Computational Advantages of a Two-level Hybrid Control Architecture

T. Moor * J. Raisch § J.M. Davoren *

Abstract

We investigate a two-level hierarchical architecture for hybrid control. On the top level, a discrete supervisory controller acts on quantised measurement information by switching between a finite number of continuous controllers in order to enforce a language inclusion specification. A widely accepted approach to this problem is to first construct a discrete abstraction of the continuous low-level feedback loops and to subsequently resort to DES techniques to solve the high-level synthesis problem. While in principle adopting this approach, we show how to use the structure induced by the low-level controllers to significantly increase computational efficiency of the abstraction procedure. Our methodology enables the system designer to exploit a trade-off between the increase in computational efficiency and the loss in controller flexibility caused by the specific hierarchical structure.

Keywords: hybrid systems, hierarchical control, discrete abstraction, supervisory controller synthesis.

1 Introduction

Hybrid control systems are mathematical models of heterogeneous systems consisting of digital components interacting in real-time with continuous processes. In particular the task of controller design for such systems has received extensive attention in the recent literature e.g. [AB+00, FDF99, KA+00, LTS99, MR99]. In this paper, we reexamine a scenario that has been commonly used as a motivation for hybrid control. It consists of a continuous plant model, a finite number of continuous controllers and a discrete supervisor which acts on quantised measurement information (events) by switching between the continuous controllers; see Figure 1. A standard problem in hybrid control is to synthesise a discrete supervisor such that the overall control system satisfies a language inclusion specification; e.g. [KA+00, MR99, RO98]. Solutions to this problem are typically based on computing a suitable (i.e., conservative) discrete abstraction for the continuous part of the overall system. The crucial computational challenge in this step is to reliably estimate sets of continuous states reachable under continuous flows from different sets of initial conditions. For fairly large classes of continuous dynamics this can be done by employing a regular quantisation grid in the continuous state space; e.g. [AB+00, FMR00, LTS99][1]. Clearly, this puts a rather stringent limit on the problem state dimension.

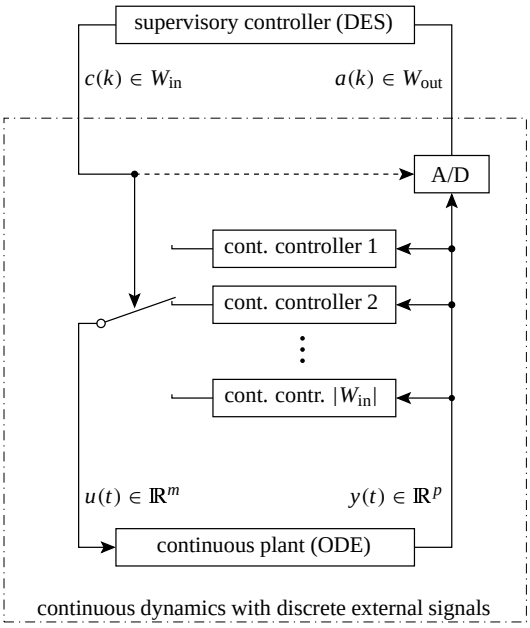

Figure 1: Two-level hierarchical control architecture

It is obvious that the scenario described above exhibits a (two-level) hierarchical structure: the continuous feedback loops can be interpreted as lower-level control, the supervisor to be designed as a higher-level controller. In the context of control, hierarchies are mostly introduced to "break" a complex problem into a number of more tractable problems [CW98, FDF99, PLS00, RIM00, WW96] and hence to reduce the overall "solution effort". We therefore expect that the hierarchical structure in our set-up can be exploited to significantly reduce the computational burden in the abstraction step. More precisely, we will argue that the presence of low-level controllers may considerably reduce the dimension of the part of the continuous state space that is relevant for the abstraction step.

The paper is organised as follows. In Section 2 we state a

*Research School of Information Sciences and Engineering, Australian National University, Canberra, email: thomas.moor@anu.edu.au, j.m.davoren@anu.edu.au, research partially supported by US Office of Naval Research, Grant N 00014-98-1-0535

§Lehrstuhl für Systemtheorie technischer Prozesse, Otto-von-Guericke Universität, and Max-Planck-Institut für Dynamik komplexer technischer Systeme, Magdeburg, Germany, email: raisch@mpi-magdeburg.mpg.de

[1]Technically, [LTS99] restates the reachability problem as a partial differential equation, which is then to be solved numerically.

model of the considered class of hybrid systems and give a link to the problem of supervisory controller synthesis. In Section 3 we characterise a class of low-level control goals and illustrate these by a simple example. The construction of a discrete abstraction is developed in Section 4. We take special account for the linear case in Section 5.

2 A two-level hierarchical control architecture

We introduce a mathematical model of a hybrid control architecture in which a discrete event supervisor switches between a finite number of continuous low-level controllers attached to a continuous plant; see Figure 1. As far as the switching and event generating mechanisms are concerned, the suggested scenario is closely related to the framework of hybrid automata; e.g. [AC+95]. However, in our model we give account of the hierarchical structure and also require an explicit notion of inputs and outputs in order to state and solve a supervisory controller synthesis problem.

We begin by modelling the continuous dynamics for the situation in which a particular low-level controller is attached to the continuous plant

$$\dot{\xi}(t) = f_{\text{plt}}(\xi(t),\, u(t))\,, \quad y(t) = g_{\text{plt}}(\xi(t),\, u(t))\,. \tag{1}$$

We assume that all low-level controllers share one and the same state variable $\zeta(t)$; i.e., they are all padded to be of the same order and there is no re-initialisation in the process of switching[2]. The active low-level controller

$$\dot{\zeta}(t) = f_c(\zeta(t),\, y(t))\,, \quad u(t) = g_c(\zeta(t)) \tag{2}$$

is selected by the most recent discrete input event $c \in W_{\text{in}}$, and the overall continuous feedback system is represented by a state space model

$$\dot{x}(t) = F_c(x(t))\,, \tag{3}$$

with state variable $x(t) = \text{Col}[\xi(t),\, \zeta(t)] \in \mathbb{R}^n$. We assume F_c to be locally Lipschitz continuous. Hence, for any given initial condition $x(0) = x_0$ the system (3) exhibits a unique solution $\Phi_c(x_0,\, \cdot\,)$, defined on the maximum interval $T_{\max}(c, x_0) \subseteq \mathbb{R}_0^+$ of existence:

$$\Phi_c(x_0,\, \cdot\,)\colon T_{\max}(c, x_0) \to \mathbb{R}^n\,. \tag{4}$$

For the following discussion on the generation of output events let $c \in W_{\text{in}}$ denote the most recent input event; let t_0 denote the time at which c was applied; and let $x_0 = x(t_0)$ denote the state of continuous feedback system at time t_0. The continuous feedback system then evolves according to $x(t_0 + t) = \Phi_c(x_0,\, t)$, $t \geq 0$. This evolution is meant to be restricted to the so called *mode invariant set* $\text{Inv}_c \subseteq \mathbb{R}^n$, which for the scope of this paper is assumed to be open and bounded. An output event is triggered when $x(t_0+t)$, $t \geq 0$, first leaves Inv_c, which happens at[3]

$$t = t'(x_0,\, c) \coloneqq \sup\{s |\, \Phi_c(x_0,\, \tau) \in \text{Inv}_c\, \forall\, 0 \leq \tau < s\}\,. \tag{5}$$

[2]Both these assumptions are not essential to our hierarchical architecture and may be dropped at the cost of some extra notation.

[3]Note that boundedness of Inv_c and Lipschitz continuity of F_c ensure that either $T_{\max}(x_0, c) = \mathbb{R}_0^+$ or the trajectory leaves Inv_c.

Equation (5) exhibits two special cases: $t'(x_0,\, c) = \infty$ indicates that the state trajectory evolves within Inv_c for all future; $t'(x_0,\, c) = 0$ corresponds to $x_0 \notin \text{Inv}_c$. Both special cases are considered as errors and trigger a distinguished output event $a_{\text{err}} \in W_{\text{out}}$. It will be the task of the supervisors to prevent a_{err} by switching low-level controllers only when the continuous state is within a appropriate mode invariant and the trajectory leaves that invariant within a finite time; i.e., $t'(x_0,\, c) \in (0, \infty)$. In this case, a quantised version of the continuous output $y(t_0 + t'(x_0,\, c))$ will be generated as an output event. The A/D-map used for output quantisation may depend on c and is modelled by a finite cover of the output space $\mathbb{R}^p$. This in turn induces a finite cover $\mathbb{R}^n = \cup_{a \in W_{\text{out}}} \mathcal{X}_{c,a}$ that relates continuous states and output events. Thus, the next output event a may not be uniquely determined by x_0 and c, but is characterised by $a \in a'(x_0,\, c)$, where

$$x'(x_0,\, c) \coloneqq \Phi_c(x_0,\, t'(x_0, c))\,, \tag{6}$$

$$a'(x_0,\, c) \coloneqq \{\alpha |\, x'(x_0,\, c) \in \mathcal{X}_{c,\alpha}\}\,, \tag{7}$$

if $t'(x_0, c) \in (0, \infty)$, or else $x'(x_0,\, c) \coloneqq x_0$, $a'(x_0,\, c) \coloneqq \{a_{\text{err}}\}$ to indicate an error.

While the internal evolution of the continuous plant under low-level control is clearly driven by continuous dynamics, the interface to the supervisor is exclusively based on discrete events: from the perspective of the supervisor[4] the switched continuous feedback loops (as indicated by the dashed box in Figure 1) can be represented by a single discrete behaviour $\mathfrak{B} \subseteq (W_{\text{in}} \times W_{\text{out}})^{\mathbb{N}_0}$, namely the set of all possible sequences of external events. Given a set of considered initial states $\mathcal{X}_0 \subseteq \mathbb{R}^n$, we formally define

$$\mathfrak{B} \coloneqq \{(c_k,\, a_k)_{k \in \mathbb{N}_0} |\, \exists\, (x_k)\ \forall\, k :\ x_0 \in \mathcal{X}_0,\ x_{k+1} = x'(x_k,\, c_k),\ a_k \in a'(x_k,\, c_k)\}\,. \tag{8}$$

See [Wi91] for a comprehensive introduction to Willems' behavioural systems theory. For the problem of supervisory controller synthesis we consider the continuous plant, the low-level continuous controllers and the A/D-map to be given. We then ask for a (high-level) discrete supervisor that restricts the behaviour $\mathfrak{B}$ according to a given language inclusion specification; i.e., the supervisor shall prevent the system to evolve on trajectories that are deemed to be unacceptable. Formally, the behaviour $\mathfrak{B}_{\text{cl}}$ of the overall hybrid control system is defined as the intersection of $\mathfrak{B}$ with the supervisory controller behaviour $\mathfrak{B}_{\text{sup}}$. The language inclusion specification for $\mathfrak{B}_{\text{spec}}$ requires that:

$$\mathfrak{B}_{\text{cl}} \coloneqq \mathfrak{B} \cap \mathfrak{B}_{\text{sup}} \subseteq \mathfrak{B}_{\text{spec}}\,. \tag{9}$$

For a thorough discussion on how to synthesise $\mathfrak{B}_{\text{sup}}$ according to (9) when $\mathfrak{B}$ and $\mathfrak{B}_{\text{spec}}$ are given we refer the

[4]We only consider DESs as potential supervisors. In particular, the timing of (traditional) DESs refers to "logic time" $k \in \mathbb{N}_0$ induced by counting events. However, if some aspects of real time shall be made available to the supervisor, a continuous timer can be formally modelled as a part of the plant, issuing timer events via quantised continuous output.

reader to our earlier work [MR99]: employing slightly modified versions of DES supervisory control methods (e.g. [RW87]), our main result is that the hybrid synthesis problem can be solved via a discrete abstraction; i.e., a finite automaton which realizes a behaviour $\mathfrak{B}_{\text{ca}} \supseteq \mathfrak{B}$. This reduces the supervisory controller synthesis problem to the construction of a discrete abstraction, to which we come back in Section 4.

3 Low-level control goals

In the absence of further restrictions on the low-level controllers and on the continuous plant, virtually any continuous closed-loop could be assembled and thus it would not be expected to gain from the hierarchical structure. However, in practical applications the low-level controllers will be designed to achieve certain low-level control goals: the low-level controllers will implement *elementary manoeuvres* that are meaningful w.r.t. the language inclusion specification which in turn is to be enforced by the supervisor. While we will not discuss the (rather challenging) question of how to choose these low-level control goals, we will characterise a relevant class of low-level control goals that allow a significant increase of computational efficiency in the course of the high-level supervisory controller design.

Given a smooth map $h_c\colon \mathbb{R}^n \to \mathbb{R}^m$, $m < n$, we assume that the low-level controller corresponding to $c \in W_{\text{in}}$ enforces the following continuous closed-loop properties for every solution $x(t)$ of (3):

$$\lim_{t\to\infty} h_c(x(t)) = 0\,, \tag{10}$$

$$h_c(x(0)) = 0 \quad \Rightarrow \quad h_c(x(t)) \equiv 0\,. \tag{11}$$

i.e., the low-level controller stabilises a quantified aspect of the state trajectories but does not necessarily stabilise the entire system (3). By the assumptions of both $\text{Rk}(h_c) \equiv m$ and the existence of a root $h_c(x_0) = 0$, we ensure that

$$\mathcal{E}_c \coloneqq \{\xi \mid h_c(\xi) = 0\} \subseteq \mathbb{R}^n \tag{12}$$

is an $(n-m)$-dimensional differentiable manifold and that the solutions of (3) can be *locally* decomposed[5] in an m-dimensional component towards $\mathcal{E}_c$ and an $n-m$-dimensional component parallel to $\mathcal{E}_c$. In order to avoid a lengthly discussion within a differential geometry framework, we restrict our further considerations to the case where a diffeomorphism $T_c\colon \mathbb{R}^n \to \mathbb{R}^n$ exists that represents a *global* coordinate transformation

$$\begin{bmatrix} w(t) \\ z(t) \end{bmatrix} \coloneqq T_c(x(t))\,, \tag{13}$$

such that $z(t) = h_c(x(t))$. Then (3) can be rewritten as

$$\dot{w}(t) = G_c(w(t),\, z(t))\,, \quad \dot{z}(t) = H_c(w(t),\, z(t))\,, \tag{14}$$

where (10) and (11) transform to $\lim_{t\to\infty} z(t) = 0$, and $z(t) \equiv 0$ whenever $z(0) = 0$, respectively.

As an example for our class of control goals, consider the following simplified model of a floating platform with accelerators in both Cartesian position coordinates:

$$\dot{w}(t) = z(t)\,, \quad \dot{z}(t) = -\alpha\, z(t) + u(t)\,, \tag{15}$$

where $w(t) \in \mathbb{R}^2$ is the position of the platform, $z(t) \in \mathbb{R}^2$ is its velocity, $u(t)$ is the actuator input, and $\alpha > 0$ is a normalised friction coefficient. Suppose that the high-level specification $\mathfrak{B}_{\text{spec}}$ demands the vehicles to travel through certain partition blocks of $\mathbb{R}^2$ in a prescribed order. A sensible design of the low-level controllers then implements e.g. straight line manoeuvres in the four directions $W_{\text{in}} = \{\text{up}, \text{dn}, \text{lft}, \text{rgt}\}$, corresponding to the unit vectors v^*_{up}, v^*_{dn}, v^*_{lft}, v^*_{rgh}. As low-level controllers we consider a proportional feedback of the velocity error:

$$u(t) = (\alpha + \beta)\, v^*_c - \beta\, z(t)\,, \tag{16}$$

for each $c \in W_{\text{in}}$. The overall gain of the low-level controller comes out to be $\alpha + \beta$. Note that physical limitations of the actuators and the given operating range translate to a limit on that gain; i.e., the continuous feedback loops may not be considered to be arbitrarily fast. See Figure 2 for a typical trajectory of our example. Along the interval of time $[t_1, t_2]$ the platform floats according to $c = c_{\text{up}}$. The trajectory in the illustration happens to start in an upwards motion and consequently evolve within the affine subspace $\mathcal{E}_c$ of states with velocity v^*_{up}. At t_2 the system switches to the low-level controller indexed by $c = c'_{\text{rgt}}$. Thus, during $[t_2, t_3]$ the velocity is driven towards v^*_{rgt}, until finally the trajectory leaves the mode invariant $\text{Inv}_{c'}$ at time t_3 and triggers an output event $a' \in W_{\text{out}}$. Whether or not this trajectory meets the high-level specification $\mathfrak{B}_{\text{spec}}$ in general depends not only on c' and a' but also on all past and all future events. Suppose our particular trajectory is meant to *not* leave $\text{Inv}_{c'}$ via the grey shaded face because that would generate an event that violates $\mathfrak{B}_{\text{spec}}$. It is clear from the

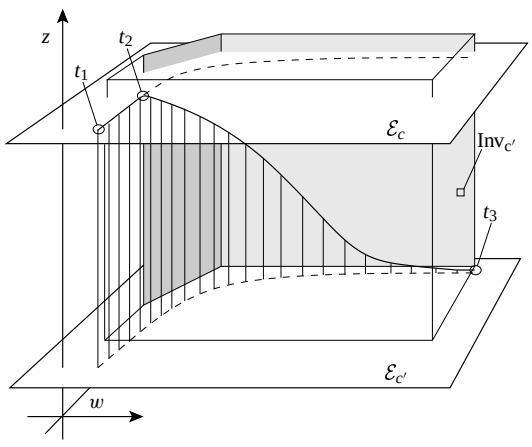

Figure 2: Elementary manoeuvres

[5] By a *local decomposition* we refer to a family of diffeomorphisms, each defined on a coordinate neighbourhood, and each performing a coordinate transformation such that the decomposition then can be represented by two canonical linear projections. In the context of this paper we can restrict our considerations to any compact superset of the bounded set Inv_c. Hence, we can assume that the family of diffeomorphisms is finite.

picture that in this situation the full dimensional continuous dynamics are still relevant even if in the long term the derivative of $z(t)$ vanishes.

4 Discrete abstraction of continuous dynamics

A discrete abstraction of the behaviour $\mathfrak{B}$ is a finite automaton P that realizes a behaviour $\mathfrak{B}_{\text{ca}} \supseteq \mathfrak{B}$; i.e., P generates every external trajectory within $\mathfrak{B}$, but there may exist trajectories of P which are not within $\mathfrak{B}$. One strategy for constructing a discrete abstraction of a continuous flow is to partition the continuous state space by a finite number of comparatively small cells and then to find a reliable approximation on the evolution of those cells; e.g. [KA+00, AB+00, FMR00, MR00]. The choice of the partition cells may be based on a careful analysis of the flows Φ_c, or one may just use a regular grid. The latter approach in principle is applicable to a large class of continuous dynamics. However, the number of cells required for a suitably accurate representation depends exponentially on the dimension of the statespace: using e.g. a regular grid of cells with a mesh width ρ to discretise an n-dimensional cube with edges of length r, $\rho \ll r$, one ends up with about $(r/\rho)^n$ cells. As every cell needs to be integrated for each input event $c \in W_{\text{in}}$, this approach imposes a stringent limit on the dimension of the state space. In this section we demonstrate how to exploit the hierarchical architecture in order to significantly reduce the number of required cells.

The crucial ingredient of our hierarchical architecture is that according to the low-level control goals the continuous state is expected to evolve "most of the time" close to a lower dimensional manifold $\mathcal{E}_c$. It is only in the situation just after switching to another low-level controller that the dynamics in fact depend on the full n-dimensional model. Note that we do need to consider this full dimensional motion in the construction of P in order to achieve the language inclusion $\mathfrak{B}_{\text{ca}} \supseteq \mathfrak{B}$. As we still want to reduce the number of cells, we restrict the supervisor not to switch low-level controllers unless their control goal is achieved; i.e., after $c \in W_{\text{in}}$ has been applied the state is required to be sufficiently close to $\mathcal{E}_c$ before the next switching of low-level controllers may take place. In turn, we only need to discretise the lower dimensional manifolds plus an extra margin of width $\rho > 0$ to cover the fact that the trajectories approach but not necessarily enter $\mathcal{E}_c$. This strategy virtually reduces the dimension of the state space to $n - m$ and thus promises substantial gain of computationally efficiency.

For each $c \in W_{\text{in}}$ let $\mathbb{R}^n = \cup_{j\in J} \mathcal{Z}_{c,j}$ denote a finite cell decomposition of the full state space. However, the only cells $\mathcal{Z}_{c,j}$ that are *relevant* are those that intersect both the mode invariant Inv_c and the manifold $\mathcal{E}_c$. In order to achieve a certain accuracy, we ask relevant cells to have a diameter not more than a target diameter $\rho > 0$; i.e., $\text{Diam}(\mathcal{Z}_{c,j}) := \sup\{||\xi - \zeta|| \mid \xi, \zeta \in \mathcal{Z}_{c,j}\} \leq \rho$. As Inv_c is bounded and $\mathcal{E}_c$ is of dimension $n - m$, J can be chosen to be finite and the number of required cells is of order $\mathcal{O}((1/\rho)^{n-m})$. Such a cell decomposition can be based on mapping a regular $\mathbb{R}^{n-m}$-grid via the diffeomorphism T_c into $\mathcal{E}_c$ and then expanding the images of the $\mathbb{R}^{n-m}$-cells by a uniform fattening of width $\gamma \ll \rho$. Formally, this γ-*expansion* of a cell $\mathcal{Z}_{c,j}$ is defined by $B_\gamma(\mathcal{Z}_{c,j}) := \{\xi \mid \exists\, \zeta \in \mathcal{Z}_{c,j} : ||\xi - \zeta|| < \gamma\}$. Due to the γ-expansion, the relevant cells will cover not only $\text{Inv}_c \cap \mathcal{E}_c$ but also $B_\gamma(\text{Inv}_c \cap \mathcal{E}_c)$. This is required in order to ensure that any trajectory eventually evolves within relevant cells only.

As the discrete state space for our abstraction we choose $Q = (C \times J) \cup \{q_{\text{err}}\}$ representing the cells $\mathcal{Z}_{c,j}$ plus one distinguished error state. We pick a set of discrete initial states Q_0 that corresponds to a cell cover of the continuous initial states; e.g. $Q_0 := \{(c, j) \mid \mathcal{Z}_{c,j} \cap \mathcal{X}_0 \neq \emptyset\}$. We then need to set up a transition relation $\delta \subseteq Q \times (W_{\text{in}} \times W_{\text{out}}) \times Q$ such that the induced behaviour

$$\mathfrak{B}_{\text{ca}} := \{(c_k, a_k)_{k\in\mathbb{N}_0} \mid \exists\, (q_k)\ \forall\, k : q_0 \in Q_0,\ (q_k, (c_k, a_k), q_{k+1}) \in \delta\} \quad (17)$$

is a superset of $\mathfrak{B}$. The construction of δ is based a conservative approximation on the evolution of individual cells $\mathcal{Z}_{c,j}$ under the flows $\Phi_{c'}$ restricted to the mode invariant $\text{Inv}_{c'}$. This problem of reachability is a fundamental issue in hybrid systems research. Known methods mainly focus on linear differential equations and polyhedral or ellipsoidal sets of states; e.g. [AB+00, FMR00, KV00]. A technique addressing the nonlinear case is given in [LTS99]. For the scope of this paper, we focus on a conservative estimate of reachable states based on integration of cells $\mathcal{Z}_{c,j}$ with a fixed sampling rate Δ; see also [MR00]. Technically, we require an implementation of two operators $\mathcal{R}_{c',t}$ and $\mathcal{S}_{c',\Delta}$ on sets of states, where $\mathcal{R}_{c',t}(X) \supseteq \Phi_{c'}(X, t)$ over-approximates the set of states reachable at a particular instance on time t, while $\mathcal{S}_{c',\Delta}(X) \supseteq \Phi_{c'}(X, [0, \Delta])$ returns an over-approximation of the states reachable within the sampling interval $[0, \Delta]$. We then iterate

$$X_0 := \mathcal{Z}_{c,j}\,, \qquad X_{i+1} := \mathcal{R}_{c',\Delta}(X_i)\,, \quad (18)$$

$$i^+ := \min\{i \mid X_i \cap \text{Inv}_{c'} = \emptyset\}\,, \qquad \tilde{X}_i := \mathcal{S}_{c',\Delta}(X_i)\,, \quad (19)$$

assuming termination with $i^+ < \infty$. For a more sophisticated version of this iteration and some account on the question of finite termination see [MR00]. By construction, the evolution of $\mathcal{Z}_{c,j}$ under $\Phi_{c'}$ restricted to $\text{Inv}_{c'}$ takes place within the union $\cup_{i<i^+} \tilde{X}_i$.

Based on this reachset estimate, we define δ to be the set of all those transitions $(q, (c', a'), q') \in Q \times (W_{\text{in}} \times W_{\text{out}}) \times Q$ which fulfil one of the following conditions:

- (A1) $q = (c, j)$, $q' = (c', j')$ and there exists $i < i^+$ such that $\tilde{X}_i \cap \mathcal{X}_{c',a'} \not\subseteq \text{Inv}_{c'}$ and $\tilde{X}_i \cap \mathcal{Z}_{c',j'} \neq \emptyset$;
- (A2) $q' = q_{\text{err}}$ and $\mathcal{Z}_{c,j} \not\subseteq \text{Inv}_{c'}$;
- (A3) $q' = q_{\text{err}}$ and there exists $i < i^+$ such that $\tilde{X}_i \not\subseteq B_\gamma(\mathcal{E}_{c'})$ and $\tilde{X}_i \cap \mathcal{X}_{c',a'} \not\subseteq \text{Inv}_{c'}$.
- (A4) $q = q' = q_{\text{err}}$.

We comment on the above conditions: (A1) captures the nominal case that a trajectory hits the boundary $\partial \mathrm{Inv}_{c'}$ and thus triggers the output event a'. (A2) gives q_{err} when the initial cell does not entirely lie within the mode invariant $\mathrm{Inv}_{c'}$. (A3) identifies the situation in which an output event is generated before the cell has evolved into the γ-expansion of $\mathcal{E}_{c'}$ to be an error. (A4) implements arbitrary events once the error state has been reached in order to ensure $\mathfrak{B}_{\mathrm{ca}} \supseteq \mathfrak{B}$.

The supervisor only switches low-level controllers as an immediate consequence of an output event, and thus (A3) rules out changes of low-level control before the recent low-level goal is attained. This is a rather conservative rule, as the supervisor may receive an output event but in reply just reconfirm the recent choice of low-level control. Depending on the specification $\mathfrak{B}_{\mathrm{spec}}$ it can turn out that one needs a more sophisticated replacement for (A3), as otherwise the abstraction becomes too crude. One possibility here is to extend Q by an additional component representing a binary "don't change the low-level controller" flag plus some reference to the recent iteration X_i. Assuming a global upper bound on the time required for approaching $\mathcal{E}_{c'}$, this will not increase the order of Q.

It is an immediate consequence from our construction that the induced behaviour $\mathfrak{B}_{\mathrm{ca}}$ is a superset of $\mathfrak{B}$ and thus a suitable basis for supervisory controller synthesis.

5 The linear case

In this section we assume both the physical plant and the low level controllers to be modelled by linear time invariant differential equations. The continuous feedback system (3) then evolves according to a linear differential equation

$$\dot{x}(t) = F_c(x(t)) = A_c\, x(t) + b_c\,, \tag{20}$$

Here, efficient implementations of the approximation operators $\mathcal{R}_{c',t}$ and $\mathcal{S}_{c',\Delta}$ are available for ellipsoidal and polyhedral cells $\mathcal{Z}_{c,j}$; e.g. [KV00, AB+00].

The considered elementary manoeuvres implemented by the low-level controllers are characterised by

$$\lim_{t\to\infty} C_c\, x(t) + d_c = 0 \quad \text{for all } x(0)\,, \tag{21}$$

$$C_c\, x(0) + d_c = 0 \quad \Rightarrow \quad C_c\, x(t) + d_c \equiv 0\,, \tag{22}$$

where $C_c \in \mathbb{R}^{m\times n}$, $\mathrm{Rk}(C_c) = m$, $d_c \in \mathbb{R}^m$. That is, the continuous state approaches the affine subspace $\mathcal{E}_c = \{\xi \,|\, C_c\, \xi + d_c = 0\} \subseteq \mathbb{R}^n$, which itself is invariant under the flow $\Phi_c(\cdot\,,\, t)$ for all t. In order to separately discuss motion parallel to $\mathcal{E}_c$ and motion towards to $\mathcal{E}_c$, we transform state coordinates by

$$\begin{bmatrix} w(t) \\ z(t) \end{bmatrix} \coloneqq T_c(x(t)) \coloneqq \begin{bmatrix} \tilde{C}_c \\ C_c \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ d_c \end{bmatrix}, \tag{23}$$

where the rows of $\tilde{C}_c \in \mathbb{R}^{(n-m)\times n}$ are chosen as a basis of the null-space $\mathrm{Kern}(C_c)$. Note that T_c is bijective by construction. As a consequence of (22) observe $\dot{z}(t) \equiv 0$ for any trajectory $x(t)$ with $z(0) = 0$. In particular, this implies $\dot{z}(0) = 0$ whenever $z(0) = 0$ and for all $w(0)$. Thus, equation (20) can be rewritten in the form

$$\dot{w}(t) = A_{c,11}\, w(t) + A_{c,12}\, z(t) + b_{c,1}\,, \tag{24}$$

$$\dot{z}(t) = A_{c,22}\, z(t)\,, \tag{25}$$

where $A_{c,11} \in \mathbb{R}^{(n-m)\times(n-m)}$, $A_{c,12} \in \mathbb{R}^{(n-m)\times m}$, $A_{c,22} \in \mathbb{R}^{m\times m}$, $b_{c,1} \in \mathbb{R}^{n-m}$ and $A_{c,22}$ is exponential stable, i.e., $\sigma(A_{c,22}) \in \mathbb{C}^-$.

Equation (24) suggets that the *reduced model*

$$\dot{\tilde{w}}(t) = A_{c,11}\, \tilde{w}(t) + b_{c,1}\,, \tag{26}$$

with $\tilde{w}(t) \in \mathbb{R}^{n-m}$, describes the long time behaviour of (20). Thus, the reduced model promises a computational benefit for the approximate integration of cells of states. However, while any solution $x(t)$ approaches $\mathcal{E}_c$, the projection $w(t)$ itself is in general *not* a solution of the reduced model (26). We will show by construction that for every $x(t)$ there exists an initial condition $\tilde{w}(0)$ such that

$$\lim_{t\to\infty} ||w(t) - \tilde{w}(t)|| = 0\,. \tag{27}$$

Observe that the error $e(t) = w(t) - \tilde{w}(t)$ is a solution of

$$\dot{e}(t) = A_{c,11}\, e(t) + A_{c,12}\, z(t)\,. \tag{28}$$

Given $x(t)$, we need to show that there exists an initial condition $\tilde{w}(0)$, such that the column vector $\mathrm{Col}[e(0),\, z(0)] \in \mathbb{R}^n$ lies in the (exponential) stable subspace $E_s \subseteq \mathbb{R}^n$ of

$$A_{\mathrm{err}} = \begin{bmatrix} A_{c,11} & A_{c,12} \\ 0 & A_{c,22} \end{bmatrix}. \tag{29}$$

Note that $\sigma(A_{\mathrm{err}}) = \sigma(A_{c,11}) \cup \sigma(A_{c,22})$, where algebraic multiplicities of common eigenvalues add up. We now construct a basis of $\mathbb{R}^n$ consisting of generalised eigenvectors s_i, $1 \le i \le n$, of A_{err}. Pick any unstable $\lambda \in \sigma(A_{\mathrm{err}})$. As $A_{c,22}$ is stable, we have $\lambda \notin \sigma(A_{c,22})$ and thus $\lambda \in \sigma(A_{c,11})$ with the same algebraic multiplicity as it is w.r.t. A_{err}. Observe that $v \in \mathrm{Kern}((\lambda I - A_{c,11})^j)$ is equivalent to $\mathrm{Col}[v,\, 0] \in \mathrm{Kern}((\lambda I - A_{\mathrm{err}})^j)$ for all $j \in \mathbb{N}$, $v \in \mathbb{R}^{n-m}$. Thus, $s_1,\, s_2,\, \ldots\, s_k$ can be chosen to have the lower m entries zero, where $k \le n - m$ is the sum of the multiplicities of unstable eigenvalues in $\sigma(A_{\mathrm{err}})$. The remaining basis vectors $S = [s_{k+1},\, s_{k+2},\, \ldots\, s_n]$ then all refer to stable eigenvalues, and thus span the stable subspace; i.e., $E_s = \mathrm{Im}(S)$. As $s_1,\, s_2,\, \ldots\, s_n$ span the entire $\mathbb{R}^n$, and as the first k vectors have their lower m entries set to zero, we conclude $\mathrm{Im}([0,\, I_m]\, S) = \mathbb{R}^m$. This implies $z(0) = [0,\, I_m]\; S\; ([0,\, I_m]\; S)^{\#}\, z(0)$, where $S^{\#}$ denotes the Moore-Penrose inverse of S. By $e(0) \coloneqq [I_{n-m},\, 0]\, S\, ([0,\, I_m]\, S)^{\#}\, z(0)$ we give evidence of an initial condition $\tilde{w}(0) = w(0) - e(0)$ for which the error vanishes in the limit $t \to \infty$.

Starting the integration of the full model (20) at an initial state $x(0)$, we want to identify a time t such that the future evolution $x(\tau)$, $\tau \ge t$, can be approximated by integrating the reduced model. Thus, we are asking for an

upper bound on $||x(\tau) - \tilde{x}(\tau))||$ for all $t \geq \tau$, where $\tilde{x}(\tau) \coloneqq T_c^{-1}(\mathrm{Col}[I_{n-m},\ 0]\ \tilde{w}(\tau))$ and the initial value $\tilde{w}(0)$ is chosen according to the above considerations. Then, the error model only needs to be examined in its stable subspace and we transform to local coordinates of E_s:

$$r(t) = S^{\#} \begin{bmatrix} e(t) \\ z(t) \end{bmatrix}, \quad \dot{r}(t) = S^{\#}\, A_{\mathrm{err}}\, S\, r(t)\,. \qquad (30)$$

As $S^{\#} A_{\mathrm{err}} S$ is stable, there exists a positive definite matrix $P \in \mathbb{R}^{(n-k)\times(n-k)}$, such that $V(r) = r^{\intercal} P r$ is a Lyapunov function of (30). In particular $V(r(t))$ is strictly monotone decreasing. Therefore,

$$||x(\tau) - \tilde{x}(\tau)|| \ \leq\ ||\begin{bmatrix} \tilde{C}_c \\ C_c \end{bmatrix}^{-1} S\, P^{-\frac{1}{2}}||\ V(r(t))^{\frac{1}{2}}\,, \quad (31)$$

for all $t \geq \tau$. Now focus attention on a particular step $i < i^+$ in the context of our iterative integration (18)-(19). Note that the choice of $\tilde{w}(0)$ depends linearly on $x(0)$. Hence $x(0)$ is mapped by an affine map to $r(i\Delta)$. Assuming polyhedral (or ellipsoidal) cells, X_i can be transformed in "r-coordinates". The right hand side of (31) is convex and, in particular, maximisation over a polyhedral can be performed by evaluation at all vertices. If this maximum is below our target accuracy ρ we can turn to integrating the reduced model rather than the full model.

Conclusions

Hybrid control systems are frequently motivated by a high-level supervisory controller that switches between a finite number of continuous low-level controllers, all acting on the same physical plant. In this paper we propose how to exploit the hierarchical structure when formulating a discrete abstraction of the involved continuous dynamics. Using a grid partitioning of the n-dimensional state space, the number of discrete states in the abstraction depends exponentially on n. We focus on a general class of low-level control goals that is characterised by an m-dimensional stable component of the state variable. This enables us to effectively reduce the dimension of the state space to $n - m$. Computational advantage is then gained for two reasons: first, the lower dimensional grid consists of significantly less cells; second, the long term continuous dynamics can be approximated by a reduced model. This second aspect requires a detailed analysis of the continuous feedback loops, and we give such an analysis for the situation of linear time invariant differential equations. Our method is reliable in the sense that it is still guaranteed that the original system will only evolve on trajectories that are generated by the abstraction. This condition is crucial when employing the discrete abstraction as a basis for supervisory controller synthesis.

We have used our method for the design of a supervisory controller that drives the floating platform (15), (16) in a circular motion through four overlapping regions in $\mathbb{R}^2$. The number of cells in the discrete approximation could be reduced from 18×10^6 down to 35×10^3, with the obvious gain of computational performance. A detailed exposition of that case study must be omitted due to page limitations. Ongoing work addresses the start-up procedure of a distillation column, where the 42 dimensional state space model exhibits trajectories that –most of the time– evolve on a 3 dimensional manifold.

References

[AC+95] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, vol. 138, pp. 3–34, 1995.

[AB+00] E. Asarin, O. Bournez, T. Dang, O. Maler, A. Pnueli. Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE*, vol. 88:7, pp. 1011–1025, 2000.

[CW98] P.E. Caines, Y.J. Wei. Hierarchical hybrid control systems: a lattice theoretic formulation. *IEEE Transactions on Automatic Control*, vol. 43:4, pp. 501–508, 1998.

[FDF99] E. Frazzoli, M.A. Dahleh, E. Feron. A hybrid control architecture for aggressive maneuvering of autonomous helicopters. *Proceedings of the 38th IEEE Conference on Decision and Control CDC99*, pp. 2471-2476, Phoenix, 1999.

[FMR00] D. Franke, T. Moor, J. Raisch. Discrete supervisory control of switched linear systems. *at-Automatisierungstechnik*, pp. 461–467, vol. 48:9, 2000.

[KA+00] X. Koutsoukos, P.J. Antsaklis, J.A. Stiver, M.D. Lemmon. Supervisory control of hybrid systems. *Proceedings of the IEEE*, vol. 88:7, pp. 1026–1049, 2000.

[KV00] A.B. Kurzhanski, P. Varaiya. Ellipsoidal techniques for reachability analysis. In N. Lynch, B. Krogh (eds) *HSCC'00*, LNCS 1790, pp. 203–213, Springer, 2000.

[LTS99] J. Lygeros, C. Tomlin, S. Sastry. Controllers for Reachability Specifications for Hybrid Systems. *Automatica*, vol. 35, pp. 349–370, 1999.

[MR00] T. Moor, J. Raisch. Approximation of multiple switched flow systems for the purpose of control synthesis. *Proceedings of the 39th IEEE Conference on Decision and Control CDC00*, pp. 3604–3609, Sydney, 2000.

[MR99] T. Moor, J. Raisch. Supervisory control of hybrid systems within a behavioural framework, *System and Control Letters*, vol. 38:3, pp. 157–166, 1999.

[PLS00] G.J. Pappas, G. Lafferriere, S. Sastry. Hierarchically consistent control systems. *IEEE Transactions on Automatic Control*, vol. 45:6, pp. 1144-1160, 2000.

[RO98] J. Raisch, S.D. O'Young. Discrete approximation and supervisory control of continuous systems. *IEEE Transactions on Automatic Control*, vol. 43:4, pp. 569–573, 1998.

[RIM00] J. Raisch, A. Itigin, T. Moor. Hierarchical control of hybrid systems. *Proc. 4th International Conference on Automation of Mixed Processes: Dynamic Hybrid Systems*, pp. 67–72, Dortmund, Germany, Shaker Verlag 2000.

[RW87] P.J. Ramadge, W.M. Wonham. Modular Feedback Logic for Discrete Event Systems. *SIAM Journal on Control and Optimization*, vol. 25, pp. 1202–1218, 1987.

[Wi91] J.C. Willems. Paradigms and puzzles in the theory of dynamic systems. *IEEE Transactions on Automatic Control*, vol. 36:3, pp. 258–294, 1991.

[WW96] K.C. Wong, W.M. Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems*, vol. 6, pp. 241–306,1996.