Title

A Computationally Tractable Implementation of Pointwise Minimum Norm State-Feedback Laws for Hybrid Systems

Permalink
https://escholarship.org/uc/item/30t0k5x9

ISBN
9781467386821

Author
Sanfelice, Ricardo G

Publication Date
2016-07-01

DOI
10.1109/acc.2016.7525591

Peer reviewed

A Computationally Tractable Implementation of Pointwise Minimum Norm State-Feedback Laws for Hybrid Systems

Ricardo G. Sanfelice

Abstract—**We propose a computationally tractable implementation of state-feedback laws for hybrid systems given by differential equations capturing the continuous dynamics or *flows*, and by difference equations capturing the discrete dynamics or *jumps*. By exploiting the availability of a control Lyapunov function, along with other properties of the system, we show that pointwise minimum norm control laws can be implemented in a sample-and-hold fashion, with events triggered by timers, to render a desired compact set semiglobally and practically asymptotically stable. Examples illustrate the results.**

I. Introduction

We consider the problem of controlling hybrid plants, namely, dynamical systems with a state, denoted z, that flows according to

$$\dot{z} = F_P(z, u_c)$$

for a given input u_c and that jumps according to

$$z^+ = G_P(z, u_d)$$

for a given input u_d. Without yet making precise what a solution means and when flows or jumps occur, our goal is to design state-feedback laws for these systems that exploit the existence of a quantity V – a control Lyapunov function – that can be made to decrease along solutions, both during flows and jumps, by properly choosing the control inputs. More precisely, given a smooth enough function $z \mapsto V(z)$, design u_c so that during flows

$$\langle \nabla V(z), F_P(z, u_c) \rangle < 0$$

and design u_d so that at jumps

$$V(G_P(z, u_d)) - V(z) < 0$$

where by "< 0" we mean negative definiteness relative to a desired compact set that is to be rendered asymptotically stable. Note that due to the presence of variables such as timers, logic variables, and memory states, the origin is of little interest in stabilization of hybrid systems.

While theoretical results guaranteeing the existence of such feedback laws are informative [1], our goal in this paper is to devise a computationally tractable way[1] to choose the inputs u_c and u_d using feedback. To this end, we propose hybrid algorithms that are able to stabilize a class of hybrid systems by implementing event-driven mechanisms that sample the state of the plant, compute the control signal, and assign it to the inputs. More precisely, we show that, under the availability of a control Lyapunov function and certain properties of the hybrid system, a static state-feedback with pointwise minimum norm allows for a time-triggered, sample-and-hold type implementation that guarantees semiglobal practical stabilization of a desired compact set. The interest in implementations of pointwise minimum norm feedback laws is because such laws require the solution of an optimization problem, which can seldom be solved analytically or implemented in continuous time. We propose an implementation of such feedback laws that gives the optimization solver a finite amount of time to terminate. The implementation uses two timers, one to trigger the computations of the feedback laws and another to take samples of the state of the plant. In this time-triggered implementation, the events occur periodically and is essentially a sample-and-hold implementation of the feedback law. The proposed construction uses ideas from [3] for sample-and-hold of hybrid feedbacks for the control of continuous-time systems. The overall proposed approach of solving an optimization problem to compute the feedback laws controlling a hybrid system is also related to the receding horizon control approach, though in our setting, an expression of the feedback law is available. We are not aware of previous computational approaches for the control of the general class of hybrid systems modeled as in [4].

The remainder of the paper is organized as follows. After a brief section on preliminaries, Section III presents the proposed implementation and associated properties of the closed-loop system. The effect of discretization in the solutions to the closed-loop system and its robustness are discussed in Section III-A and Section III-B, respectively. Examples are given in Section IV.

Notation: $\mathbb{R}^n$ denotes n-dimensional Euclidean space, $\mathbb{R}$ denotes the real numbers. $\mathbb{R}_{\geq 0}$ denotes the nonnegative real numbers, i.e., $\mathbb{R}_{\geq 0} = [0, \infty)$. $\mathbb{N}$ denotes the natural numbers including 0, i.e., $\mathbb{N} = \{0, 1, \ldots\}$. $\mathbb{B}$ denotes the closed unit ball in a Euclidean space. Given a set K, $\overline{K}$ denotes its closure. Given a set S, ∂S denotes its boundary. Given $x \in \mathbb{R}^n$, $|x|$ denotes the Euclidean vector norm. Given a set $K \subset \mathbb{R}^n$ and $x \in \mathbb{R}^n$, $|x|_K := \inf_{y \in K} |x - y|$. Given x and y, $\langle x, y \rangle$ denotes their inner product and $[x^\top y^\top]^\top$ is equivalently represented by (x, y).s A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to class-$\mathcal{K}_\infty$ if it is continuous, zero at zero, strictly increasing, and unbounded. A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class-$\mathcal{KL}$ function, also written $\beta \in \mathcal{KL}$, if it is nondecreasing in its first argument, nonincreasing in its second argument, $\lim_{r \to 0^+} \beta(r, s) = 0$ for each $s \in \mathbb{R}_{\geq 0}$, and $\lim_{s \to \infty} \beta(r, s) = 0$ for each $r \in \mathbb{R}_{\geq 0}$. Given a closed set $K \subset \mathbb{R}^n \times \mathcal{U}_\star$ with $\star$ being either c or d and $\mathcal{U}_\star \subset \mathbb{R}^{m_\star}$, define $\Pi(K) := \{x : \exists u_\star \in \mathcal{U}_\star \text{ s.t. } (x, u_\star) \in K\}$ and $\Psi(x, K) := \{u : (x, u) \in K\}$. That is, given a set K, $\Pi(K)$ denotes the "projection" of K onto $\mathbb{R}^n$ while,

R. G. Sanfelice is with the Department of Computer Engineering, University of California, Santa Cruz, CA 95064, USA. Email: `ricardo@ucsc.edu`. Research partially supported by NSF Grants no. ECS-1150306 and CNS-1544396, and by AFOSR Grant and FA9550-16-1-0015.

[1] By *computationally tractability* of an algorithm, we mean that a computer can calculate the outcome of the algorithm in a reasonable amount of time; see [2, Chapter 8].

given x, $\Psi(x,K)$ denotes the set of values u such that $(x,u) \in K$. Then, for each $x \in \mathbb{R}^n$, define the set-valued maps $\Psi_c : \mathbb{R}^n \rightrightarrows \mathcal{U}_c$, $\Psi_d : \mathbb{R}^n \rightrightarrows \mathcal{U}_d$ as $\Psi_c(x) := \Psi(x, C)$ and $\Psi_d(x) := \Psi(x, D)$, respectively. Given a map f, its graph is denoted by $\mathrm{gph}(f)$.

II. Preliminaries

In this section, we recall some concepts and results from [4], [5].

A. Hybrid plant, controller, and closed-loop system

We consider plants modeled as a hybrid system with state denoted $z \in \mathbb{R}^{n_P}$ and dynamics

$$\mathcal{H}_P \begin{cases} \dot{z} & = & F_P(z, u_c) & \quad (z, u_c) \in C_P \\ z^+ & = & G_P(z, u_d) & \quad (z, u_d) \in D_P \end{cases} \tag{1}$$

where (C_P, F_P, D_P, G_P) is the data. Specifically, the set $C_P \subset \mathbb{R}^{n_P} \times \mathcal{U}_c$ is the *flow set*, the map $F_P : \mathbb{R}^{n_P} \times \mathbb{R}^{m_c} \rightrightarrows \mathbb{R}^{n_P}$ is the *flow map*, the set $D_P \subset \mathbb{R}^{n_P} \times \mathcal{U}_d$ is the *jump set*, and the map $G_P : \mathbb{R}^{n_P} \times \mathbb{R}^{m_d} \rightrightarrows \mathbb{R}^{n_P}$ is the *jump map*. The space for the state is $\mathbb{R}^{n_P}$ and the space for the input $u = (u_c, u_d)$ is $\mathcal{U} = \mathcal{U}_c \times \mathcal{U}_d$, where $\mathcal{U}_c \subset \mathbb{R}^{m_c}$ and $\mathcal{U}_d \subset \mathbb{R}^{m_d}$.

The state z of the hybrid system $\mathcal{H}_P$ can include multiple logic variables, timers, memory states as well as physical (continuous) states, e.g., $z = (q, \tau, \xi)$ is a state vector with a state component given by a logic variable q taking values from a discrete set $\mathcal{Q}$, a state component given by a timer τ taking values from the interval $[0, \tau^*]$, where $\tau^* > 0$ is the maximum allowed value for the timer, and with a state component $\xi \in \mathbb{R}^{n_c}$ representing the continuously varying state – note that in such a case, $\mathcal{Q} \times [0, \tau^*] \times \mathbb{R}^{n_c}$ can be embedded in $\mathbb{R}^{n_P}$ for $n_P = 1 + 1 + n_c$.

We will design a hybrid controller $\mathcal{H}_K$ with the same structure as the hybrid plant, but with state denoted ζ and the jump map being set valued. When controlling a hybrid plant, the feedback interconnection will result in a hybrid system without inputs. The model of this closed-loop system is given in the next section.

The hybrid system $\mathcal{H}_P$ under the effect of the hybrid controller $\mathcal{H}_K$ leads to a closed-loop system given by a hybrid system, which we denote $\mathcal{H}_{cl}$.

Solutions to a hybrid systems $\mathcal{H}$ are given in terms of hybrid arcs and hybrid inputs on hybrid time domains; see [4].

For a hybrid system $\mathcal{H}$, a compact set $\mathcal{A}$ is said to be pre-asymptotically stable if for each $\varepsilon > 0$ there exists $\delta > 0$ such that each maximal solution ϕ starting from $\mathcal{A} + \delta\mathbb{B}$ satisfies $\phi(t,j) \in \mathcal{A} + \varepsilon\mathbb{B}$ for each $(t,j) \in \operatorname{dom}\phi$, and there exists $\mu > 0$ such that each maximal solution starting from $\mathcal{A} + \mu\mathbb{B}$ is bounded and the complete ones satisfy $\lim_{t+j\to\infty} |\phi(t,j)|_{\mathcal{A}} = 0$. When the attractivity property holds for any $\mu > 0$, we say that $\mathcal{A}$ is globally pre-asymptotically stable.

B. Well-posedness

We will require the hybrid plant and the resulting closed-loop system $\mathcal{H}_{cl}$ to satisfy the following mild properties. We state these conditions for a general hybrid system $\mathcal{H}$ with data (C, F, D, G), which reduce to conditions for the closed-loop system when the inputs are removed.

Definition 2.1 (hybrid basic conditions): A hybrid system $\mathcal{H}$ is said to satisfy the *hybrid basic conditions* if its data (C, F, D, G) is such that

(A1) C and D are closed subsets of $\mathbb{R}^n \times \mathcal{U}_c$ and $\mathbb{R}^n \times \mathcal{U}_d$, respectively;

(A2) $F : \mathbb{R}^n \times \mathbb{R}^{m_c} \rightrightarrows \mathbb{R}^n$ is outer semicontinuous relative to C and locally bounded, and for all $(x, u_c) \in C$, $F(x, u_c)$ is convex and nonempty.

(A3) $G : \mathbb{R}^n \times \mathbb{R}^{m_d} \rightrightarrows \mathbb{R}^n$ is outer semicontinuous relative to D and locally bounded, and for all $(x, u_d) \in D$, $G(x, u_d)$ is nonempty.

Since the flow map and the jump map of the hybrid plant $\mathcal{H}_P$ are single valued, F_P and G_P satisfy (A2) and (A3), respectively, when they are continuous.

C. Control Lyapunov functions and pointwise min-norm feedback

Next, we recall the concept of control Lyapunov function for hybrid systems; see [1] for more details.

Definition 2.2 (control Lyapunov function): Given a compact set $\mathcal{A} \subset \mathbb{R}^{n_P}$ and sets $\mathcal{U}_c \subset \mathbb{R}^{m_c}$,$\mathcal{U}_d \subset \mathbb{R}^{m_d}$, a continuous function $V : \mathbb{R}^{n_P} \to \mathbb{R}$, continuously differentiable[2] on an open set containing $\overline{\Pi(C)}$ is a *control Lyapunov function with $\mathcal{U}$ controls for the hybrid plant $\mathcal{H}_P = (C_P, F_P, D_P, G_P)$* if there exist $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ and a positive definite function α_3 such that

$$\alpha_1(|z|_{\mathcal{A}}) \quad \leq \quad V(z) \quad \leq \quad \alpha_2(|z|_{\mathcal{A}}) \tag{2}$$
$$\forall z \in \Pi(C_P) \cup \Pi(D_P) \cup G_P(D_P),$$

$$\inf_{u_c \in \Psi_c(z)} \langle \nabla V(z), F_P(z, u_c) \rangle \leq -\alpha_3(|z|_{\mathcal{A}}) \tag{3}$$
$$\forall z \in \Pi(C_P),$$

$$\inf_{u_d \in \Psi_d(z)} V(G_P(z, u_d)) - V(z) \leq -\alpha_3(|z|_{\mathcal{A}}) \tag{4}$$
$$\forall z \in \Pi(D_P).$$

Given a hybrid system $\mathcal{H}$ satisfying the hybrid basic conditions, a compact set $\mathcal{A}$, and a control Lyapunov function V satisfying Definition 2.2, define, for each $r \in \mathbb{R}_{\geq 0}$, the set

$$\mathcal{I}(r) := \{ z \in \mathbb{R}^{n_P} \ : \ V(z) \geq r \}.$$

Moreover, for each $(z, u_c) \in \mathbb{R}^{n_P} \times \mathbb{R}^{m_c}$ and $r \in \mathbb{R}_{\geq 0}$, define the function

$$\Gamma_c(z, u_c, r) \quad := \quad \begin{cases} \langle \nabla V(z), F_P(z, u_c) \rangle + \dfrac{1}{2}\alpha_3(|z|_{\mathcal{A}}) \\ \qquad \text{if } (z, u_c) \in C_P \cap (\mathcal{I}(r) \times \mathbb{R}^{m_c}), \\ -\infty \qquad\qquad \text{otherwise} \end{cases}$$

and, for each $(z, u_d) \in \mathbb{R}^{n_P} \times \mathbb{R}^{m_d}$ and $r \in \mathbb{R}_{\geq 0}$, the function

$$\Gamma_d(z, u_d, r) \quad := \quad \begin{cases} V(G_P(z, u_d)) - V(z) + \dfrac{1}{2}\alpha_3(|z|_{\mathcal{A}}) \\ \qquad \text{if } (z, u_d) \in D_P \cap (\mathcal{I}(r) \times \mathbb{R}^{m_d}), \\ -\infty \qquad\qquad \text{otherwise.} \end{cases}$$

[2] The locally Lipschitz case can be treated similarly using the (Clarke) generalized directional derivative.

Then, respectively, evaluate the functions Γ_c and Γ_d at points (z, u_c, r) and (z, u_d, r) where $r = V(z)$ to define the functions

$$\begin{aligned}(z, u_c) &\mapsto \Upsilon_c(z, u_c) := \Gamma_c(z, u_c, V(z)),\\ (z, u_d) &\mapsto \Upsilon_d(z, u_d) := \Gamma_d(z, u_d, V(z))\end{aligned} \tag{5}$$

and the set-valued maps

$$\begin{aligned}\mathcal{T}_c(z) &:= \Psi_c(z) \cap \{ u_c \in \mathcal{U}_c \ : \ \Upsilon_c(z, u_c) \leq 0 \},\\ \mathcal{T}_d(z) &:= \Psi_d(z) \cap \{ u_d \in \mathcal{U}_d \ : \ \Upsilon_d(z, u_d) \leq 0 \}.\end{aligned} \tag{6}$$

In [5], we established that pointwise minimum norm control laws can be designed to pre-asymptotically stabilize the compact set

$$\mathcal{A}_r := \{ z \in \mathbb{R}^{n_P} \ : \ V(z) \leq r \} \tag{7}$$

In fact, under appropriate assumptions, the feedback laws

$$\rho_c(z) \ := \ \arg\min \{ |u_c| \ : \ u_c \in \mathcal{T}_c(z) \} \tag{8}$$

$$\rho_d(z) \ := \ \arg\min \{ |u_d| \ : \ u_d \in \mathcal{T}_d(z) \} \tag{9}$$

guarantee a pre-asymptotic stability property of $\mathcal{A}_r$. When certain conditions hold for each $r > 0$, then the feedback pair (ρ_c, ρ_d) is continuous and renders $\mathcal{A}_r$ pre-asymptotically stable for a restriction of $\mathcal{H}_P$ to $\mathcal{I}(r)$, while when further conditions hold, the feedback pair is also continuous and renders $\mathcal{A}_r$ pre-asymptotically stable, even with $r = 0$.

In the remainder of this paper, we will assume that the feedback pair (ρ_c, ρ_d) in (8)-(9) is continuous and renders a given compact set $\mathcal{A} \subset \mathbb{R}^{n_P}$ globally pre-asymptotically stable.

III. Implementation of Pointwise Minimum-norm Control Laws

We propose an implementation of the feedback pair (ρ_c, ρ_d) in (8)-(9) using timers and memory states, which resembles the sample-and-hold paradigm. Let $\tau_s \in [0, T_s]$ and $\tau_u \in [0, T_u]$ be timers, $\ell_s \in \mathbb{R}^{n_P}$ be a memory state storing a sample of the state z, $\ell_c \in \mathbb{R}^{m_c}$ be a memory state storing the value of the flow control law ρ_c, and $\ell_d \in \mathbb{R}^{m_d}$ be a memory state storing the value of the jump control law ρ_d, respectively. The parameters T_u and T_s determine the time elapsed between updates of the memory variables and reset of the timers. These variables are used to implement the following logic:

- At every T_s units of time, store the value of the state z in ℓ_s and compute the control laws;
- At every T_u units of time:
  - Update ℓ_c to the computed control law ρ_c if the stored value of the state and of the control law for flows are nearby the flow set;
  - Update ℓ_d to the computed control law ρ_d if the stored value of the state and of the control law for jumps are nearby the jump set;

Since the samples of z as well as the stored values of ρ_c and ρ_d may not always belong to the regions where these functions are defined, we treat ρ_c and ρ_d as set-valued maps that are empty outside of these regions and use continuous functions $\delta_c : \mathbb{R}^{n_P} \times \mathbb{R}^{n_P} \to [0, \delta]$ and $\delta_d : \mathbb{R}^{n_P} \to [0, \delta]$, $\delta > 0$, to determine how far from these regions the state and the memorized values of the feedbacks can be while still allowing the closed-loop system to evolve. More precisely, updates of ℓ_c will occur when $\tau_u = 0$ and

$$(\ell_s + \delta_c(\ell_s, \ell_c)\mathbb{B}, \ell_c + \delta_c(\ell_s, \ell_c)\mathbb{B}) \cap C_P \neq \emptyset \tag{10}$$

in which case ℓ_c will be updated to a point in[3]

$$\begin{aligned}\mathcal{K}_c(\ell_s, \ell_c) \ := \ \{ \ell'_c \ : \ & \ell'_c = \rho_c(\ell'_s), \ell'_s \in \ell_s + \delta_c(\ell_s, \ell_c)\mathbb{B},\\ & (\ell'_s, \ell_c + \delta_c(\ell_s, \ell_c)\mathbb{B}) \cap C_P \neq \emptyset \}\end{aligned} \tag{11}$$

Note that (10) holds when (ℓ_s, ℓ_c) is $\delta_c(\ell_s, \ell_c)$ close to points in C_P. Though $\mathcal{K}_c$ collects all values of ρ_c around those points, one only needs to compute one such value of ρ_c at one such point. Moreover, one would typically pick the function δ_c to be zero in C_P, at which points $\mathcal{K}_c$ would be single valued, and nonzero outside of C_P, so as to permit the solutions to continue evolving nearby C_P; see Section III-A for more details on this issue.

Similarly, ℓ_d will be updated when $\tau_u = 0$ and $(\ell_s + \delta_d(\ell_s, \ell_d)\mathbb{B}, \ell_d + \delta_d(\ell_s, \ell_d)\mathbb{B}) \cap D_P \neq \emptyset$, in which case ℓ_d will be updated to a point in

$$\begin{aligned}\mathcal{K}_d(\ell_s, \ell_d) \ := \ \{ \ell'_d \ : \ & \ell'_d = \rho_d(\ell'_s), \ell'_s \in \ell_s + \delta_d(\ell_s, \ell_d)\mathbb{B},\\ & (\ell'_s, \ell_d + \delta_d(\ell_s, \ell_d)\mathbb{B}) \cap D_P \neq \emptyset \}\end{aligned} \tag{12}$$

Combining the above constructions, the state of the controller implementing this logic is denoted as $\zeta = (\tau_s, \tau_u, \ell_s, \ell_c, \ell_d) \in [0, T_s] \times [0, T_u] \times \mathbb{R}^n \times \mathbb{R}^{m_c} \times \mathbb{R}^{m_d} =: O_K$ and its input is denoted $v \in \mathbb{R}^{n_P}$. The controller $\mathcal{H}_K = (C_K, F_K, D_K, G_K, \kappa_K)$ is given by the following data:

- Flow set:

$$C_K := \{ \zeta \in O_K \ : \ \tau_s \in [0, T_s], \tau_u \in [0, T_u] \}$$

- Flow map:

$$F_K(\zeta) := (1, 1, 0, 0, 0) \qquad \forall \zeta \in C_K$$

- Jump set:

$$D_K := D_{K,s} \cup D_{K,c} \cup D_{K,d}$$

where

$$D_{K,s} := \{ \zeta \in O_K \ : \ \tau_s = T_s \}$$

$$\begin{aligned}D_{K,c} := \{ \zeta \in O_K \ : & \tau_u = T_u,\\ & (\ell_s + \delta_c(\ell_s, \ell_c)\mathbb{B}, \ell_c + \delta_c(\ell_s, \ell_c)\mathbb{B}) \cap C_P \neq \emptyset \}\end{aligned}$$

and

$$\begin{aligned}D_{K,d} := \{ \zeta \in O_K \ : & \tau_u = T_u,\\ & (\ell_s + \delta_d(\ell_s, \ell_d)\mathbb{B}, \ell_d + \delta_d(\ell_s, \ell_d)\mathbb{B}) \cap D_P \neq \emptyset \}\end{aligned}$$

- Jump map:

$$G_K(\zeta, v) := G_{K,s}(\zeta, v) \cup G_{K,c}(\zeta) \cup G_{K,d}(\zeta)$$

where

$$G_{K,s}(\zeta, v) = (0, \tau_u, v, \ell_c, \ell_d)$$

for all $\zeta \in D_{K,s}$ and $v \in \mathbb{R}^{n_P}$, and empty everywhere else;

$$G_{K,c}(\zeta) = (\tau_s, 0, \ell_s, \mathcal{K}_c(\ell_s, \ell_c), \ell_d)$$

[3] An alternative approach is to project the states ℓ_s and ℓ_c to the flow set when nearby it.

for all $\zeta \in D_{K,c}$ and $v \in \mathbb{R}^{n_P}$, and empty everywhere else; and

$$G_{K,d}(\zeta) = (\tau_s, 0, \ell_s, \ell_c, \mathcal{K}_d(\ell_s, \ell_c))$$

for all $\zeta \in D_{K,c}$ and $v \in \mathbb{R}^{n_P}$, and empty everywhere else.

- Output map: $\zeta \mapsto \kappa_K(\zeta)$ given by $\kappa_K(\zeta) = (\ell_c, \ell_d)$.

The closed-loop system is defined by the assignments

$$u_c = \ell_c, \qquad u_d = \ell_d, \qquad v = z$$

and is given by the following hybrid system:

$$\mathcal{H}_{cl} \left\{ \begin{array}{rcll} \begin{bmatrix} \dot{z} \\ \dot{\zeta} \end{bmatrix} & = & \begin{bmatrix} F_P(z, \ell_c) \\ F_K(\zeta) \end{bmatrix} & (z, \ell_c) \in C_P, \zeta \in C_K, \\ \begin{bmatrix} z^+ \\ \zeta^+ \end{bmatrix} & = & \begin{bmatrix} G_P(z, \ell_d) \\ \zeta \end{bmatrix} & (z, \ell_d) \in D_P, \zeta \notin D_K, \\ \begin{bmatrix} z^+ \\ \zeta^+ \end{bmatrix} & \in & \begin{bmatrix} z \\ G_K(\zeta, z) \end{bmatrix} & (z, \ell_d) \notin D_P, \zeta \in D_K, \\ \begin{bmatrix} z^+ \\ \zeta^+ \end{bmatrix} & \in & \left\{ \begin{bmatrix} G_P(z, \ell_d) \\ \zeta \end{bmatrix}, \begin{bmatrix} z \\ G_K(\zeta, z) \end{bmatrix} \right\} & \\ & & & (z, \ell_d) \in D_P, \zeta \in D_K, \end{array} \right.$$

This closed-loop system model enforces flows when both flows of the plant and controller are possible, and allows jumps according to the three possible conditions triggering a jump: jumps due to being in the jump set of the plant but not of the controller (first difference equation), jumps due to being in the jump set of the controller but not of the plant (next difference inclusion), and jumps due to being in both the jump set of the plant and of the controller (last difference inclusion).

Theorem 3.1: Suppose that the hybrid plant $\mathcal{H}_P = (C_P, F_P, D_P, G_P)$ in (1) *satisfies the hybrid basic conditions and that $G_P(D_P) \cap \Pi(D_P) = \emptyset$. Let $\mathcal{A}, K \subset \mathbb{R}^{n_P}$ be compact sets, r and Δ be given positive numbers, and (ρ_c, ρ_d) be a globally pre-asymptotically stabilizing continuous feedback pair given as in* (8)-(9)*, where V is a control Lyapunov function for $\mathcal{H}_P$ with $\mathcal{U}$ controls, with associated compact set $\mathcal{A}$, functions $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$, and positive definite function α_3. Suppose that either*

1) *for some $\widetilde{\lambda} > 0$*

$$\widetilde{\lambda} V(z) \leq \alpha_3(|z|_{\mathcal{A}}) \qquad \forall z \in L_V(\Delta)$$

or

2) *for some $k > 0$*

$$\alpha_3 \circ \alpha_2^{-1}(s) \geq ks \qquad \forall s \geq 0$$

Then, for every $\varepsilon > 0$ there exist $\beta \in \mathcal{KL}$ and positive parameters T_s, T_u, and δ such that each solution ϕ to $\mathcal{H}_{cl}$ with z component ϕ_z, $\phi_z(0,0) \in K \cap L_V(\Delta)$, satisfies

$$|\phi_z(t,j)|_{\mathcal{A}_r} \leq \beta(|\phi_z(0,0))|_{\mathcal{A}_r}, t+j) + \varepsilon \qquad \forall (t,j) \in \operatorname{dom} \phi \tag{13}$$

Sketch of Proof: The closed-loop system satisfies the hybrid basic conditions – this follows by its construction and the fact that $\mathcal{H}_P$ satisfies the hybrid basic conditions and the feedback pair (ρ_c, ρ_d) is continuous. Let $r > 0$ and $\Delta > 0$ be given. Since $G_P(D_P) \cap \Pi(D_P) = \emptyset$ and $\mathcal{I}(r) \cap L_V(\Delta)$ is compact, by arguments similar to those in establishing [6, Lemma 2.7], there exist $\underline{T} > 0$ such that every solution ϕ to the closed-loop system that starts from $\mathcal{I}(r) \cap L_V(\Delta)$ has jump times t_j satisfying $t_{j+1} - t_j \geq \underline{T}$ for all $j \in \mathbb{N} \setminus \{0\}$.

Now, let $\lambda_s, \lambda_u > 0$ to be fixed later and define

$$W(z, \zeta) := \exp(\lambda_s \tau_s) \exp(\lambda_u \tau_u) V(z) \tag{14}$$

During flows of $\mathcal{H}_{cl}$, we have

$$\begin{aligned} \langle \nabla W(z, \zeta), \begin{bmatrix} F_P(z, \ell_c) \\ F_K(\zeta) \end{bmatrix} \rangle \;\; \leq \;\; & (\lambda_s + \lambda_u) W(z, \zeta) \\ & - \exp(\lambda_s \tau_s) \exp(\lambda_u \tau_u) \frac{1}{2} \alpha_3(|z|_{\mathcal{A}}) \\ & + \exp(\lambda_s \tau_s) \exp(\lambda_u \tau_u) \chi(z, e_1) \end{aligned}$$

where $e_1 := \ell_c - \rho_c(z)$ and

$$\chi(z, e_1) := |\nabla V(z)| \, |F_P(z, \rho_c(z) + e_1) - F_P(z, \rho_c(z))|$$

The function χ is continuous and vanishes with e_1. It follows that for each $\varepsilon > 0$, there exists $\delta_1 > 0$ such that

$$|\chi(z, e_1) - \chi(z, 0)| = \chi(z, e_1) \leq \varepsilon \qquad \forall e_1 \in \delta_1 \mathbb{B} \tag{15}$$

Then, there exists $\underline{\delta} > 0$ and $c \in (0,1)$ such that, on $\{z \in \mathbb{R}^{n_P} \; : \; \underline{\delta} \leq V(z) \leq \Delta \; \}$,

$$\chi(z, e_1) \leq c(\lambda_s + \lambda_u) V(z)$$

Using the upper bound on V, we get

$$\begin{aligned} \langle \nabla W(z, \zeta), \begin{bmatrix} F_P(z, \ell_c) \\ F_K(\zeta) \end{bmatrix} \rangle \;\; \leq \;\; & (1 + c)(\lambda_s + \lambda_u) W(z, \zeta) \\ & - \exp(\lambda_s \tau_s) \exp(\lambda_u \tau_u) \frac{1}{2} \alpha_3(|z|_{\mathcal{A}}) \end{aligned}$$

Then, using the assumptions on α_2, α_3, and V, and picking λ_s and λ_u positive and small enough, there exists $\lambda' > 0$ such that

$$\langle \nabla W(z, \zeta), \begin{bmatrix} F_P(z, \ell_c) \\ F_K(\zeta) \end{bmatrix} \rangle \;\; \leq \;\; -\lambda' W(z, \zeta)$$

At jumps, we have the following:

- If $(z, \ell_d) \in D_P, \zeta \notin D_K$,

  $$\begin{aligned} W(G_P(z, \ell_d), \zeta) - W(z, \zeta) \;\; = \\ \exp(\lambda_s \tau_s) \exp(\lambda_u \tau_u)(V(G_P(z, \ell_d)) - V(z)) \end{aligned}$$

  Using smoothness of the functions, we have that there exist positive δ_2 and δ_3 such that

  $$\begin{aligned} W(G_P(z, \rho_d(\ell_s)), \zeta) - W(z, \zeta) \;\; \leq \\ -\tfrac{1}{4} \exp(\lambda_s \tau_s) \exp(\lambda_u \tau_u) \alpha_3(|z|_{\mathcal{A}}) \end{aligned} \tag{16}$$

  for each (z, ℓ_s, ℓ_d) such that $|\ell_s - z| \leq \delta_2$ and $|\ell_d - \rho_d(\ell_s)| \leq \delta_3$.

- If $(z, \ell_d) \notin D_P, \zeta \in D_{K,c} \cup D_{K,d}$, for any $\rho_u \in [\exp(-\lambda_u T_u), 1)$, we have

  $$W(z, G_K(\zeta)) - W(z, \zeta) \;\; \leq \;\; -(1 - \rho_u) W(z, \zeta)$$

- If $(z, \ell_d) \notin D_P, \zeta \in D_{K,s}$, for any $\rho_s \in [\exp(-\lambda_s T_s), 1)$,

  $$W(z, G_K(\zeta)) - W(z, \zeta) \;\; \leq \;\; -(1 - \rho_s) W(z, \zeta)$$

- If $(z, \ell_d) \in D_P, \zeta \in D_K$, for each $\eta \in \left\{ \begin{bmatrix} G_P(z) \\ \zeta \end{bmatrix}, \begin{bmatrix} z \\ G_K(\zeta) \end{bmatrix} \right\}$, we have that, if $\eta =$

$(G_P(z),\zeta)$, then $W(\eta)-W(z,\zeta)$ is bounded as in (16), while if $\eta=(z,G_K(\zeta))$, from the above bounds, we have that

$$W(\eta)-W(z,\zeta) \quad \leq \quad -(1-\max\{\rho_s,\rho_u\})W(z,\eta)$$

for each (z,ℓ_s) such that $|\ell_s-z| \leq \delta_2$ and $|\ell_d-\rho_d(\ell_s)| \leq \delta_3$.

Then, combining the bounds above on the change of W and using the lower bound on V, there exists a positive definite function α_4 such that at jumps

$$W(\eta)-W(z,\zeta) \quad \leq \quad -\alpha_4(|z|_{\mathcal{A}})$$

for each

$$\eta \in \left\{ \begin{bmatrix} G_P(z) \\ \zeta \end{bmatrix}, \begin{bmatrix} z \\ G_K(\zeta) \end{bmatrix} \right\}$$

for each (z,ℓ_s) such that $|\ell_s-z| \leq \delta_2$ and $|\ell_d-\rho_d(\ell_s)| \leq \delta_3$.

The bounds during flows above are guaranteed to hold when

$$e_1=\ell_c-\rho_c(z), \qquad e_2:=\ell_s-z, \qquad e_3:=\ell_d-\rho_d(\ell_s)$$

are smaller than $\delta'':=\min\{\delta_1,\delta_2,\delta_3\}$. Under the stated assumptions, the following result establishes that for each given $\delta''>0$ the parameters T_s and T_u can be appropriately chosen so that for each maximal solution ϕ to the closed-loop system from $\mathcal{I}(r)\cap L_V(\Delta)$, there exists $(T^*,J^*)\in \operatorname{dom}\phi$ such that the trajectory $e:=(e_1,e_2,e_3)$ obtained from the solution ϕ, which we denote as $\phi_e=(\phi_{e_1},\phi_{e_2},\phi_{e_3})$, has components bounded by δ'' after (T^*,J^*) amount of hybrid time. The claim follows by combining this property with the properties of W stated above. □

A. On the Effect of Discretization on Maximal Solutions and their Completeness

In the proposed time-triggered implementation, we have that after a jump due to $\tau_u=T_u$, (z,ℓ_c^+) is in C_P or nearby it. Then, since $\dot{\ell}_c=0$, the state z may flow to a value from where, for the current held value of ℓ_c, the solution may not be able to further evolve due to the definitions of C_P and D_P (neither by flowing, because the state and input pair reaches a point in C_P from where flowing is not allowed, nor by jumping, because the state and input pair reaches a point that is not in D_P). Note that the property guaranteed by Theorem 3.1 is for maximal solutions of the closed-loop system, which are not necessarily complete. As typically done for hybrid systems, completeness of maximal solutions is a property that needs to be checked separately from the Lyapunov inequalities; see [4, Proposition 6.10] for a set of checkable conditions to assure that maximal solutions are complete. In some cases, these issues can be resolved by enlarging the sets using the functions δ_c,δ_d, augmenting the plant dynamics to points nearby C_P and D_P, and by exploiting the continuity of the CLF so that its properties hold in a larger region while sacrificing some of the negativity within the CLF bounds.

B. On the Robustness Properties of the Closed-loop System

The controller $\mathcal{H}_K$ in Section III is such that, if $\mathcal{H}_P$ satisfies the hybrid basic conditions, then the resulting closed-loop system also satisfies the hybrid basic conditions. This fact already implies that the closed-loop system has structural robustness properties to small perturbations. In particular, [4, Proposition 6.14] and [4, Proposition 6.34] assure that, over finite time horizons, small perturbations on the initial conditions and on the data of the closed-loop system change the behavior of the solutions only slightly. The latter kind of perturbations allows for uncertainty in the model of the hybrid plant and of the controller, such as unmodeled plant dynamics and uncertainty in the values of T_s and T_u, as well as the presence of small external disturbances, such as small noise in the samples of z and in the computation of the feedback laws; see [4, Definition 6.27] for more details.

IV. Numerical Examples

Example 4.1 (impact control of a pendulum): We illustrate our results in the control of a point-mass pendulum impacting on a controlled slanted surface. Denote the pendulum's angle (with respect to the vertical) by z_1 and the pendulum's velocity (positive when the pendulum rotates in the counterclockwise direction) by z_2. When $z_1 \geq \mu$ with μ denoting the angle of the surface, its continuous evolution is given by

$$\dot{z}_1=z_2, \quad \dot{z}_2=-a\sin z_1-bz_2+\tau,$$

where $a>0$, $b\geq 0$ capture the system constants (e.g., gravity, mass, length, and friction) and τ corresponds to torque actuation at the pendulum's end. For simplicity, we assume that $z_1\in[-\frac{\pi}{2},\frac{\pi}{2}]$ and $\mu\in[-\frac{\pi}{2},0]$. Impacts between the pendulum and the surface occur when

$$z_1 \leq \ \mu, \quad z_2 \leq 0.$$

At such events, the jump map takes the form

$$z_1^+=z_1+\widetilde{\rho}(\mu)z_1, \qquad z_2^+=-e(\mu)z_2,$$

where the functions $\widetilde{\rho}:[-\pi/2,0]\to(-1,0)$ and $e:[-\pi/2,0]\to[0,1)$ are continuous and capture the effect of pendulum compression and restitution at impacts, respectively, as a function of μ. As done in [5, Example 4.2], these hybrid dynamical system can be written as a hybrid plant $\mathcal{H}_P$ as in (1). Furthermore, with $\mathcal{A}=\{(0,0)\}$. the function

$$V(z)=z^\top Pz, \qquad P=\begin{bmatrix}2 & 1\\ 1 & 1\end{bmatrix}$$

is a CLF for $\mathcal{H}_P$.

For given $r>0$, a pointwise minimum norm control can be employed to asymptotically stabilize $\mathcal{A}_r$. For this purpose, following Theorem 3.1, we implement the controller $\mathcal{H}_K$ and the plant $\mathcal{H}_P$ in the Hybrid Equations Toolbox [7] to obtain the closed-loop simulations shown in Figure 1. [4] For the computation of the feedback laws via (11) and (12), we use `fmincon` to find the optimizers. The large planar plot shows that the resulting trajectory (blue with $\star$'s) is close to the optimal trajectory (green) – the $\star$'s indicate the instants when computations of ρ_c take place. The small snapshot in Figure 1

[4] Code at https://github.com/HybridSystemsLab/CLFPendComp

shows that the resulting trajectories approach the nominal one (solid, green) as the computation time parameter T_u gets smaller ($T_s = 0.1$ and $r = 0.0015$ in all simulations).

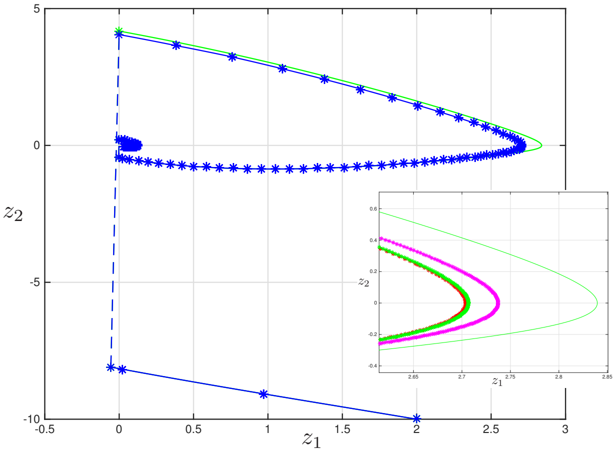
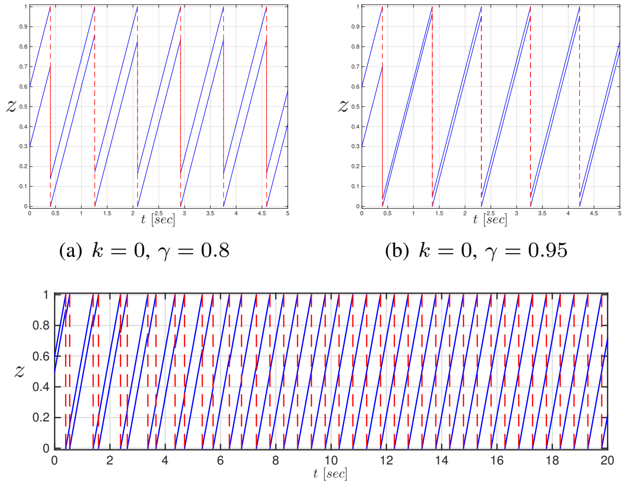


Fig. 1. Closed-loop trajectories (and zoom in) to the pendulum system on the plane starting from $z(0,0) = (2,-10)$ and evolving within $\{z \in \mathbb{R}^2 \ : \ V(z) \geq r\ \}$, $r = 0.0015$: ideal trajectory (solid, green), $T_u = 0.1$ ($\star$, blue), $T_u = 0.01$ ($\star$, red), $T_u = 0.005$ ($\star$, green), $T_u = 0.001$ ($\star$, magenta).

Example 4.2 (experimenting with resetting timers): Consider the hybrid system with state $z := (\tau_1, \tau_2) \in [0, \bar{\tau}] \times [0, \bar{\tau}] =: [0, \bar{\tau}]^2$, with τ_1, τ_2 being timer states with threshold $\bar{\tau} > 0$ and input u_1 and u_2. The state z evolves continuously according to the flow map $F_P(z) := (1, 1)$ when $z \in C_P := [0, \bar{\tau}]^2$. The state z jumps when any of the timers expires, namely, when $\max\{\tau_1, \tau_2\} = \bar{\tau}$. If τ_i expires, then it is reset to zero while τ_j is reset to u_j, where $i, j \in \{1, 2\}$, $i \neq j$. According to [8], a similar model can be used to capture the hybrid dynamics of spiking neurons for the study of synchronization and desynchronization – the models in [8] essentially make a specific state-dependent choice of the inputs according to the so-called phase response curve associated to the type of neuron. Motivated by such applications, we are interested in the asymptotic stabilization of the set $\mathcal{A} := \{z \in C_P \ : \ |\tau_1 - \tau_2| = k\ \}$, which, for $k = 0$, corresponds to the two timers asymptotically synchronizing and, for an appropriate $k > 0$, would correspond to the two timers being desynchronized. Without using the explicit constructions of the sets $\Psi_\star$ and the function α_3, we define the function

$$V(z) = \min\{|z_2 - z_1 + k|, |z_2 - z_1 - k|\}$$

and numerically experiment with the static state-feedback law (ρ_c, ρ_d) as in Section III by computing the minimizer of V after jumps. The range of possible valued for u_i at jumps are taken to be $z_i + \gamma\mathbb{B}$, where $\gamma > 0$ characterizes the size of input ranges allowed. From the initial condition $z(0,0) = (0.6, 0.3)$ and with $k = 0$, Figure 2(a) shows a solution[5] for $\gamma = 0.8$ and Figure 2(b) shows a solution for $\gamma = 0.95$. These plots indicate that solution components get closer to synchronization as the range of the inputs gets larger. Perfect synchronization can be achieved when $\gamma \geq 1$. Figure 2(c) shows a solution from $z(0,0) = (0.6, 0.5)$ with $k = 0.5$ and $\gamma = 0.05$. Even though the range of inputs is small, the feedback at jumps is capable of desynchronizing the solutions since small input values can steer the solution components to desynchronization.



(a) $k = 0$, $\gamma = 0.8$

(b) $k = 0$, $\gamma = 0.95$

(c) $k = 0.5$, $\gamma = 0.05$

Fig. 2. Control of two timers with resets for synchronization (in (a)) and for desynchronization (in (b) and (c)).

V. Conclusion

We established that the availability of a control Lyapunov function enables event-driven implementations of a state-feedback law with pointwise minimum norm for hybrid systems. Granted the optimization problem has a solution, which is guaranteed by the assumptions imposed in our results, the proposed implementation guarantees computational tractability as it allows a finite amount of time for computations to terminate. A particular challenge to the implementation is guaranteeing that maximal solutions to the closed loop are complete. Current research efforts focus on corollaries for special cases of the dynamics of the hybrid plants.

References

[1] R. G. Sanfelice. On the existence of control Lyapunov functions and state-feedback laws for hybrid systems. *IEEE Transactions on Automatic Control*, 58(12):3242–3248, December 2013.

[2] J. D. Hartline. Mechanism design and approximation. *Online, available at http://jasonhartline.com/MDnA/*, 2013.

[3] R. G. Sanfelice and A. R. Teel. Lyapunov analysis of sample-and-hold hybrid feedbacks. In *Proc. 45th IEEE Conference on Decision and Control*, pages 4879–4884, 2006.

[4] R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, New Jersey, 2012.

[5] R. G. Sanfelice. Pointwise minimum-norm control laws for hybrid systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2665–2670, 2013.

[6] R. G. Sanfelice, R. Goebel, and A. R. Teel. Invariance principles for hybrid systems with connections to detectability and asymptotic stability. *IEEE Transactions on Automatic Control*, 52(12):2282–2297, 2007.

[7] R. G. Sanfelice, D. A. Copp, and P. Nanez. A toolbox for simulation of hybrid systems in Matlab/Simulink: Hybrid Equations (HyEQ) Toolbox. In *Proceedings of Hybrid Systems: Computation and Control Conference*, pages 101–106, 2013.

[8] S. Phillips and R. G. Sanfelice. A framework for modeling and analysis of robust stability for spiking neurons. In *Proceedings of the 2014 American Control Conference*, pages 1414–1419, June 2014.

[5] Code at https://github.com/HybridSystemsLab/CLFTimersComp