Closed-loop incremental stability for efficient symbolic control of non-linear systems

Pouria Tajvar, Pierre-Jean Meyer, Jana Tumova

Closed-loop incremental stability for efficient symbolic control of non-linear systems

Pouria Tajvar * **Pierre-Jean Meyer** ** **Jana Tumova** *

* *KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: {tajvar,tumova}@kth.se).*
** *COSYS-ESTAS, Univ Gustave Eiffel, Univ Lille, F-59666 Villeneuve d'Ascq, France (e-mail: pierre-jean.meyer@univ-eiffel.fr)*

Abstract: In this paper, we introduce a hybridization-based feedback control synthesis method for potentially unstable nonlinear continuous-time systems. We construct a discretization and feedback control that ensures dissipation form of incremental input-to-state stability property that a number of abstraction-based control methods rely on or benefit from. This enables the use of these methods also for the case of potentially unstable systems, to achieve a reachability or a temporal logic specification. We furthermore show that the algorithm can also improve abstraction-based methods that do not rely on stability assumptions by reducing the number of constructed abstract states and non-deterministic transitions. We illustrate the benefits of our approach in simulations featuring a cart-pendulum.

Keywords: Stability and stabilization of hybrid systems, Reachability analysis

1. INTRODUCTION

The main idea of abstraction-based control synthesis tools, such as CoSyMa, SCOTS, or ROCS, is to break down the state space of a dynamical system into sub-regions, seen as abstract states, and capture the dynamics as transitions between these abstract states (Mouelhi et al., 2013; Rungger and Zamani, 2016; Li and Liu, 2018). Controllers can then be synthesized as maps from the sub-regions to control inputs. Abstraction-based methods are motivated by three main reasons. First, different classes of dynamics can be represented in the form of a generic state-transition system as shown by Reißig (2011). In addition, they enable consideration of hybrid systems that include both continuous dynamics and discrete switches (Tabuada, 2006). They also allow control synthesis for specifications beyond stability or reachability, for example by integration of various forms of temporal logic (Liu and Ozay, 2014; Yordanov et al., 2011).

However, abstraction-based control synthesis can become computationally prohibitive in complex systems. Transitions between abstract states may be non-deterministic since each abstract state corresponds to an infinite set of states in the actual system. As a result, a choice of very fine sub-regions can be required, resulting in many abstract states. Furthermore, to ensure a correct correspondence between the synthesized controller for the abstraction and its implementation on the original system, various abstraction methods rely on imposing certain assumptions on the dynamics that allow for deriving formal relations between the abstraction and the system; for example, simulation relation (Girard and Pappas, 2007) or refinement relation (Rungger and Zamani, 2016). An abstraction refinement procedure is proposed by Li and Liu (2018) that allows for less restrictive assumptions on the dynamics by an adaptive iterative refinement of the sub-regions. While an adaptive refinement policy can be computationally beneficial compared to static fine abstractions, it can still result in many abstract states, depending on the dynamics.

In this work, we address the problem of abstraction-based control synthesis for a non-linear system with dynamics uncertainty. We propose a local multi-step state-feedback synthesis approach to reduce the non-deterministic transitions and the need for abstraction refinement to ultimately move toward more scalable abstraction-based control synthesis. We show that unstable non-linear systems can be transformed to incrementally stable systems in regions with controllable local linearizations; such regions commonly span most of the state-space in physical systems. For example, in a cart pendulum system, local linearizations are controllable everywhere except around the horizontal positions of the pendulum. We demonstrate in simulations that even in the other regions, the proposed multi-step feedback synthesis approach can allow for fewer abstract states compared to a purely symbolic approach with fixed discretized control inputs. Furthermore, we discuss how such multi-step state-feedback can be used to speed up abstraction-based methods in general.

Contribution The contributions of this paper can be summarized as follows: We first provide a method that utilizes established works in reachable set computation and affine dynamics approximation to provide the time discretization of the system (Section 3). Second, in our main contribution, we design multi-step feedback control synthesis algorithm to ensure incremental input-to-state stability for the closed loop dynamics; the stable system can then be used for symbolic control (Section 4). Finally,

we demonstrate the benefits of using our approach via a simulation on a cart-pendulum system example (Section 5).

1.1 Related work

The idea of using feedback control in order to reduce the number of transitions in an abstraction has also been explored by Sinyakov and Girard (2020), specifically for monotone systems. In contrast to this work, we aim at decoupling the state-feedback policy from the abstraction construction to allow integration with various existing abstraction tools, including tools that do not assume system monotonicity.

The main challenge of feedback synthesis for non-linear non-monotone systems is that even in case where the system has controllable Lie groups, the linearization of the continuous-time system might not be controllable and the discretization of the system might not be controllable in one time step. Some works thus proposed multi-step state-feedback control synthesis for discretizations of non-linear systems (Schürmann and Althoff, 2017; Tajvar et al., 2019). The main contrast in this paper is that instead of dynamic linearization around pre-defined system trajectories, we adopt a hybridization approach to construct local discrete-time models that allow computation of state-feedback policies. While the proposed approach requires an additional hybridization step, it can be used to construct abstraction that can be reused for different specifications.

1.2 Motivating example

Consider a simple double-integrator system $f : \ddot{x} = u$. Let us assume that we control this system with a 1 second sampling time and bounded input $u \in [-1, 1]$.

This system is linear but unstable, and as a result, the reachable set from any initial set in the state space grows over time. Fig. 1.(a) shows the reachable set in 2 seconds from set $[-0.5, 0.5] \times [-0.5, 0.5]$ in light green for the fixed input $u = 0.5$. It can be seen that the reachable set spans three cells in the state space. This would result in a nondeterministic transition in the corresponding abstraction. However, the reachable set can be smaller if state-feedback is used, as illustrated in dark green for 2-step state-feedback constructed via our proposed method. State-feedback policies can be derived globally for linear systems, but in general not for non-linear systems. We suggest an approach to construct hybridizations of non-linear systems for computation of local state-feedback policies. Namely, we divide the state space into finite set of smaller regions, and construct a local state-feedback for each of them.

The expansion of the reachable set in general is one of the major bottlenecks in the application of abstraction-based methods. It can in general result in infeasibility of tools that use fixed abstractions, such as SCOTS (Rungger and Zamani, 2016), for unstable systems. In tools such as ROCS (Li and Liu, 2018) where adaptive abstraction refinement is employed for computing backward reachable sets, the expansion of the reachable set will require the starting set being smaller than the target set in backward reachability computation. For instance, let us look at the

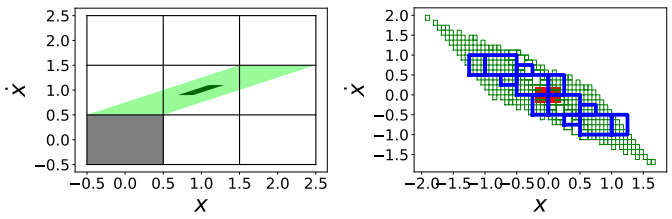

Fig. 1. (a) Reachable set contraction with feedback. (b) Backward reachable set using feedback.

backward reachable set of the target set $[-0.2, 0.2] \times [-0.2, 0.2]$ (in red) in 2 seconds with bounded input $[-1.0, 1.0]$ in Fig. 1.(b). The green (thinly outlined) cells are computed following the abstraction refinement method used by Li and Liu (2018) without state-feedback, while the blue (thickly outlined) cells are computed using the same abstraction refinement approach augmented with state-feedback proposed in this paper. It can be seen that using feedback allows finding backward reachable sets that can be larger than the target set, which is desired for computational efficiency.

For the local state-feedback computation, we propose synthesizing k-step discrete-time control to keep the reachable set size within pre-defined bounds. This is not necessarily possible in $k = 1$ step depending on input degrees of freedom, but it is possible in finite time steps as long as the system is locally controllable. We will later show that even for uncontrollable linearized dynamics, the synthesized controller can reduce non-deterministic transitions by shrinking the reachable sets. This approach allows direct incorporation of input bound constraints and does not require design of a convergence assessment function as for example control Lyapunov functions. Using our approach, a part of the input space becomes unavailable for symbolic control as it is reserved for feedback. For instance, in Fig. 1 we observe that the blue cells do not entirely cover the area that is covered by the green cells; this is because we cannot use the entire input range as a result of allocating a part of it to the state feedback. Hence, there is a trade-off between computational efficiency determined by the size of the reachable sets, and conservativeness influenced by not being able to use a part of the input space for symbolic control. We discuss this trade-off in Section 4.

1.3 Notation

Let us denote the set of real numbers by $\mathbb{R}$, an n-dimensional vector space by $\mathbb{R}^n$, and an $m \times n$ matrix space by $\mathbb{R}^{m \times n}$. We use p_i to refer to the i-th element of a vector $p \in \mathbb{R}^n$.

The Minkowski sum $A \oplus B$ of two sets $A \subset \mathbb{R}^n$ and $B \subset \mathbb{R}^n$ refers to the following set:

$$A \oplus B = \{a + b \mid a \in A, b \in B\}. \tag{1}$$

A *zonotope* $Z(\mu, \mathbf{G}) \subset \mathbb{R}^n$, where $\mu \in \mathbb{R}^n$ and $\mathbf{G} \in \mathbb{R}^{m \times n}$, refers to the centrally symmetric set

$$Z(\mu, \mathbf{G}) = \{x \in \mathcal{X} \mid x = \mu + G\omega,\ \omega \in [-1, 1]^n\}, \tag{2}$$

where $[-1, 1]^n \subset \mathbb{R}^n$ represents the Cartesian product of closed $[-1, 1]$ intervals. We refer to μ as the center of the zonotope and G as the set of generators. $Z.\mu$ and $Z.G$

are used to refer to the corresponding parameters of the zonotope Z.

A *box* $B(\mu, \mathbf{G})$ is a special case of a zonotope where all of the off-diagonal elements of $\mathbf{G}$ are zero. The size of a box $|B|$ is the vector of its interval lengths in each dimension. We use the box operator $\square Z$ to denote the tight box over-approximation of the zonotope Z.

The following zonotope operations are used in this paper:

$$\begin{aligned}
&\mathbf{A} \times Z, \text{ where} && (\mathbf{A} \times Z).\mu = \mathbf{A}Z.\mu,\\
& && (\mathbf{A} \times Z).\mathbf{G} = \mathbf{A}Z.\mathbf{G};\\
&Z + Z', \text{ where} && (Z + Z').\mu = Z.\mu + Z'.\mu,\\
& && (Z + Z').\mathbf{G} = Z\mathbf{G} + Z'\mathbf{G};\\
&Z \oplus Z', \text{ where} && (Z \oplus Z').\mu = Z.\mu + Z'.\mu,\\
& && (Z \oplus Z').\mathbf{G} = [Z.\mathbf{G}|Z'.\mathbf{G}],
\end{aligned}$$

where $[Z.\mathbf{G}|Z'.\mathbf{G}]$ denotes the concatenation of the two matrices along their columns.

We denote the farthest distance between two sets S and S' as $\Delta(S, S') = \sup_{s \in S} \sup_{s' \in S'} ||s - s'||_\infty$.

A function $\beta : [0, \infty) \to [0, \infty)$ is said to be class $\mathcal{K}_\infty$ if it is strictly increasing, $\beta(0) = 0$, and $\lim_{v \to \infty} \beta(v) = \infty$.

2. PROBLEM FORMULATION

Let us consider continuous-time system Σ with dynamics that is affine in relation to disturbance $w(t)$:

$$\Sigma : \quad \dot{x}(t) = f(x(t), u(t)) + w(t), \tag{3}$$

where $x \in \mathcal{X} \subset \mathbb{R}^n$, $u \in \mathcal{U} \subset \mathbb{R}^m$, and $w \in \mathcal{W} \subset \mathbb{R}^n$ are the system state, input and disturbance respectively. We use $\mathbf{x} : \mathbb{R} \to \mathcal{X}$, $\mathbf{u} : \mathbb{R} \to \mathcal{U}$, and $\mathbf{w} : \mathbb{R} \to \mathcal{W}$, to denote signals in the corresponding spaces.

Regarding the continuity of the dynamics, let us make following assumption:

Assumption 1. The function f is twice differentiable and its Hessian is element-wise bounded:

$$\max_{l, l', z \in \mathcal{X} \times \mathcal{U}} \left| \frac{\partial^2 f}{\partial z_l \partial z_{l'}}(z) \right| \leq \overline{h}. \tag{4}$$

Problem 1. Given a continuous-time system Σ, an initial set $X_0 \subset \mathcal{X}$, and a sample interval T, find a control policy $\nu(x) : \mathcal{X} \to \mathcal{U}$ such that the closed-loop time-sampled system $F^{cl}(x, u^*) = F(x, u^* + \nu(x))$ satisfies:

$$\begin{aligned}
&\forall x, x' \in X_0 \wedge u^* \in \mathcal{U}^*,\\
&\Delta(F^{cl}(x, u^*), F^{cl}(x', u^*)) - ||x - x'||_\infty\\
&\qquad \leq -\beta(||x - x'||_\infty) + \omega,
\end{aligned} \tag{5}$$

where β is a class $\mathcal{K}_\infty$ function, $\omega \in \mathbb{R}$ and $\mathcal{U}^* = \{u^* \mid \forall x \in X_0, u^* + \nu(x) \in \mathcal{U}\}$ is the set of inputs that will remain in $\mathcal{U}$ after the application of the state-feedback control.

Equation (5) is a special case of the dissipation-form incremental input-to-state stability, i.e. Definition 7 in Tran et al. (2016), where energy function $V(.) = ||.||_\infty$ and the discrete time function is an implicit map function of the disturbance. $F^{cl}(x, u^*)$ will be then the discrete-time stabilized dynamics that can be used for symbolic control synthesis towards, for example, a reachablity or temporal logic specification.

3. DISCRETE-TIME HYBRIDIZATION

We recall that $F(x, u)$ is the reachable set of system (3) from state $x \in \mathcal{X}$ and under input $u \in \mathcal{U}$ that cannot, in general, be computed precisely for a non-linear system. Hence, in this section, we focus on computing $\overline{\Sigma}^k : \mathcal{X} \times \mathcal{U} \rightrightarrows \mathbb{R}^n$, that is a k-step over-approximation of the time-discretization F of system (3). Furthermore, we require $\overline{\Sigma}^k$ to be piecewise-affine for the application of our local state-feedback synthesis in Section 4. In $\overline{\Sigma}^k$, the state-input space $\mathcal{X} \times \mathcal{U}$ is decomposed into P partitions, $\mathcal{X} \times \mathcal{U} = \bigcup_{p=1}^{P} X_p \times U_p$. For each partition $p \in P$, $\overline{F}_p$ is an affine map and is computed so that $\forall (x, u) \in X_p \times U_p$, $F(x, u) \subseteq \overline{F}_p(x, u) = \{\mathbf{A}_p x + \mathbf{B}_p u + \mathbf{c}_p\} \oplus W_p$.

The problem of reachable set construction has been previously addressed for affine systems by Girard et al. (2006), who introduced a highly efficient procedure to compute approximations of reachable sets. The key idea therein is to use zonotopes for the representation of the approximations. Since zonotopes are closed under linear transformations, the wrapping effect (propagation of error) is avoided, and the resulting approximations are tight.

While some approaches can be used to compute over-approximations of the reachable set for nonlinear systems (Althoff and Krogh, 2014; Rungger and Zamani, 2018), representations used therein are not directly applicable for the purpose of this paper, i.e. synthesis of a feedback controller. This is particularly because in computation of the reachable set in the mentioned works, the time interval is divided to sub-intervals and at the end of each sub-interval the reachable set up to that time point is decomposed into sub-sets to reduce the approximation errors. As a result of these intermediate decompositions, the final reachable set cannot be directly formulated as a function of the initial state and input, which is a property that we need to design a state-feedback controller.

3.1 1-step hybridization

In our proposed hybridization method, we track the reachable set Ω from an initial set $X_S \subseteq \mathcal{X}$ as the Minkowski sum of three components that construct the set: $\Omega = \Omega_X \oplus \Omega_U \oplus \Omega_W$ where Ω_X is the state reachablity component, Ω_U is the input reachability component, and Ω_W is the disturbance reachability component and each component is represented as a zonotope. We denote Ω as a 3-tuple $(\Omega_X, \Omega_U, \Omega_W)$. When computing the intermediate reachable sets for sub-intervals, we only let the algorithm decompose the state and input components, which allows us to retain a map between the state and input to the reachable set.

The hybridization procedure is summarized in Algorithm 1 which takes as an input the nonlinear system, the sample interval T, a starting zonotope X_S and two design parameters – the desired error bound δ and the number of sub-intervals N that the sample interval T should be split into. Each of the N sub-intervals will have length τ (line 1). In the beginning, $X_S \times \mathcal{U}$ is the only mode that is considered for affine approximation (line 4). The main

Input : Nonlinear system $\dot{x} = f(x, u)$; Disturbance zonotope $\mathcal{W}$; Sample interval T; Starting zonotope $X_S \subseteq \mathcal{X}$ and $U_S \subseteq \mathcal{U}$; Desired error bound δ; Number of sub-intervals N;
Output: Discrete-time PWA system $\overline{\Sigma}$;

1 $\tau \leftarrow T/N$;
2 $\Omega_X \leftarrow X_S$; $\Omega_U \leftarrow \emptyset$; $\Omega_W \leftarrow \emptyset$;
3 $\Omega \leftarrow (\Omega_X, \Omega_U, \Omega_W)$;
4 $List \leftarrow \{(\Omega, U_S)\}$;
5 $\overline{\Sigma}^T \leftarrow \{\}$;
6 **while** $List \neq \emptyset$ **do**
7 $(\Omega^0, U) \leftarrow pop(List)$
8 **for** $i \leftarrow 0$ **to** $N-1$ **do**
9 $\Phi^i \leftarrow Inflate(\Omega^i, U, f, \mathcal{W}, \tau)$;
10 $(\mathbf{A}, \mathbf{B}, \mathbf{c}, V) \leftarrow AffineApprox(f, \Phi^i, U)$
11 $(\mathbf{A}^d, \mathbf{B}^d, \mathbf{c}^d) \leftarrow DiscretizeAffine(\mathbf{A}, \mathbf{B}, \mathbf{c}, \tau)$
12 $V^d \leftarrow DisturbApprox(V \oplus \mathcal{W}, \mathbf{A}, \tau)$
13 $\Omega_X^{i+1} \leftarrow \mathbf{A}^d \Omega_X^i$;
14 $\Omega_U^{i+1} \leftarrow \mathbf{A}^d \Omega_U^i + \mathbf{B}^d U$;
15 $\Omega_W^{i+1} \leftarrow \mathbf{A}^d \Omega_W^i \oplus V^d$;
16 $\Omega^{i+1} \leftarrow (\Omega_X^{i+1}, \Omega_U^{i+1}, \Omega_W^{i+1})$;
17 **if** $|\Omega_W^N| \leq \delta$ **then**
18 $\overline{F}(\Omega^0) \leftarrow AffineParameters(\Omega^0, U)$
19 $\overline{\Sigma}.append((\Omega^N, U, \overline{F}))$
20 **else**
21 $List.append(\{Refine((\Omega^0, U))\})$
22 **return** $\overline{\Sigma}$

Algorithm 1. Discrete-Time Hybridization

cycle (lines 6-21) then processes affine modes one by one and either finds a discrete time affine function $\overline{F}$ for them (line 18-19), or refines them and queues them for affine approximation in the next iteration of the cycle (line 21). The inner cycle considers each sub-interval and computes intermediate reachable sets for them (lines 8-16).

Let us now go in detail through the functionalities used in the algorithm.

Inflate In order to compute a valid discretization of the dynamics in region Ω^i, we need to construct the continuous-time affine over-approximation in region Φ^i where the states are guaranteed to stay during the discretization interval. This can be computed using a reachable tube approximation algorithm such as one presented by Althoff and Krogh (2014) and is invoked in the algorithm as $Inflate(\Omega, U, f, W, \tau)$ (line 9).

AffineApprox Under the Assumption 1 we can compute an affine over-approximation of the non-linear function f in the region $\Phi^i \times U \subseteq \mathcal{X} \times \mathcal{U}$ as proposed by Dang et al. (2010) and refer to it as $AffineApprox(f, \Phi^i, U)$. This way, we have computed affine inclusion approximation of f that is valid in Φ^i, i.e. $f(x, u) \in \mathbf{A}x + \mathbf{B}u + \mathbf{c} \oplus V$, with V corresponding to the zonotope over-approximation of the linearization error set (line 10).

DiscretizeAffine For an affine system $\dot{x} = \mathbf{A}x + \mathbf{B}u + \mathbf{c}$, the three-component representation of the reachable set can be computed in a closed form $\Omega_X = \mathbf{A}^d \times X_S + \mathbf{c}^d$, $\Omega_U = \mathbf{B}^d \times U_S$ and $\Omega_W = \emptyset$, where:

$$\begin{bmatrix} \mathbf{A}^d & \mathbf{B}^d & \mathbf{c}^d \\ \mathbf{0} & \mathbf{I} & \end{bmatrix} = e^{\begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{c} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^T} \tag{6}$$

as explained by DeCarlo (1989). In our method, we invoke this functionality as $DiscretizeAffine(\mathbf{A}, \mathbf{B}, \mathbf{c}, T)$ (line 11).

DisturbApprox For a linear system subject to a disturbance $\dot{x} = \mathbf{A}x + \mathbf{B}u + w$, the Ω_X and Ω_U can be computed using (6). Ω_W, however, cannot be computed in a similar way as the disturbance does not remain constant between the time samples. Given that $w \in W$, the reachable set from the disturbance can be computed as follows:

$$\Omega_W = \int_{t=0}^{T} e^{At} W. \tag{7}$$

The exact computation of (7) is not a zonotope, but we can compute a zonotope over-approximation of (7). Such over-approximation can be achieved for example via the method proposed by Althoff et al. (2008); let us assume that the function $DisturbApprox(W, \mathbf{A}, T)$ provides this. The discrete time error is thus set including both the contribution from the continuous disturbance and the linearization error (line 12).

We note that the input reachable set component is computed as a $\mathbf{A}^d {\Omega_U}^i + \mathbf{B}^d U$ (line 14), hence, keeping the number of generators fixed throughout the iterations. This is because of the ZOH discretization that keeps the input fixed throughout the time step T. In contrast the disturbance reachable set component is computed as the Minkowski sum, hence, adding new generators at each iteration (line 15).

If the discrete-time error of the affine mode is within the predefined size δ (line 17), this mode is added to the resulting PWA $\overline{\Sigma}$ (line 19). Otherwise it will be refined into smaller regions in the state input product space (line 21) and the procedure is repeated for each region.

AffineParameters When a mode is being added to the resulting PWA, we have its representation in the means of the sum of three zonotope components. From them, we can calculate the parameters of the discrete-time affine approximation of the dynamics through the *AffineParameters* procedure (line 18) as follows:

$$A = \Omega_X^N.G \; X.G^{-1} \tag{8a}$$
$$B = \Omega_U^N.G \; U.G^{-1} \tag{8b}$$
$$c = \Omega_X^N.\mu + \Omega_U^N.\mu - A \; X.\mu - B \; U.\mu \tag{8c}$$
$$W = \square(\Omega_{W\,N}). \tag{8d}$$

Note that since X is a box, $X.G$ is diagonal matrix and invertible. Similar holds for U.

Correctness Note that since all states reachable from Ω^i within τ are in Φ^i and the i-th step continuous approximation $\overline{f}^i = \mathbf{A}x + \mathbf{B} + \mathbf{c} \oplus V$ is conservative in Φ^i, Ω^{i+1} is indeed an over-approximation of the reachable set from Ω^i within τ. Based on this observation, we formulate the following lemma summarizing the properties of the resulting PWA.

Lemma 1. $\overline{F}$ is a discrete-time affine function and it is a conservative abstraction of F in X_S.

Proof 1. Consider $x(0) \in X_S = \Omega^0$. Let $x(i\tau) \in \Omega^i$. By the definition of inflation we have: $\forall t \in [i\tau, (i+1)\tau],\ x(t) \in \Phi^i$. As a result of affine approximation we have:

$$\forall u \in U, t \in [i\tau, (i+1)\tau]\ f(x,u) \in \overline{f}^i(x,u),$$

and hence:

$$\chi_f(\tau; x(i\tau), u) \in \chi_{\overline{f}^i}(x(i\tau), u),$$

where $\chi_f(\tau; x(i\tau), u)$ is the reachable set of the original system from $x(i\tau)$ under constant input u and $\chi_{\overline{f}^i}$ is the reachable set of the over-approximation $\overline{f}^i$. We can further see that

$$\chi_{\overline{f}^i}(x(i\tau), u) = \{A_i x + B_i u + c_i\} \oplus \Omega_W$$

and $x((i+1)\tau) = \chi_f(\tau; x(i\tau), u)$, which implies that $x((i+1)\tau) \in \Omega^{i+1}$. Therefore, by induction we can show that for all $x(0) \in X_S$, $u \in U$:

$$F(x,u) \in A_N(A_{N-1}...) \oplus \{B_N u\} \oplus \{c_N\} \oplus W_N \overset{\text{def}}{=} \overline{F}(x,u).$$

□

Termination Algorithm 1 guarantees that in case of termination, the calculated error set will be within the predefined bounds δ; however, Algorithm 1 is not guaranteed to terminate for an arbitrarily small δ. The imposed lower bound on δ is an inevitable artifact of imposing the three component structure on the reachable set. As the disturbance component cannot be refined in the intermediate steps τ, the accumulated error cannot be arbitrarily small. This is in contrast to the methods where reachable sets can be computed with arbitrary precision and such structure cannot be imposed on the reachable set, such as (Rungger and Zamani, 2018).

3.2 Multi-step hybridization

To obtain a k-step hybridization $\overline{\Sigma}^k : \mathcal{X} \times \mathcal{U}^k \rightrightarrows \mathbb{R}^n$ of the continuous dynamics f for a given design parameter k, Algorithm 1 is first run with $X_S = \mathcal{X}$, $U_S = \mathcal{U}$ (or a zonotope over-approximation of those, if the are not zonotopes themselves) to obtain $\overline{\Sigma}^1$. We note that the reachable set of each mode of the PWA output of Algorithm 1 is a zonotope itself and is therefore compatible as an input for this algorithm. The 2-step hybridization is obtained by setting $X_S = \Omega^0$ and $U_S = U$ for each mode of $\overline{\Sigma}^1$, etc., and finally $\overline{\Sigma}^k$ is obtained by applying Algorithm 1 to X_S, U_S obtained from $\overline{\Sigma}^{k-1}$.

4. FEEDBACK CONTROL SYNTHESIS

In this section we present our approach to synthesizing a feedback controller for the k-step PWA hybridization $\overline{\Sigma}^k$ obtained using k applications of Algorithm 1. For each affine mode $\overline{F}^k$ of the system, we will compute a feedback controller that is valid in a subspace $\Omega^0 \times U \subseteq \mathcal{X} \times \mathcal{U}$. Such a controller is designed to guarantee that – starting from a given set $\Psi \subseteq \mathcal{X}$ in the state-space – the size of the reachable set will be smaller than a prescribed design parameter $|\Xi|$ ($|\Xi|$ is a vector and Ξ is a box) after the fixed k number of time steps.

The control synthesis problem is formulated as a linear programming problem, where we aim to minimize the input-effort that is required to keep the size of the reachable set within the prescribed values:

$$\begin{aligned} \min_{\nu,\,\alpha}\ & \alpha \\ \text{s.t.}\ & \forall x^0 \in \Psi,\ C_\nu(x^0 : x^{k-1}) \subseteq \alpha\mathcal{U} && \text{(9a)} \\ & \forall x^0 \in \Psi,\ \overline{F}^{(k)}(x^0, C_\nu(x^0 : x^{k-1})) \subseteq \Xi && \text{(9b)} \\ & 0 \le \alpha \le 1, && \text{(9c)} \end{aligned}$$

where C_ν denotes a general form of control ν given a sequence of states from time step 0 to $k-1$. The goal of the optimization is to find a controller that keeps the state uncertainty within the required bounds with the least input effort. The remainder of the input spectrum, i.e. $\mathcal{U}^*$, can then be used for steering purposes later.

We propose two different implementations of (9).

Control synthesis with initial state feedback In this formulation, the controller becomes $C_\nu(x^0 : x^{k-1}) = \nu x^0$; ν matrix should therefore have $k \times m$ rows and n columns to map the initial state to each input during the k time-steps, where n and m are the number of states and inputs, respectively. This formulation makes the controller indifferent to the evolution of the state during the k time-steps as the control inputs are planned ahead. The k-step disturbance set $\overline{W}^{(k)}$ in this case will be the accumulated disturbance during this time as it will not be compensated:

$$\overline{W}^{(k)} = \overline{W} \oplus A\overline{W} \oplus ... \oplus A^{k-1}\overline{W}.$$

This approach is thus favorable for cases where the system disturbance is small as it only requires $k \times m \times n = \mathcal{O}(kmn)$ variables for the controller, leading to a relatively small linear programming problem. However, since the controller uses state feedback only once in every T time steps, a feasible policy might not exist to ensure the prescribed reachable set size. As a remedy, we propose an alternative implementation with full state feedback.

Control synthesis with full state feedback Let $x^0 : x^{k-1}$ be the discrete sequence of states from time step 0 to $k-1$. In this formulation, the controller becomes $C_\nu(x^0 : x^{k-1}) = \nu(x^0 : x^{k-1})$; ν matrix should therefore have $k \times m$ rows and $k \times n$ columns to map the the states during the k time-steps to each input during this period. Note that considering causality, the number of actual control variables is $(k \times (k+1))/2 \times m \times n = \mathcal{O}(k^2 mn)$. This formulation results in a larger linear programming problem, but allows a fully closed-loop policy.

4.1 Implementation of LP constraints

Let us go through the implementation of the constraints for the case where the controller is synthesized as a function of the initial state, i.e. $C_\nu(x^0 : x^{k-1}) = \nu x^0$. The extension to the controller with full state feedback is straightforward.

To impose constraint (9a), we introduce an auxiliary matrix ν^{abs} that is the element-wise absolute value of

matrix ν. We can impose (9a) as linear constraints as follows:

$$\nu \leq \nu^{abs} \tag{10a}$$
$$-\nu \leq \nu^{abs} \tag{10b}$$
$$\sum_{g \in X.G} \nu^{abs}|g| \leq \nu U. \tag{10c}$$

where g iterates the generators of the zonotope X and inequalities are element-wise. To impose constraint (9b), we note that the closed loop dynamics of the system becomes $(\mathbf{A} - \nu\mathbf{B})x \oplus W$. We introduce another auxiliary matrix $\mathbf{A}_{cl}^{abs}$ that is the element-wise absolute value of the the matrix $\mathbf{A} - \nu\mathbf{B}$. We can impose (9b) as linear constraints as follows:

$$\mathbf{A} - \nu\mathbf{B} \leq \mathbf{A}_{cl}^{abs} \tag{11a}$$
$$-\mathbf{A} - \nu\mathbf{B} \leq \mathbf{A}_{cl}^{abs} \tag{11b}$$
$$\sum_{g \in X.G} \mathbf{A}_{cl}^{abs}|g| + \sum_{g' \in W.G} |g'| \leq |\Xi|. \tag{11c}$$

where g and g' iterate the generators of the zonotopes X and W respectively and inequalities are element-wise.

4.2 Analysis

Let us now look into the necessary and sufficient conditions for feasibility of the LP problem (9) for an affine mode with dynamics $\overline{F}^k$. We can obtain the upper bounds on the initial set size $|\Psi|$ and lower bounds on the target set size $|\Omega|$ that guarantee the feasibility of (9) from the reachable set components of the dynamics.

Regardless of the control policy, the reachable set of the closed loop dynamics $\overline{F}^k(x^0, C_\nu(x^0 : x^{k-1}))$ is larger than or equal to the reachable set from disturbance $|\square\Omega_W|$ as the disturbance is not known a-priori. So one necessary condition is that $|\square\Omega_W| \leq |\Xi|$. We note that Ω_W includes both the continuous time disturbance contribution and the linearization error in the affine mode. On the other hand if the target is larger than the reachable set without input $|\square(\Omega_X \oplus \Omega_W)| \leq |\Xi|$, (9) is trivially solved with $\alpha = 0$.

Let us introduce the *inner box* $\sqsubset \Omega_U$ of the zonotope Ω_U.The inner box of a zonotope can be computed as an LP. $\sqsubset \Omega_U$ indicates the area where the reachable set of a state can be moved independently in each dimension through the input. This is therefore a lower bound on how the reachable set can be modified through state feedback. As a result (9) is feasible under the following sufficient condition:

$$|\Xi| \geq max(|\square\Omega_X| + |\square\Omega_W| - |\sqsubset \Omega_U|, |\square\Omega_W|). \tag{12}$$

Fig. 2 shows an example of the three components of the reachable set, i.e. $\Omega_X, \Omega_U, \Omega_W$ for an inverted pendulum in a region around the upright position.

Theorem 1. Let us assume for a starting set Ψ, a target set Ξ can be chosen such that it satisfies (12) for affine dynamics $\overline{F}$ and $|\Xi| \leq r|\Psi| + |\square\Omega_W|$ where $0 \leq r < 1$. The closed loop dynamics in Ψ (controlled with the solution to (9)) satisfies dissipation from input-to-state incremental stability (5).

Proof. The closed loop dynamics from Ψ as a solution to (9) becomes $(\mathbf{A} - \nu\mathbf{B})x \oplus \square\Omega_W$. Let $\overline{e}$ be the greatest eigenvalue of $\mathbf{A} - \nu\mathbf{B}$ and $\overline{E}$ be the boundary state in Ψ along the corresponding eigenvector. Assuming $\overline{e} > r$, the reachable set from the E becomes $\overline{e}E \oplus \square\Omega_W$ that contradicts $|\Xi| \leq r|\Psi| + |\square\Omega_W|$. Therefore, r is an upper bound on the eigenvalues of $\mathbf{A} - \nu\mathbf{B}$. We get as a result:

$$\Delta(F^{cl}(x, u^*), F^{cl}(x', u^*)) \leq r(||x - x'||_\infty) + ||\square\Omega_W||_\infty \tag{13}$$

following:

$$\begin{aligned}\Delta(F^{cl}(x, u^*), F^{cl}(x', u^*)) - ||x - x'||_\infty \\ \leq (r-1)(||x - x'||_\infty) + ||\square\Omega_W||_\infty\end{aligned} \tag{14}$$

that satisfies (5). □

Input allocation: In the optimization problem (9) we find the smallest input range αU required to satisfy the reachable set constraint from a set Ψ. However if we apply the computed feedback rule to a set with a different size than $|\Psi|$, The required input range will be different. Given that in our approach we want to keep part of the input range for the symbolic controller, we introduce the input allocation design parameter $0 < \rho < 1$ to ensure that our input allocation will not exceed that value regardless of the cell size. If applying the feedback rule to a cell results in a higher input range $\lambda\rho U$ where $\lambda > 1$, we weaken the feedback, i.e. multiply by $1/\lambda$ such that the input range will remain within the allocated ρU.

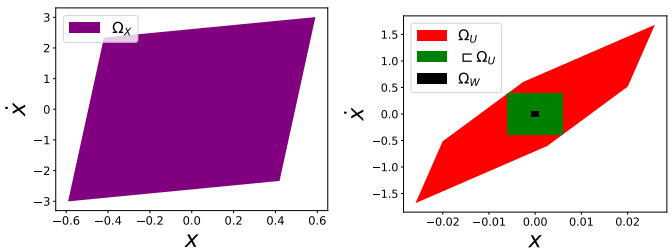

Fig. 2. (a) State reach set. (b) Input/disturbance reach sets.

5. SIMULATIONS

We have implemented the proposed algorithm in Python and applied it to the reach-stay problem for the cart-pendulum system with parameters presented in an earlier study of this problem in ROCS (Li and Liu, 2017):

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{mgl}{J_t}\sin x_1 - \frac{b}{J_t} + \frac{l}{J_t}\cos x_1 u, \end{cases} \tag{15}$$

with $x_1 = \theta$ and $x_2 = \dot{\theta}$ corresponding to pendulum's angle and angular velocity ($\theta = 0$ corresponding to upright position). The parameters are: $J_t = J + ml^2$, $m = 0.2kg$, $g = 9.8m/s^2$, $l = 0.3m$, $J = 0.006kgm^2$, $b = 0.1N/m/s$. The goal is to design a controller to reach the upright position.

System (15) is used by Li and Liu (2018) with time discretization $T = 0.01s$ to evaluate the ROCS symbolic controller. We have used the same time discretization to construct a k-step hybridizations of the system and feedback controllers for each mode of the hybridization. For hybridization, $N = 10$ and $\delta = [0.1, 0.5]$ are selected. For control synthesis, the initial state feedback is used and the prescribed reachable set size $|\Xi|$ is selected as $1.1|\Omega_W|$; this

k	ρ	T_{Hybrid}	T_{LP}	T_{Symb}	T_{Total}	n_{cells}
2	0.1	0.9	0.1	567	569	40299
2	0.2	0.9	0.1	420	421	25345
2	0.4	0.9	0.1	284	285	12173
4	0.1	No solutions found				
4	0.2	6.5	0.5	137	144	2841
4	0.4	6.5	0.5	91	98	1245
6	0.1	No solutions found				
6	0.2	10	1.5	120	132	1707
6	0.4	10	1.5	55	67	779

Table 1. Computation times in sec.

is motivated by the fact that the reachable set is always larger than or equal to Ω_W. We then use (our python re-implementation of) ROCS symbolic controller to compare the results with and without feedback application. The input by Li and Liu (2018) is chosen from the discrete set {-10,-9.95,...,9.95,10}, therefore we consider continuous bounded control input $u \in [-10, 10]$ for feedback synthesis.

The algorithm has been run on a laptop with a 2.8 GHz CPU and 16 GB of RAM on Python. We have used Gurobi Optimization (2020) as solver for the LPs. In Table 1, the hybridization, feedback synthesis via linear programming, symbolic synthesis times, and total computation times are reported as well as number of constructed abstract cells for different selections of steps k and input allocation for feedback ρ. We observe that increasing both k and ρ results in faster synthesis and fewer abstract cells as intuitively, larger cells can be constructed for backward reachablity. In comparison, running symbolic synthesis only without hybridization and feedback synthesis for the same problem takes 400s in the original ROCS implementation and 630s in our Python re-implementation and results in 26340 states. This means that with a 6-step controller and $\rho = 0.4$ we observe an 8 times improvement in the symbolic synthesis speed and 30 times reduction in number of abstract states.

However, with a higher k, the reachable set from disturbance Ω_W also becomes larger and at some point becomes the dominating factor and impedes successful controller synthesis for shrinking the reachable set. This effect can already be seen in $k = 4$ and $k = 6$ when only a small part of the input is allocated for feedback, i.e. $\rho = 0.1$. In this case, due to large Ω_W, the feedback effort may not be enough to guarantee solution existence to the LP problem (9), i.e. shrinking reachable sets. ρ cannot also be selected arbitrarily high as it limits the input range available for symbolic controller. In fact for $\rho > 0.5$ in this problem, we are unable to find a solution as a higher input value is required to bring the pendulum back from the state $[1, 1]$.

The abstraction and the resulting feedback controller are shown in Fig. 3 for $k = 6$ and $\rho = 0.4$. The red stars correspond to time-stamps where the control sequence is obtained from the abstraction and the blue stars correspond to the intermediate steps.

6. CONCLUSIONS AND FUTURE WORK

We introduced a hybridization-based feedback control synthesis method for potentially unstable nonlinear continuous-time systems. After time discretization of the given system, we proposed design of a multi-step feedback control that ensures dissipation form of incremental input-to-state stability property. We implemented and evaluated the algorithm on a cart-pendulum use case and noticed that application of feedback control can indeed improve abstraction-based control by reducing the number of required abstract states and synthesis time. We realized that while increasing the number of steps and input range allocated to feedback increasingly improves the synthesis time, they are limited by accumulated error over time and input range needed for symbolic control synthesis.

Our future work will look into several directions. We plan to explore the benefits of integrating the technique with abstraction-based tools for control synthesis under reachability and temporal logic specifications, such as SCOTS or ROCS. We also aim to investigate in detail conditions under which Algorithm 1 terminates, namely to derive the smallest error bound δ that guarantees termination.

REFERENCES

Althoff, M. and Krogh, B.H. (2014). Reachability analysis of nonlinear differential-algebraic systems. *IEEE Transactions on Automatic Control*, 59(2), 371–383.

Althoff, M., Stursberg, O., and Buss, M. (2008). Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Proc. of the 47th IEEE Conference on Decision and Control*.

Dang, T., Maler, O., and Testylier, R. (2010). Accurate hybridization of nonlinear systems. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, 11–20. ACM.

DeCarlo, R.A. (1989). *Linear systems: A state variable approach with numerical implementation*. Prentice-Hall, Inc.

Girard, A., Le Guernic, C., and Maler, O. (2006). Efficient computation of reachable sets of linear time-invariant systems with inputs. In J.P. Hespanha and A. Tiwari (eds.), *Hybrid Systems: Computation and Control*, 257–271. Springer Berlin Heidelberg, Berlin, Heidelberg.

Girard, A. and Pappas, G.J. (2007). Approximate bisimulation relations for constrained linear systems. *Automatica*, 43(8), 1307–1317.

Gurobi Optimization, L. (2020). Gurobi optimizer reference manual. URL `http://www.gurobi.com`.

Li, Y. and Liu, J. (2017). Invariance control synthesis for switched nonlinear systems: An interval analysis approach. *IEEE Transactions on Automatic Control*, 63(7), 2206–2211.

Li, Y. and Liu, J. (2018). ROCS: A robustly complete control synthesis tool for nonlinear dynamical systems. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*, 130–135.

Liu, J. and Ozay, N. (2014). Abstraction, discretization, and robustness in temporal logic control of dynamical systems. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, 293–302.

Mouelhi, S., Girard, A., and Gössler, G. (2013). CoSyMA: a tool for controller synthesis using multi-scale abstractions. In *Proceedings of the 16th international confer-*

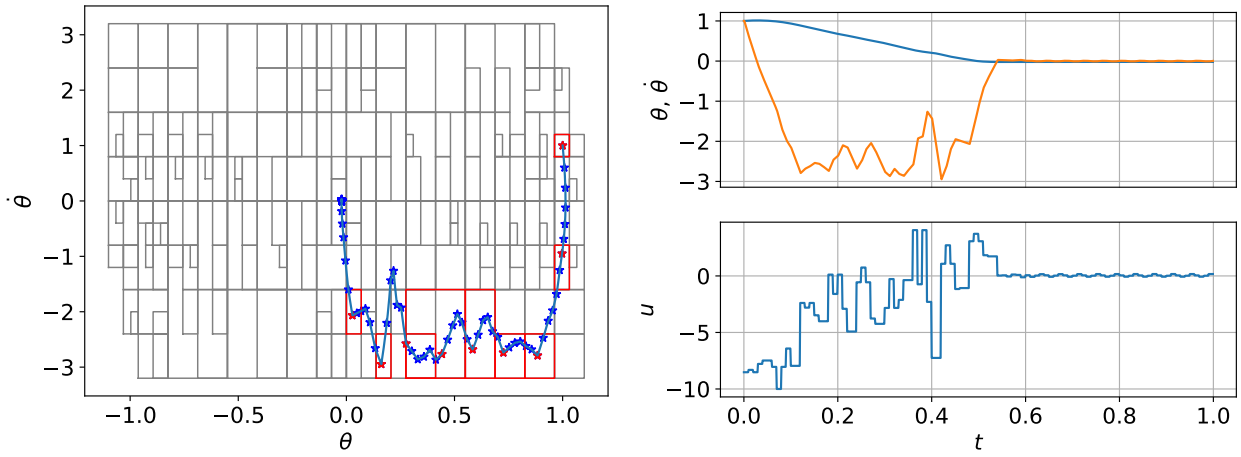

Fig. 3. (a) Abstraction of the state space. (b) Close loop simulation from $x = (1.0, 1.0)$.

ence on Hybrid systems: computation and control, 83–88.

Reißig, G. (2011). Computing abstractions of nonlinear systems. *IEEE Transactions on Automatic Control*, 56(11), 2583–2598.

Rungger, M. and Zamani, M. (2016). SCOTS: A tool for the synthesis of symbolic controllers. In *Proceedings of the 19th international conference on hybrid systems: Computation and control*, 99–104.

Rungger, M. and Zamani, M. (2018). Accurate reachability analysis of uncertain nonlinear systems. In *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control (part of CPS Week)*, 61–70.

Schürmann, B. and Althoff, M. (2017). Guaranteeing constraints of disturbed nonlinear systems using set-based optimal control in generator space. *IFAC-PapersOnLine*, 50(1), 11515–11522.

Sinyakov, V. and Girard, A. (2020). Abstraction of monotone systems based on feedback controllers. In *21st IFAC World Congress*.

Tabuada, P. (2006). Symbolic control of linear systems based on symbolic subsystems. *IEEE Transactions on Automatic Control*, 51(6), 1003–1013.

Tajvar, P., Varava, A., Kragic, D., and Tumova, J. (2019). Robust motion planning for non-holonomic robots with planar geometric constraints. In *The International Symposium on Robotics Research October 6-10, 2019, Hanoi, Vietnam*.

Tran, D.N., Rüffer, B.S., and Kellett, C.M. (2016). Incremental stability properties for discrete-time systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 477–482. IEEE.

Yordanov, B., Tumova, J., Cerna, I., Barnat, J., and Belta, C. (2011). Temporal logic control of discrete-time piecewise affine systems. *IEEE Transactions on Automatic Control*, 57(6), 1491–1504.