (abstraction)	239	465	1020
vertices/10^3	1.87	3.78	7.29
size/10^3	88.2	182	356
vertices (shrunk)/10^3	1.21	2.46	4.73
size (shrunk)/10^3	58.7	122	236
cart-pole system:			
vertices/10^6	13.9	49.6	213
size/10^9	2.54	9.21	40.0
CPU (shrink)	0.466	1.86	8.09
vertices (shrunk)/10^6	8.63	31.1	134
size (shrunk)/10^9	1.65	6.01	26.0
CPU (supervisor)	50.0	197	956

pole, and by the four overflow symbols $\pm[2.4,\infty[\times\mathbb{R}$, $\pm(\mathbb{R}\times[2.4,\infty[)$ in the case of the cart. See Fig. 4. We implicitly consider (6) on the cylinder [12], i.e., on $X_2' := [0,2\pi]\times[-\pi,\pi]\times[-2.4,2.4]\times[-2.4,2.4]$ with x and $x+(2\pi,0,0,0)$ identified. We also use analogous, but finer, state space quantizations to steer the state into two shrunk copies of Γ; see Tab. I.

Our computational experiments have been preformed on an i7-920 CPU at $2.67GHz$ using 4 GBytes of RAM. Abstractions of the sub-plants have been determined using an implementation of the method from [1] in *Mathematica 7.0* [13], which is an interpreter, run on 6 CPU threads for the pole, and 2, for the cart system. The algorithm in Fig. 3 has been implemented in C and run on the same computer using a single thread. Computational results are collected in Tab. I, where we report the number of vertices and the size of the factor hypergraphs $(Z_{2,i}, E_i', U_i')$ and their shrunk variants (A_i, E_i'', U_i''), which are computed by the algorithm in Fig. 3. Properties of the product of those factor hypergraphs are also reported ("cart-pole system"), together with wall clock times in seconds for computing the abstractions of the factors, for shrinking all the factor hypergraphs, and for computing a supervisor. All quantities are rounded to three decimal digits. Note that the abstractions of the two factors can be computed concurrently. Hence, despite their complexity, the three instances of the continuous attainability problem III.1 considered here can be solved within less than 11, 21 and 55, respectively, minutes. We emphasize that these computations are to be performed off-line. They yield a feedback controller in the form of a look-up table, suitable for real-time application in the closed loop of Fig. 1(b).

From the data in Tab. I it is also seen that the computational effort to determine the abstractions of the factors grows linearly with the number of cells used for quantizing the state space, and the effort to determine the supervisor grows linearly with the size of the hypergraph realizing an abstraction of the overall plant. Finally, the heuristics for shrinking the factor hypergraphs proves very effective, and the computational effort required by it is negligible.

VI. CONCLUSION

We have proposed a method that uses finite state models for solving continuous attainability problems for nonlinear decomposable plants. Specifically, the plants consist of sub-plants that may share a common input but are otherwise coupled only through the specification of the attainability problem, possibly including state constraints. Our method only requires computation of abstractions of the sub-plants rather than of the overall plant. The explicit representation of an abstraction of the latter is also avoided in the solution of the auxiliary discrete attainability problems that arise. This way, we are able to solve highly complex, previously intractable continuous attainability problems within minutes on a customary personal computer. In the special case of reachability problems, these off-line computations yield a feedback controller in the form of a look-up table, suitable for real-time application.

The extension of our method to cover abstractions of finite but otherwise arbitrary memory span and to solve optimal control variants of attainability problems, and its combination with local refinement strategies [14], [15] and with hierarchical approaches [16] will be the subject of future research.

REFERENCES

[1] G. Reißig, "Computing abstractions of nonlinear systems," 2009, submitted for publication, Sep. 29, 2009, http://arxiv.org/abs/0910.2187.

[2] P. Tabuada, *Verification and control of hybrid systems*. Springer, 2009.

[3] G. Reißig, "Computation of discrete abstractions of arbitrary memory span for nonlinear sampled systems," in *HSCC'2010, San Francisco, U.S.A., Apr. 13-15, 2009*, Lect. Notes Computer Science, vol. 5469, pp. 306–320, http://www.reiszig.de/gunther/pubs/i09HSCC.abs.html.

[4] J. C. Willems, "Models for dynamics," in *Dynamics reported, Vol. 2*, Ser. Dynam. Systems Appl. Wiley, 1989, vol. 2, pp. 171–269.

[5] M. Thakur and R. Tripathi, "Linear connectivity problems in directed hypergraphs," *Theoret. Comput. Sci.*, vol. 410, no. 27-29, pp. 2592–2618, 2009.

[6] T. Moor and J. Raisch, "Supervisory control of hybrid systems within a behavioural framework," *Systems Control Lett.*, vol. 38, no. 3, pp. 157–166, 1999, hybrid control systems.

[7] T. Moor, J. M. Davoren, and B. D. O. Anderson, "Robust hybrid control from a behavioural perspective," in *Proc. CDC 2002*, pp. 1169–1174.

[8] C. M. Özveren, A. S. Willsky, and P. J. Antsaklis, "Stability and stabilizability of discrete event dynamic systems," *J. Assoc. Comput. Mach.*, vol. 38, no. 3, pp. 730–752, 1991.

[9] G. Gallo, G. Longo, S. Pallottino, and S. Nguyen, "Directed hypergraphs and applications," *Discrete Appl. Math.* 42, no. 2-3, pp. 177–201, 1993.

[10] D. Henrion and A. Garulli, Eds., *Positive polynomials in control*, Lecture N. in Control and Information Sci. Springer, 2005, vol. 312.

[11] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. CACSD 2004, Taipei, Taiwan, Sep. 2-4*, 2004.

[12] E. D. Sontag, *Mathematical control theory*, 2nd ed., Springer, 1998.

[13] S. Wolfram, *The Mathematica® book*, 5th ed. Wolfram Media, 2003.

[14] E. Clarke, A. Fehnker, Z. Han, B. Krogh, J. Ouaknine, O. Stursberg, and M. Theobald, "Abstraction and counterexample-guided refinement in model checking of hybrid systems," *Internat. J. Found. Comput. Sci.*, vol. 14, no. 4, pp. 583–604, 2003.

[15] O. Stursberg, "Supervisory control of hybrid systems based on model abstraction and guided search," *Nonl. Anal.* 65, pp. 1168–1187, 2006.

[16] K. Schmidt, T. Moor, and S. Perk, "Nonblocking hierarchical control of decentralized discrete event systems," *IEEE Trans. Automat. Control*, vol. 53, no. 10, pp. 2252–2265, 2008.