UNIVERSITY OF CALIFORNIA
SANTA CRUZ

SAFETY AND ASYMPTOTIC STABILITY WHILE EXPLOITING UNCERTIFIED CONTROLLERS VIA UNITING FEEDBACK

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

APPLIED MATHEMATICS

by

Paul Kenna Wintz

December 2025

The dissertation of Paul Kenna Wintz is approved by:

Professor Ricardo Sanfelice, Chair

Professor Qi Gong

Professor Daniele Venturi

Associate Adjunct Professor Alessandro Pinto

Peter Biehl
Dean of Graduate Studies

Table of Contents

Abstract

Safety and Asymptotic Stability while Exploiting Uncertified Controllers via Uniting Feedback

by

Paul Kenna Wintz

Certificate functions, such as barrier functions and Lyapunov functions, are commonly used to verify control system properties. The construction of these certificates, however, is often difficult, typically requiring significant trial and error. Once a certificate function is found, modifications to the controller are hindered because each change requires the construction of a new certificate function. This problem is addressed in this dissertation by the design of uniting feedback strategies that allow uncertified controllers to be safely used by exploiting a controller with a known certificate as a backup. In uniting feedback, an automatic supervisor switches between two controllers. The result is a hybrid control strategy that switches between certified and uncertified controllers while preserving the safety or asymptotic property that is guaranteed for the certified controller. By using a certified controller as a backup, these uniting feedback strategies allow for exploiting uncertifiable controllers that may have other desirable properties. A general framework is developed that allows for the design of supervisors for systems with both the controllers and the plant modeled as hybrid dynamical systems with set-valued dynamics, while ensuring the closed-loop system is well-posed and the switching does not occur too often.

Several auxiliary tools and results are also included. A hybrid Lyapunov theorem is presented that relaxes several key assumptions in prior hybrid Lyapunov theorems. These relaxations make it easier to construct Lyapunov certificates and are used to prove results in this dissertation. Additionally, the conical transition graph is presented as a tool for algorithmically checking stability in conical hybrid systems, guiding the search for Lyapunov functions or identifying when such a search is futile. Finally, the Simulator for Hardware Architecture and Real-time Control (Sharc) is a simulation tool for verifying the performance of computationally delayed control systems, providing a useful testing platform for verifying uniting feedback strategies when deployed on systems with limited computational resources.

To Mom and Dad

Although you didn't teach me to love math,
you did teach me to love learning and to love people.

That has made all of the difference.

In Memory of Nitesh Kumar Singh
One of the kindest people I have known.

Acknowledgments

This dissertation is the product of support, mentorship, and instruction from countless (but not uncountable) teachers, coaches, friends, and family who helped me along the way. From my Kindergarten teacher, Mrs. Johnston, to the all my professors at UCSC, I've learned important lessons from each of you. I am especially thankful to my parents, Loa and Brian Wintz, for homeschooling me from 1st grade until high school graduation and giving me constant support through college and grad school; to Professor Anna Parmely at Santa Barbara City College for showing me that I can enjoy and succeed in mathematical studies; and to Professor Michael Sommermann for helping me to think rigorously and communicate clearly about math and physics. The work in this dissertation was done under the ever patient guidance of Professor Ricardo Sanfelice. If I had listened to all of his advice, I would have finished much sooner.

Finally, I want to thank all of my friends in Santa Cruz and beyond for making grad school a time I will remember fondly, including Austin Chan, Caitlin Voorhees, David Kooi, Delaney Newhouse, Dev Purandare, Isabel Kain, Mike Briden, Peter Braun, Sean Riedel, Sofie Tuyls-Eckert, and many others. I am deeply grateful to my friends Ryan Johnson, Callie Chappell, Adam Grant, and my sister, Mary Alice Wintz, for their help preparing for my defense, and their persistent encouragement, advice, and kindness throughout the years.

Chapter 1

Introduction

This dissertation considers the design of controllers for control systems to produce desired properties in the closed-loop system. We consider a broad class of control systems and controllers, but as an initial illustration consider a continuous-time control system with state $z \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ that we write as an ordinary differential equation (ODE) with an input:

$$\dot{z} = f(z, u),$$

where values of u are generated by the controller and $\dot{z} := dz/dt$. A static state feedback controller assigns the input as a function of the current state, e.g., $u(t) = \kappa(z(t))$, where $\kappa : \mathbb{R}^n \to \mathbb{R}^m$. Such a controller does not depend on the past, so it is called *memoryless*. For historical reasons, the system that is to be controlled is called the *plant* (derived from the early application of control engineering to chemical plants). Aside from modeling the evolution of a plant continuous-time as an ODEs, the dynamics can also be given in discrete time as difference equations, or in both as a combination of continuous- and discrete-time, which we call hybrid systems. The class of control systems can be further broadened by allowing for set-valued dynamics, which in the case of a continuous-time system results in a *differential inclusion*,

$$\dot{z} \in F(z, u)$$

where F is a set-valued map that describes a set of possible values of the derivative $\dot{z}$. Generally, solutions are not unique for systems with set-valued dynamics, which makes them suitable for modeling uncertainty or non-deterministic control systems.

Similarly, we can also consider a broader class of controllers beyond static feedback. In fact, controllers can also have internal states that evolve according to

some dynamics. Returning to the context of continuous-time systems, as *dynamic feedback* controller with state $\eta \in \mathbb{R}^p$ can be written as

$$\begin{cases} \dot{\eta} = f_c(z, \eta) \\ u = \kappa(z, \eta), \end{cases}$$

where z and u are the state and input of the plant, as before. As with plants, the dynamic feedback controllers can have dynamics in continuous-, discrete-, or hybrid-time as well as be set valued.

There are two of the fundamental goals in the design of control systems: i) Move the system to a target. ii) Avoid obstacles and any unsafe or prohibited states. The first goal, target-reaching, can be stated mathematically as convergence of the system's state to a set in its configuration space. Typically, one wishes to design a system that can converge to the target from any point in some region around it. We call this property asymptotic stability. If the system converges to the target from every initial state (within some domain of consideration), then we say that the target is globally asymptotically stable. Thus, the first goal can be restated as "design a controller that renders the target set to be globally asymptotically stable."

The second goal, called *constraint-satisfaction* or *safety*, is described mathematically by the concept of forward invariance. Informally, a set of configurations in a dynamical system's state space is called forward invariant if every trajectory that starts inside the set will always remain inside the set as it evolves according to the system dynamics.[1] When designing a safe controller, the particular goal is to produce a forward invariant set that is fully within the constraints and contains all possible initial system states.

The process of achieving the two goals has parts relevant to this work, namely controller synthesis and formal verification.[2] Although I will refer to these as "steps,"

[1] Trajectories may, however, enter a forward invariant set. The "forward" indicates that solutions do not leave the set time progresses forward. In contrast, *backward invariance* would indicate that solutions remain in the set as you consider trajectories moving backward in time, indicating that no trajectories move into the set.

[2] Other tasks are also involved. Before one can synthesize a controller, one typically creates or learns a model of the system, including finding numerical values for system parameters. Once a controller is designed, mathematically, the implementation onto a physical system also carries significant challenges. On a physical system, one must also estimate the state of the system using measurements and account for unmodeled disturbances and uncertainties.

their execution is not sequential, as they typically require multiple iterations. The synthesis step consists of selecting a controller that is intended to satisfy the system specifications. There are existing formulaic methods for controller synthesis for particular classes of control systems, such as linear control systems, but the task is generally non-trivial. The verification step consists of demonstrating that the goal is achieved. Although such verification can be preformed empirically, via statistical analysis of physical or simulated experiments, we are interested in formal methods of proving that a mathematical model of a control system satisfies given properties through deductive reasoning.

One prominent method for verification of asymptotic stability or safety of a closed-loop system is via the construction of a certificate function with certain properties, namely positive definiteness and monotonic rate of change along flows in certain regions. For stability properties, certificate functions are called *Lyapunov functions* and for safety, certificates are called *barrier functions* or *barrier certificates*. There are, in fact, many varieties of Lyapunov functions. Differences arise from the particular type of stability desired (Lyapunov stability, local/global asymptotic stability, finite-time stability, fixed-time stability, input-to-state stability, practical asymptotic stability), with a similar variety of theorems that apply for various combinations of system and type of stability. There are also several varieties of barrier functions, although fewer, with distinctions primarily arising from differences in the types of systems considered. In this work, we consider Lyapunov functions for certifying global asymptotic stability and barrier functions for safety with respect to hybrid dynamical systems, including hybrid systems with set-valued dynamics.

Constructing Lyapunov and barrier functions for nonlinear systems is generally a difficult problem. For each variety of Lyapunov function, there is condition that the function is positive definite with respect to the target set and one or more other conditions that are sufficient for ensuring that Lyapunov function is monotonically decreasing or nondecreasing for all trajectories the system can take. The process typically involves trial and error. Once a Lyapunov or barrier function is known, modifications to the controller is difficult because each change requires the construction of a new certificate function, limiting iteration. We address this problem in Chapters 2–4 by introducing uniting feedback strategies that allow uncertified controllers to be used by exploiting a controller with a known certificate as

a backup. The result is a hybrid control strategy that switches between the certified and uncertified controllers while preserving the safety or asymptotic property that is guaranteed for the certified controller. Our control strategy is designed to prevent chattering and ensures that the safety or asymptotic stability property is robust to small measurement noise.

In Chapters 3 and 4 we give a uniting feedback strategy for global asymptotic stability. To prove that the uniting feedback produces global asymptotic stability, we generate a new certificate function for the hybrid closed-loop system, which turns out to be a non-smooth function. Although there are existing theorems for nonsmooth Lyapunov functions in the context of hybrid systems, we found the assumptions too strict for what we needed. Thus, we developed a hybrid Lyapunov theorem with relaxed assumptions, which is presented in Chapter 5.

The application of our uniting feedback strategy requires that one must first construct a Lyapunov function. The construction of Lyapunov functions is generally difficult. In fact, is often unclear whether or not one's search for a Lyapunov function is futile. The search can be unsuccessful because the right function has not been tried or because the system is actually unstable. While constructing Lyapunov functions, it is not obvious during if failure to find a Lyapunov function is due to not yet trying the right function or if no such function exists because the system is unstable. Thus, it would be useful to have reliable methods for checking the target set is asymptotically stable or unstable before one starts searching for a Lyapunov function. For continuous- or discrete-time systems, stability of the origin can be immediately determined by examining the eigenvalues of the system's matrix. In Chapter 6, we investigate a method for evaluating stability in a class of hybrid systems that we call *conical hybrid systems* and is a reasonably close analog to linear systems for hybrid systems. We developed a tool called the conical transition graph (CTG) that can automatically determine stability for conical hybrid systems, including systems that switch between multiple modes.

One promising application of our uniting feedback strategies is for deployment of computationally demanding controllers on cyber-physical systems with limited computational resources. Suppose system designers want to deploy an advanced controller that is expected to perform well but is difficult to verify due to complex interactions between the system's trajectory, delays in updating the control values,

and the variable amount of time required to compute updates. By using a computationally inexpensive controller as the certified controller in the uniting feedback strategy, the designers can deploy advanced controllers without jeopardizing the system. In Chapter 7, we present a tool called SHARC for computing trajectories of cyber-physical systems that uses a cycle-accurate simulation of the system's computing hardware to incorporate the computational delays into the timing of the system's control updates.

1.1 Preliminaries

For notation, we use $\mathbb{N} := \{0, 1, 2, \dots\}$ and $\mathbb{R}_{\geq 0} := [0, \infty)$. The Euclidean norm of $x \in \mathbb{R}^n$ is written $|x|$ and the inner product between x and $y \in \mathbb{R}^n$ is written $\langle x, y \rangle$. The concatenation of vectors $x_1 \in \mathbb{R}^{n_1}$ and $x_2 \in \mathbb{R}^{n_2}$ is denoted $(x_1, x_2) := \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right] \in \mathbb{R}^{n_1+n_2}$. We write the unit ball in $\mathbb{R}^n$ as $\mathbb{B} := \{x \in \mathbb{R}^n : |x| \leq 1\}$. For a set $S \subset \mathbb{R}^n$, we denote the boundary as ∂S, the interior as $\text{int}(S)$, the closure as $\overline{S}$, and the convex hull as $\text{conv}(S)$. A *neighborhood* of S is any open set U such that $S \subset U$. The distance from x to S is $|x|_S := \inf_{y \in \mathcal{A}} |y - x|$. For any sets $S_1, S_2 \in \mathbb{R}^n$, we write $\text{dist}(S_1, S_2) := \inf\{|s_1 - s_2| : s_1 \in S_1, s_2 \in S_2\}$. Given a nonempty set $\mathcal{A} \subset \mathbb{R}^n$, the distance from any $x \in \mathbb{R}^n$ to $\mathcal{A}$ is

$$|x|_{\mathcal{A}} := \inf_{a \in \mathcal{A}} |a - x|.$$

We write the domain of a function f as $\text{dom}(f)$. "Continuously differentiable" is abbreviated as $\mathcal{C}^1$. If $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at $x \in \text{dom}(f)$, then the gradient of f at x is denoted $\nabla f(x)$. We say f is *lower semicontinuous* (LSC) if

$$f(x_0) \leq \liminf_{x \to x_0} f(x) \qquad \forall x_0 \in \text{dom}(f).$$

If f is LSC, then $g := -f$ is *upper semicontinuous* (USC).

A continuous function $\alpha : [0, c) \to \mathbb{R}_{\geq 0}$ is class–$\mathcal{K}$ if α is zero at zero and strictly increasing. A continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be in class–$\mathcal{K}_\infty$ if it is zero at zero, strictly increasing, and $\lim_{r \to \infty} \alpha(r) = \infty$. A function $\rho : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is *positive definite* if $\rho(0) = 0$ and $\rho(r) > 0$ for all $r > 0$. We write the set of all positive definite functions on $\mathbb{R}_{\geq 0}$ as $\mathcal{PD}(0)$. Given nonempty sets $\mathcal{A} \subset \mathbb{R}^n$ and $\mathcal{X} \subset \mathbb{R}^n$, a function $\sigma : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is said to be *positive definite on $\mathcal{X}$ with respect to $\mathcal{A}$* if

$\sigma(x) = 0$ for all $x \in \mathcal{A} \cap \mathcal{X}$ and $\sigma(x) > 0$ for all $x \in \mathcal{X} \setminus \mathcal{A}$. The set of all positive definite functions on $\mathcal{X} = \mathbb{R}^n$ with respect to $\mathcal{A}$ is denoted $\mathcal{PD}(\mathcal{A})$. A function f is said to be *negative definite* if $g := -f$ is positive definite.

The contingent cone [1] to a set S at x is denoted $T_S(x)$. For nonsmooth functions, we use the Clarke generalized gradient and Clarke generalized directional derivative [2]. For a locally Lipschitz function $V : \mathbb{R}^n \to \mathbb{R}$, the *Clarke generalized gradient* of V at $x \in \mathbb{R}^n$ is

$$\partial^\circ V(x) := \operatorname{conv}\left\{ \lim_{i\to\infty} \nabla V(x_i) \;\middle|\; \begin{array}{l} \exists (x_i \to x) \text{ s.t. } V(x_i) \text{ is} \\ \text{differentiable at each } x_i \end{array} \right\}. \tag{1.1}$$

The *Clarke generalized directional derivative* of V at $x \in \mathbb{R}^n$ in the direction $w \in \mathbb{R}^n$ is given by

$$V^\circ(x, w) = \max_{\zeta \in \partial^\circ V(x)} \langle \zeta,\, w \rangle. \tag{1.2}$$

A set-valued map $F : X \rightrightarrows Y$ maps each $x \in X$ to a set $F(x) \subset Y$. The domain of F is defined as $\operatorname{dom}(F) := \{x \in X \mid F(x) \neq \varnothing\}$. We say F is *outer semicontinuous* (OSC) [3, Def. A.32] if for each $x_0 \in \operatorname{dom}(F)$, each sequence $\{x_i\}_{i=1}^\infty$ in $\operatorname{dom}(F)$ converging to x_0, and each convergent sequence $\{y_i\}_{i=1}^\infty$ with each $y_i \in F(x_i)$, we have that

$$\lim_{i\to\infty} y_i \in F(x_0).$$

We say F is *locally bounded* if for each $x_0 \in \operatorname{dom}(F)$, there exists a neighborhood U of x_0 such that $F(U \cap \operatorname{dom} F)$ is bounded [3, Def. A.11].

1.1.1 Hybrid Systems

We consider hybrid systems on $\mathbb{R}^n$ written as

$$\mathcal{H} : \begin{cases} \dot{x} \;\in F(x) & x \in C \\ x^+ \in G(x) & x \in D \end{cases} \tag{1.3}$$

with state $x \in \mathbb{R}^n$, flow set $C \subset \mathbb{R}^n$, flow map $F : C \rightrightarrows \mathbb{R}^n$, jump set $D \subset \mathbb{R}^n$, and jump map $G : D \rightrightarrows \mathbb{R}^n$. We write $\mathcal{H}$ compactly as $\mathcal{H} = (C, F, D, G)$. The continuous-time system formed by removing the discrete dynamics of $\mathcal{H}$ is written as (C, f).

Solutions to hybrid systems are defined on hybrid time domains, which consist of pairs of values $(t, j) \in \mathbb{R} \times \mathbb{N}$ where t parameterizes the passage of ordinary time and j parameterizes the passage of discrete time, counted as the number of

discrete jumps that have occurred. More precisely, a *hybrid time domain* E is a subset of $\mathbb{R}_{\geq 0} \times \mathbb{N}$ such that there exists $J \in \mathbb{N} \cup \{\infty\}$ and a nondecreasing sequence $\{t_j\}_{j \in \{0,1,\ldots,J\}}$ with $t_0 := 0$ such that for every $(t^*, j^*) \in E$, the following holds:

$$E \cap \big([0,t^*] \times \{0,1,\ldots,j^*\}\big) = \big([t_0,t_1] \times \{0\}\big) \cup \big([t_1,t_2] \times \{1\}\big) \cup \cdots \cup \big([t_{j^*},t^*] \times \{j^*\}\big). \tag{1.4}$$

Each $t_1, t_2, \ldots, t_J$ is called a *jump time* in E. If $t_j < t_{j+1}$, then the interval $I_j := \{t \mid (t,j) \in \operatorname{dom}(\varphi)\}$ (which has nonempty interior) is called an *interval of flow* in E. We write

$$\begin{aligned} \sup_t E &:= \sup\{t \in \mathbb{R}_{\geq 0} \mid (t,j) \in E\}, \\ \sup_j E &:= \sup\{j \in \mathbb{N} \mid (t,j) \in E\}, \\ \operatorname{length}(E) &:= \sup_t E + \sup_j E. \end{aligned}$$

A function $\varphi : \operatorname{dom}(\varphi) \to \mathbb{R}^n$ is called a *hybrid arc* if $\operatorname{dom}(\varphi)$ is a hybrid time domain and φ is absolutely continuous on each interval of flow in $\operatorname{dom}(\varphi)$.[3] A hybrid arc φ is said to be *complete* if $\operatorname{length}(\operatorname{dom}\varphi) = \infty$. A hybrid arc φ is said be an *extension of* another hybrid arc ψ if $\operatorname{dom}(\psi)$ is a strict subset of $\operatorname{dom}(\varphi)$ and

$$\varphi(t,j) = \psi(t,j) \qquad \forall (t,j) \in \operatorname{dom}(\psi).$$

Definition 1.1 (Hybrid Solution)**.** A hybrid arc φ is called a *solution* of $\mathcal{H}$ if it satisfies the following:

- At each jump time t_j in $\operatorname{dom}(\varphi)$, the hybrid arc φ must satisfy

$$\varphi(t_j, j-1) \in D \tag{1.5a}$$

$$\varphi(t_j, j) \in G\big(\varphi(t_j, j-1)\big). \tag{1.5b}$$

- For each interval of flow I_j in $\operatorname{dom}(\varphi)$ (with $I_j = [t_j, t_{j+1}]$ or $I_j = [t_j, \infty)$ for some j),

$$\varphi(t,j) \in C \qquad \text{for all } t \in \operatorname{int} I_j \tag{1.6a}$$

$$\frac{d\varphi}{dt}(t,j) \in F(\varphi(t,j)) \qquad \text{for almost all } t \in I_j. \tag{1.6b}$$

[3] Absolute continuity is the weakest form of continuity that implies a function is differentiable almost everywhere, making it the weakest assumption suitable for the definition of solutions to differential inclusions.

A solution φ to $\mathcal{H}$ is said to be a *maximal solution* if it cannot be extended. That is, φ is maximal if there does not exist another solution ψ to $\mathcal{H}$ such that $\mathrm{dom}(\varphi)$ a strict subset of $\mathrm{dom}(\psi)$ and $\varphi(t,j) = \psi(t,j)$ for all $(t,j) \in \mathrm{dom}(\varphi)$. At a jump time t in a hybrid domain, t corresponds to several values of j, so it is useful to define a function that maps each t to a single value of j. In particular, for each $(t,j) \in \mathrm{dom}(\varphi)$, we define

$$t \mapsto \bar{\jmath}(t) := \max\{j \mid (t,j) \in \mathrm{dom}(\varphi)\}. \tag{1.7}$$

For more on hybrid systems, see [3, 4].

A hybrid system $\mathcal{H}$ is called *well-posed* if its set of solutions is sequentially compact, meaning that the limit of any graphically convergent sequence of solutions is also a solution. Well-posedness is useful for establishing properties such as robustness of asymptotic stability of compact sets. The following conditions [4, Assumption 6.5] are sufficient for a hybrid system to be well-posed.

Definition 1.2 (Hybrid Basic Conditions [3, Def. 2.20])**.** A hybrid system $\mathcal{H} = (C, F, D, G)$ on $\mathbb{R}^n$ as in (1.3) is said to satisfy the *hybrid basic conditions* if

(A1) C and D are closed;

(A2) $C \subset \mathrm{dom}(F)$, F is outer semicontinuous and locally bounded relative to C, and $F(x)$ is convex for each $x \in C$; and

(A3) $D \subset \mathrm{dom}(G)$, and G is outer semicontinuous and locally bounded relative to D. ◇

1.1.2 Notation for Sets and Set-valued Maps

In this work we use several notations that simplify writing about dynamical systems with set-valued maps. For sets $A \subset \mathbb{R}^n$ and $B \subset \mathbb{R}^m$, we write the Cartesian product using array notation, as shown here:

$$\begin{bmatrix} A \\ B \end{bmatrix} := A \times B.$$

We also use the same notation to construct a mixture of column vectors and sets, e.g., for $A \subset \mathbb{R}^n$ and $v \in \mathbb{R}^m$,

$$\begin{bmatrix} A \\ v \end{bmatrix} := A \times \{v\}.$$

For set-valued maps $F_1 : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ and $F_2 : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$, we define $F_1 \cap F_2 : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ as

$$(F_1 \cap F_2)(x) = F_1(x) \cap F_2(x) \qquad \forall x \in \mathbb{R}^n.$$

1.1.3 Set-valued Lie derivative

For a differential inclusion $\dot{x} \in F(x)$ with a set-valued map $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ and a Lipschitz continuous function $V : \mathbb{R}^n \to \mathbb{R}$, we define the set-value Lie derivative of V along F as

$$\mathcal{L}_F V(x) := \left\{ \langle \zeta,\, f \rangle \;\middle|\; \zeta \in \partial^\circ V(x),\ f \in F(x) \right\} \qquad \forall x \in \mathbb{R}^n.$$

The set-valuedness of $\mathcal{L}_F V(x)$ comes from two places:

1. The set-valued right-hand side of the differential inclusion $\dot{x} = F(x)$, and
2. the set-valued Clarke generalized gradient of V, which we use because V is only assumed to be Lipschitz—not necessarily differentiable.

Thus, even if F is single valued or, alternatively, V is smooth, the Lie Derivative of V along F would still be set-valued. For each $x_0 \in \mathbb{R}^n$, $\mathcal{L}_F V(x_0)$ is a set in $\mathbb{R}$, we can write the least upper bound as $\sup \mathcal{L}_F V(x_0)$, which gives an upper bound on the rate of change of V for solutions to $\dot{x} \in F(x)$ at x_0. This is essential to Lyapunov-like theorems for differential inclusions, as considered in the next section.

1.1.4 Stability Properties and Lyapunov Functions

We consider a variety of stability properties for hybrid systems, which we collect into two groups: local stability properties, given in Definition 1.3 and global stability properties, given in Definition 1.4. In both definitions, the prefix "pre-" indicates that these properties allow for maximal solutions that terminate after finite time (e.g., due to leaving $C \cup D$).

Definition 1.3. For a hybrid system $\mathcal{H}$ as in (1.3), a nonempty set $\mathcal{A} \subset \mathbb{R}^n$ is said to be

- *stable* for $\mathcal{H}$ if for all $\varepsilon > 0$, there exists $\delta > 0$ such that for every solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \leq \delta$, we have that $|\varphi(t,j)|_{\mathcal{A}} \leq \varepsilon$ for all $(t,j) \in \operatorname{dom}(\varphi)$; and
- *pre-attractive* for $\mathcal{H}$ if there exists $\mu > 0$ such that for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \leq \mu$, we have that $(t,j) \mapsto |\varphi(t,j)|_{\mathcal{A}}$ is bounded and, if φ is complete, then
$$\lim_{t+j\to\infty} |\varphi(t,j)|_{\mathcal{A}} = 0.$$

If $\mathcal{A}$ is stable and pre-attractive for $\mathcal{H}$, then it is said to be *pre-asymptotically stable* (pAS). $\diamond$

For the global stability properties, we consider *uniform* global pre-asymptotic stability of sets, which is a stronger condition than global pre-asymptotic stability. The "uniformity" in the defined term refers to the requirement that for each $\varepsilon > 0$ and $r > 0$, there is a uniform bound $T > 0$ on the hybrid time it takes any hybrid solution that starts within a distance of r from $\mathcal{A}$ to converge within a distance ε from $\mathcal{A}$.

Definition 1.4 ([3, Def. 3.7])**.** For a hybrid system $\mathcal{H}$ as in (1.3), a nonempty set $\mathcal{A} \subset \mathbb{R}^n$ is said to be

- *uniformly globally stable* for $\mathcal{H}$ if there exists a class-$\mathcal{K}_\infty$ function α such that every solution φ to $\mathcal{H}$ satisfies $|\varphi(t,j)|_{\mathcal{A}} \leq \alpha(|\varphi(0,0)|_{\mathcal{A}})$ for each $(t,j) \in \operatorname{dom}(\varphi)$; and
- *uniformly globally pre-attractive* for $\mathcal{H}$ if for each $\varepsilon > 0$ and $r > 0$, there exists $T > 0$ such that every solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \leq r$ satisfies $|\varphi(t,j)|_{\mathcal{A}} \leq \varepsilon$ for all $(t,j) \in \operatorname{dom}(\varphi)$ such that $t + j \geq T$.

If $\mathcal{A}$ is both uniformly globally stable and uniformly globally pre-attractive for $\mathcal{H}$, then it is said to be *uniformly globally pre-asymptotically stable* (UGpAS) for $\mathcal{H}$. $\diamond$

If every maximal solution to $\mathcal{H}$ is complete, then the "pre-" prefixes are omitted,

in which case, if $\mathcal{A}$ is pAS or UGpAS, then we say $\mathcal{A}$ is, respectively, *asymptotically stable* (AS) or *uniformly globally asymptotically stable* (UGAS).

A classical approach to show stability is by construction of a Lyapunov function (of some sort). The following definition gives basic assumptions that make a function a viable candidate to be a Lyapunov function in the context of hybrid systems with set-valued maps.

Definition 1.5 (Lyapunov function candidate)**.** Consider a hybrid system $\mathcal{H} := (C, F, D, G)$ on $\mathbb{R}^n$ and a set $\mathcal{A} \subset \mathbb{R}^n$. A function $V : \operatorname{dom}(V) \to \mathbb{R}$ is a *Lyapunov function candidate* with respect to $\mathcal{A}$ for $\mathcal{H}$ if the following conditions hold:

(LFC1) $\big(\overline{C} \cup D \cup G(D)\big) \subset \operatorname{dom} V$;

(LFC2) V is continuous and is locally Lipschitz on an open neighborhood of $\overline{C}$;

(LFC3) V is positive definite on $C \cup D \cup G(D)$ with respect to $\mathcal{A}$. ◇

A Lyapunov theorem for hybrid systems is given in [3, Thm. 3.19]. In Chapter 5, we present an alternative hybrid Lyapunov theorem by relaxing the assumptions of [3, Thm. 3.19]. The following corollary gives a special case of the results in Theorem 5.1 which

1. illustrates the general form of a hybrid Lyapunov theorem and
2. is used in Chapter 4 to assert uniform global asymptotic stability.

Corollary 1.1. *Consider a hybrid system $\mathcal{H} = (C, F, D, G)$ on $\mathbb{R}^n$, a nonempty compact set $\mathcal{A} \subset \mathbb{R}^n$, and a Lyapunov function candidate V with respect to $\mathcal{A}$ for $\mathcal{H}$. Suppose there exists $\alpha \in \mathcal{K}_\infty$ and an LSC function $\sigma_{\mathrm{C}} \in \mathcal{PD}(\mathcal{A})$ such that*

$$\alpha(|x|_{\mathcal{A}}) \leq V(x) \qquad \forall x \in C \cup D \cup G(D), \tag{1.8a}$$

$$\sup \mathcal{L}_{F \cap T_C} V(x) \leq -\sigma_{\mathrm{C}}(x) \qquad \forall x \in C, \tag{1.8b}$$

$$V(g) \leq V(x) \qquad \forall x \in D,\ \forall g \in G(x), \tag{1.8c}$$

and, for each $r > 0$, there exist $\Delta_T > 0$ and $\Delta_J > 0$ such that for every solution with $|\varphi(0,0)|_{\mathcal{A}} \leq r$ and every $(t_0, j_0) \in \operatorname{dom}(\varphi)$ and $(t_1, j_1) \in \operatorname{dom}(\varphi)$,

$$|t_1 - t_0| \leq \Delta_T \implies |j_1 - j_0| \leq \Delta_J. \tag{1.9}$$

Then, $\mathcal{A}$ is UGpAS for $\mathcal{H}$.

1.1.5 Safety, Forward Invariance, and Barrier Functions

Definition 1.6 (Forward Invariance)**.** A set $K \subset \mathbb{R}^n$ is said to be *forward pre-invariant* for a hybrid system $\mathcal{H}$ if, for each $x_0 \in K$ and each maximal solution φ starting from $\varphi(0,0) = x_0$: $\varphi(t,j) \in K$ for all $(t,j) \in \mathrm{dom}(\varphi)$. If, additionally, each maximal solution starting in K is complete, then K is called *forward invariant*. $\diamond$

Forward invariance is a useful concept for the analysis of dynamical systems, but to show that a control system is "safe," we must show the existence of a forward invariant set that does not intersect the inadmissible (that is, every point inside the forward invariant set satisfies the system's constraints) and that contains the initial set (the set of all possible starting configurations of the system).

Definition 1.7. A control system on $\mathbb{R}^n$ with admissible set $\mathcal{X} \subset \mathbb{R}^n$ and initial set $\mathcal{X}_0 \subset \mathbb{R}^n$ is called *safe* if there exists a forward invariant set K such that $K \subset \mathcal{X}$ and $\mathcal{X}_0 \subset K$. $\diamond$

One of the standard tools for demonstrating that a set is forward (pre-)invariant is by finding a barrier function.

Definition 1.8. Consider a hybrid system $\mathcal{H} = (C, F, D, G)$ in $\mathbb{R}^n$ as in Definition 1.1 and a set $K \subset \mathbb{R}^n$. We call a $\mathcal{C}^1$ function B a *barrier function* of K for $\mathcal{H}$ if

(B1) $K = \{z \in \mathbb{R}^n \mid B(z) \leq 0\}$,

(B2) There exists a neighborhood U of K such that for all $x \in (U \setminus K) \cap C$,

$$\sup \mathcal{L}_{F \cap T_C} B(x) \leq 0,$$

(B3) For all $x \in K \cap D$,

$$G(x) \subset C \cup D,$$

(B4) For all $x \in K \cap D$ and $\gamma \in G(x)$,

$$B(\gamma) \leq 0.$$ $\diamond$

By [5, Thm. 1], the existence of a barrier function implies that K is forward pre-invariant. In the following corollary, we give a simplified version of [5, Thm. 1] that is sufficient for our uses.

Corollary 1.2. *Suppose that $\mathcal{H}$ is a hybrid system as in Definition 1.1 that satisfies the hybrid basic conditions in Definition 1.2. If there exists a $\mathcal{C}^1$ barrier function of a set K for $\mathcal{H}$ as in Definition 1.8, then K is forward pre-invariant for $\mathcal{H}$.*

For a differential equation $\dot{z} = f(z)$, conditions (B3) and (B4) hold vacuously and (B2) simplifies to the following:

(B2′) There exists a neighborhood U of K such that

$$\langle \nabla B(x),\, f(x) \rangle \leq 0 \quad \forall x \in U \setminus K.$$

Chapter 2

Uniting Feedback For Safety with Static Controllers

This chapter introduces a uniting feedback strategy for ensuring safety while using an uncertified controller for a continuous-time plant and static feedback controllers, which are assumed to be continuous functions. Compared to subsequent chapters, this setting is the simplest since it does not require a mechanism to ensure convergence, as in Chapter 3, nor does it involve set-valued dynamics as in Chapter 4.

For a constrained nonlinear control system, we introduce a supervisor that controls switches between a barrier function–certified controller and an uncertified controller. The supervisor's switching strategy allows for properties of the uncertified controller to be exploited while preserving the forward invariance that is guaranteed by the barrier function for the certified controller. Tunable threshold functions determine regions of the state space where the supervisor switches between controllers. Conditions are given to prevent chattering by establishing a positive minimum time between switches. An example illustrates achieving forward invariance despite using an uncertified MPC controller with delayed computations.

Control systems often have operational constraints, such as physical obstacles, legal regulations, or limits on the amount of force or electrical current that a system can safely endure. A popular approach to verify that a system satisfies its constraints is via a barrier function (also called a *barrier certificate*) [5], [6], [7]. There are several definitions of barrier functions in the literature [8]. For the definition used in this chapter, a *barrier function* maps the system's state space to $\mathbb{R}$ and satisfies conditions such that its zero-sublevel set is forward invariant and every point in that set is admissible. The zero-level set is a barrier that the state cannot cross, so if the system

starts in the zero-sublevel set, then it is *safe*.

We consider a continuous-time nonlinear plant with state space $\mathbb{R}^n$ and a set $K \subset \mathbb{R}^n$ that we want to render forward invariant. If, for a controller κ, the set K is rendered forward invariant and a barrier function of K is known for the closed-loop system, then we say κ is *barrier-certified*. A controller for which a barrier function is unavailable is *uncertified*. Per Definition 1.7, we assume that system designers would pick K such that it is fully within the system constraints and contains all of the initial states so that forward invariance of K implies system is safe.

Although uncertified controllers are not expected to render the set K forward invariant, they can have other desirable properties, such as tracking a reference trajectory, minimizing control effort, or reducing computational demands. As an example, consider model predictive control (MPC). An MPC controller computes the input at discrete sample times by solving a finite-horizon optimization problem. The advantages of MPC are that it computes an approximately optimal control input that satisfies constraints. For nonlinear systems with nonlinear constraints, however, computing an MPC input is computationally expensive, which can lead to delayed updates that cause the system to violate constraints (see Example 2.2 and [9]). This motivates the development of supervisory control that uses a certified controller as "guardrails"—if the uncertified controller moves the system too close to the unsafe set, the supervisor triggers a switch to the certified controller so that the system stays in the safe set.

The *Simplex architecture* is an approach for switching between an "advanced," unverified controller and a "simple," easy-to-verify controller [10], [11]. In the Simplex architecture, a *decision module* decides at each time step whether to use the unverified controller—if it is performing safely—or to fall back to the verified controller. In [11], barrier functions are used with the Simplex architecture to achieve safety for hybrid systems, but this approach requires costly reachability analysis and has only "one way" switching—that is, there are no conditions given for returning to the unverified controller after switching to the verified controller. The Simplex architecture is also used with a barrier certificate in [12], but there are several limitations to their approach that we overcome in this chapter; namely, only rectangular constraints are considered, and the switching criteria depends on the extremal values of the vector field over the entire admissible set, leading to excessive

conservatism.

In this chapter, we introduce a hybrid control strategy for switching between a barrier-certified controller κ_0 and an uncertified controller κ_1 such that the set K is forward invariant for the resulting hybrid closed-loop system; the uncertified controller κ_1 is preferred over the certified controller κ_0; and the switching between κ_0 and κ_1 does not chatter (i.e., the time between all switches is greater than some positive constant). In our switching strategy, user-defined thresholds on the value and the rate-of-change of the barrier function determine where switches occur. The thresholds are defined as functions of the state, so that larger margins can be chosen in regions where the system has faster dynamics. We show that our hybrid control strategy renders K forward invariant, and we provide conditions for establishing a positive minimum time between switches. This work was first published in [13].

2.1 Problem Setting

Consider a continuous-time plant

$$\dot{z} = f_{\mathrm{P}}(z, u) \tag{2.1}$$

with state $z \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$, and $f_{\mathrm{P}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. Suppose we are given a closed set $K \subset \mathbb{R}^n$ to be rendered forward invariant, and two static feedback controllers κ_0, $\kappa_1 : \mathbb{R}^n \to \mathbb{R}^m$ such that the vector fields

$$z \mapsto f_{\mathrm{P}}(z, \kappa_0(z)) \quad \text{and} \quad z \mapsto f_{\mathrm{P}}(z, \kappa_1(z))$$

are continuous. In conjunction with κ_0, we are also given a $\mathcal{C}^1$ barrier function $B : \mathbb{R}^n \to \mathbb{R}$ of K for the closed–loop

$$\dot{z} = f_0(z) := f_{\mathrm{P}}(z, \kappa_0(z)). \tag{2.2}$$

The controller κ_1 is not assumed to render K forward invariant for the closed–loop

$$\dot{z} = f_1(z) := f_{\mathrm{P}}(z, \kappa_1(z)). \tag{2.3}$$

Since B guarantees that K is forward invariant for (2.2), we call κ_0 a *certified* controller, whereas κ_1, which has no such guarantee, is called *uncertified*.

Given the $\mathcal{C}^1$ barrier function B of K for (2.2), we define

$$\dot{B}_q(z) := \langle \nabla B(z),\, f_{\mathrm{P}}\big(z, \kappa_q(z)\big)\rangle \qquad \forall (z,q) \in \mathcal{X}, \tag{2.4}$$

which is the (hypothetical) rate of change of $t \mapsto B(z(t))$ if $t \mapsto z(t)$ were to evolve according to $\dot{z} = f_q(z)$.

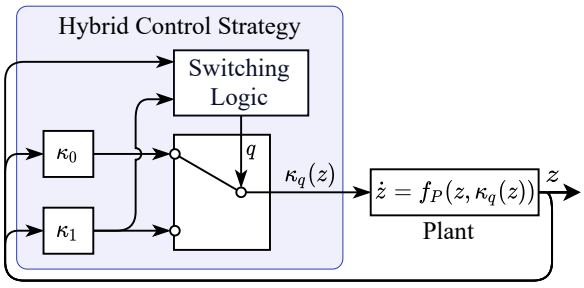

Figure 2.1. Feedback diagram for the closed–loop system $\mathcal{H}$ in (2.6).

The decision unit that determines when to switch between κ_0 and κ_1 is called a *supervisor*. As shown in Figure 2.1, an auxiliary logic variable $q \in \{0, 1\}$ is used to select which controller is used. When $q = 0$, the certified controller κ_0 is used and when $q = 1$, the uncertified controller κ_1 is used. The supervisor's switching logic is defined by two *switching sets*: $\mathcal{Z}_{0\mapsto 1}$, $\mathcal{Z}_{1\mapsto 0} \subset \mathbb{R}^n$. The set $\mathcal{Z}_{0\mapsto 1}$ specifies where the supervisor switches from $q = 0$ to $q = 1$ and the set $\mathcal{Z}_{1\mapsto 0}$ specifies where the supervisor switches from $q = 1$ to $q = 0$. As complements of the switching sets, we define *hold sets*

$$\mathcal{Z}_0 := \overline{\mathbb{R}^n \setminus \mathcal{Z}_{0\mapsto 1}} \quad \text{and} \quad \mathcal{Z}_1 := \overline{\mathbb{R}^n \setminus \mathcal{Z}_{1\mapsto 0}} \tag{2.5}$$

that specify where the supervisor holds constant $q = 0$ and $q = 1$, respectively. In Section 2.2, we design $\mathcal{Z}_{0\mapsto 1}$ and $\mathcal{Z}_{1\mapsto 0}$ such that the hybrid closed–loop system with the switched feedback $u := \kappa_q(z)$ satisfies the following properties:

- The set K is forward invariant.
- The uncertified controller κ_1 is preferred over the certified controller κ_0.
- The switching between κ_0 and κ_1 does not chatter.

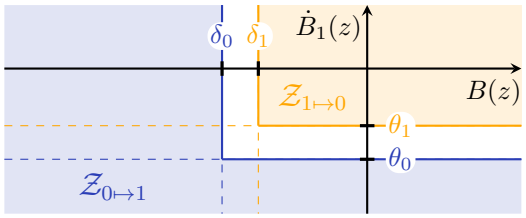

Figure 2.2. Diagram of the switching sets $\mathcal{Z}_{0\mapsto 1}$ and $\mathcal{Z}_{1\mapsto 0}$.

2.2 Hybrid Closed–Loop System

We model the closed–loop system with a supervisor for switching between controllers κ_0 and κ_1 as a hybrid system $\mathcal{H}$ with state $x := (z, q)$ in state space $\mathcal{X} := \mathbb{R}^n \times \{0, 1\}$, and dynamics given by

$$\mathcal{H}: \begin{cases} \begin{bmatrix} \dot{z} \\ \dot{q} \end{bmatrix} = f(z,q) := \begin{bmatrix} f_q(z) \\ 0 \end{bmatrix} & (z,q) \in C := C_0 \cup C_1 \\ \begin{bmatrix} z^+ \\ q^+ \end{bmatrix} = g(z,q) := \begin{bmatrix} z \\ 1-q \end{bmatrix} & (z,q) \in D := D_0 \cup D_1 \end{cases} \tag{2.6}$$

where

$$C_0 := \mathcal{Z}_0 \times \{0\}, \qquad C_1 := \mathcal{Z}_1 \times \{1\},$$
$$D_0 := \mathcal{Z}_{0\mapsto 1} \times \{0\}, \quad D_1 := \mathcal{Z}_{1\mapsto 0} \times \{1\}.$$

To design $\mathcal{Z}_{0\mapsto 1}$ and $\mathcal{Z}_{1\mapsto 0}$, we introduce four *threshold functions* $\delta_0, \delta_1, \theta_0, \theta_1 : \mathbb{R}^n \to \mathbb{R}_{\leq 0}$, such that

$$\delta_0(z) < \delta_1(z) \leq 0 \quad \text{and} \quad \theta_0(z) < \theta_1(z) \leq 0 \quad \forall z \in \mathbb{R}^n. \tag{2.7}$$

We use the functions δ_0 and δ_1 as thresholds on B and the functions θ_0 and θ_1 as thresholds on $\dot{B}_1$ to determine where switches occur. Thus, we define the switching sets as

$$\begin{aligned} \mathcal{Z}_{0\mapsto 1} &:= \{z \in \mathbb{R}^n \mid B(z) \leq \delta_0(z) \text{ or } \dot{B}_1(z) \leq \theta_0(z)\} \\ \mathcal{Z}_{1\mapsto 0} &:= \{z \in \mathbb{R}^n \mid B(z) \geq \delta_1(z),\ \dot{B}_1(z) \geq \theta_1(z)\}. \end{aligned} \tag{2.8}$$

The switching sets $\mathcal{Z}_{0\mapsto 1}$ and $\mathcal{Z}_{1\mapsto 0}$ are shown in Figure 2.2. Expanding the definitions in (2.5) of the hold sets $\mathcal{Z}_0$ and $\mathcal{Z}_1$ produces

$$\begin{aligned} \mathcal{Z}_0 &= \{z \in \mathbb{R}^n \mid B(z) \geq \delta_0(z),\ \dot{B}_1(z) \geq \theta_0(z)\} \\ \mathcal{Z}_1 &= \{z \in \mathbb{R}^n \mid B(z) \leq \delta_1(z) \text{ or } \dot{B}_1(z) \leq \theta_1(z)\}. \end{aligned} \tag{2.9}$$

We have that $C \cup D = \mathcal{X}$ because $\mathcal{Z}_0 \cup \mathcal{Z}_{0\mapsto 1} = \mathcal{Z}_1 \cup \mathcal{Z}_{1\mapsto 0} = \mathbb{R}^n$.

The set $\mathcal{Z}_0$ is designed such that the supervisor continues to use the certified controller κ_0 so long as the state is close to the boundary of K (namely, $B(z) \geq \delta_0(z)$) and the hypothetical rate of change of B under κ_1 is too large ($\dot{B}_1(z) \geq \theta_0(z)$). As the complement, $\mathcal{Z}_{0\mapsto 1}$ is designed such that the supervisor switches to the uncertified controller κ_1 when the state is either far from ∂K (i.e., $B(z) \leq \delta_0(z)$) or the hypothetical rate that B would decrease under κ_1 is fast enough ($\dot{B}_1(z) \leq \theta_0(z)$).

For $q = 1$, the set $\mathcal{Z}_1$ is designed such that the supervisor continues to use the uncertified controller κ_1 at each state $z \in K$ that is far from ∂K or where the rate that B would decrease under κ_1 is fast enough. The set $\mathcal{Z}_{1\mapsto 0}$ is the closed complement of $\mathcal{Z}_1$ and is designed to trigger a switch to the certified controller κ_0 whenever the state is too close to K and is moving toward K (or, more precisely, not moving away fast enough).

Example 2.1. To illustrate the design of $\mathcal{H}$, consider the double integrator plant

$$\dot{z} = f_{\mathrm{P}}(z, u) := \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u. \tag{2.10}$$

Suppose we want the system to avoid a disk with radius 1, centered on the z_1-axis at $c := (5, 0) \in \mathbb{R}^2$. The admissible set, which we want to render forward invariant, is

$$K := \big\{ z \in \mathbb{R}^2 : |z - c| \geq 1 \big\} = \big\{ z \in \mathbb{R}^2 : 1 - |z - c| \leq 0 \big\}.$$

Let $\kappa_0(z) := [-1 \quad 1](z - c)$ and $B(z) := \frac{1}{2}(1 - |z - c|^2)$.

The closed–loop system is

$$\dot{z} = f_0(z) = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix} (z - c)$$

and

$$\dot{B}_0(z) = -(z - c)^\top \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix} (z - c) = -z_2^2 \leq 0.$$

Thus, K is forward invariant for the system $\dot{z} = f_0(z)$.

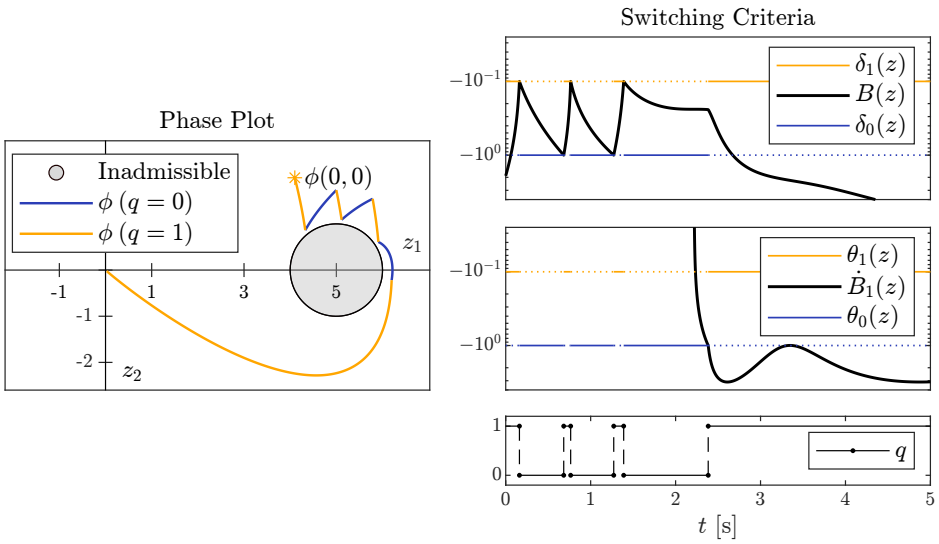

Figure 2.3. A solution φ (left) for Example 2.1 and the corresponding switching criteria (right). The first five switches occur when $B(z) \leq \delta_0(z)$ or $B(z) \geq \delta_1(z)$. At $t = 2.4\,\text{sec}$, a switch to $q = 1$ occurs because $\dot{B}_1(z) \leq \theta_0(z)$. Dotted lines indicate thresholds that do not have an effect for the current value of q.

For the uncertified controller, let $\kappa_1(z) := [-1 \quad -2]z$, which renders the origin of system (2.10) globally exponentially stable, but violates constraints. The closed–loop system under κ_1 is

$$\dot{z} = f_1(z) = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} z.$$

The set K, however, is not forward invariant under κ_1. At $z = (5, 1) \in \partial K$,

$$\dot{B}_1(z) = (z - c)^\top \begin{bmatrix} 0 & -1 \\ 1 & 2 \end{bmatrix} z > 0.$$

We use constant threshold functions, which we write (with abuse of notation) as $\theta_0 := -1$, $\theta_1 := -0.1$, $\delta_0 := -1$, and $\delta_1 := -0.1$. Figure 2.3 shows a solution to $\mathcal{H}$ and the corresponding switching criteria.[1] These plots show that the system is controlled by the uncertified controller κ_1 until it becomes too close to the obstacle and switches to the certified controller κ_0. The closed–loop system $\mathcal{H}$ satisfies the assumptions of Theorem 2.2 given in Section 2.3, so the set K is forward invariant for $\mathcal{H}$. ◇

[1] Simulations are computed in MATLAB with the HyEQ Toolbox [14].

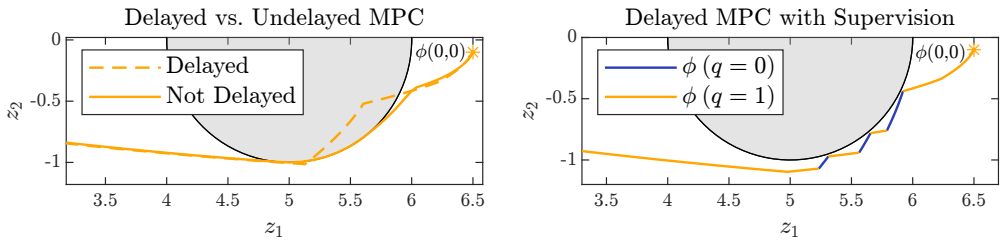

Figure 2.4. Trajectories for Example 2.2 using only κ_1 without supervision (left) and using supervised switching between κ_0 and κ_1 (right). Computational delays cause unsupervised κ_1 to violate constraints (dashed line), using our supervisor preserves the constraint due to switching to κ_0 near the inadmissible set.

Example 2.2. Consider the system given in Example 2.1 with the uncertified controller κ_1 replaced by an MPC controller. If each periodic MPC computation finishes immediately, then the trajectory grazes the boundary of the unsafe set but does not enter it. MPC computations can be slow, however, in the presence of nonlinear or non-convex constraints. In Figure 2.4, we see that adding small, random delays to the update times for the MPC input causes the solution to violate the constraint, but using our supervisory control strategy, $\mathcal{H}$ respects the constraint by switching to the certified controller. ◇

2.3 Forward Invariance of K

Our first result, Theorem 2.1, states that K is forward pre-invariant for $\mathcal{H}$, meaning that each solution to $\mathcal{H}$ remains in K for as long as the solution exists. Under stronger assumptions, Theorem 2.2 asserts that K is forward invariant by establishing that every maximal solution φ is complete ($\sup_t \operatorname{dom}(\varphi) = \infty$ or $\sup_j \operatorname{dom}(\varphi) = \infty$) and, if φ is bounded, then $\sup_t \operatorname{dom}(\varphi) = \infty$.

Theorem 2.1 (Forward Pre-Invariance)**.** *Consider $\mathcal{H}$ in* (2.6)*. Suppose B is a $\mathcal{C}^1$ barrier function of K for $\dot{z} = f_0(z)$; the vector fields f_0 and f_1 are continuous; and the threshold functions δ_0, δ_1, θ_0, and θ_1 satisfy the inequalities in* (2.7)*. Then, $K' := K \times \{0, 1\}$ is forward pre-invariant for $\mathcal{H}$ in* (2.6)*.*

Proof. Let $B' := (z, q) \mapsto B(z)$. The proof proceeds by first showing B' is a barrier function of K' for $\mathcal{H}$, and is completed by applying Corollary 1.2.

By assumption, B is a barrier function of K for $\dot{z} = f_0(z)$, so we have that $K = \{z \in \mathbb{R}^n \mid B(z) \leq 0\}$. Thus, for all $(z, q) \in \mathcal{X}$,

$$K' = \big\{(z, q) \in \mathcal{X} \;\big|\; B'(z, q) \leq 0\big\},$$

because $B(z) = B'(z, q)$. Thus, B' is a barrier function candidate of K', meaning it satisfies (B1).

Because B is a barrier function of K for $\dot{z} = f_0(z)$, we can take from (B2′) a neighborhood U of K where $\dot{B}_0(z) \leq 0$. Consequently, the set $U' := U \times \{0, 1\}$ is a neighborhood of K'. We want to show that $\langle \nabla B'(z, q),\, f(z, q)\rangle \leq 0$ for all $(z, q) \in (U' \setminus K') \cap C$, as required by (B2). Every element (z, q) of $(U' \setminus K') \cap C$ satisfies one of two disjoint cases:

- If $q = 0$ and $z \in (U \setminus K) \cap \mathcal{Z}_0$, then, by (B2′),

$$\langle \nabla B'(z, 0),\, f_0(z)\rangle = \dot{B}_0(z) \leq 0.$$

- If $q = 1$ and $z \in (U \setminus K) \cap \mathcal{Z}_1$, then by the design of $\mathcal{Z}_1$, either $B(z) \leq \delta_1(z) \leq 0$ or $\dot{B}_1(z) \leq \theta_1(z) \leq 0$. Because $z \notin K$, we must have $B(z) > 0 \geq \delta_1(z)$. Thus, every z in $(U \setminus K) \cap \mathcal{Z}_1$ satisfies $\dot{B}_1(z) \leq \theta_1(z)$, so

$$\langle \nabla B'(z, 1),\, f_1(z)\rangle = \dot{B}_1(z) \leq \theta_1(z) \leq 0.$$

Therefore, (B2) is satisfied.

From the definition of g, we have that $B'(g(z, q)) = B'(z, 1 - q)$. Additionally, the assumption (B1) states that $B(z) \leq 0$ for all $z \in K$. Thus,

$$B'(g(z, q)) \leq 0 \quad \forall (z, q) \in K' \cap D. \tag{2.11}$$

Finally, the condition

$$g(z, q) \in C \cup D \quad \forall (z, q) \in K' \cap D$$

holds trivially because $C \cup D = \mathcal{X}$. Thus, (B4) is satisfied, so B' is a barrier function of K' for $\mathcal{H}$.

Checking the remaining assumptions of Corollary 1.2, we have that f is continuous because f_0 and f_1 are continuous by assumption. The set K' is closed because B' is $\mathcal{C}^1$. Therefore, by Corollary 1.2, K' is forward pre-invariant for $\mathcal{H}$. □

In the following results, we assert, under appropriate assumptions, that the closed–loop system $\mathcal{H}$ is well-posed (Lemma 2.2), the existence of a lower bound on the time between switches for bounded solutions (Lemma 2.2), and that every maximal solution to $\mathcal{H}$ is complete.

Lemma 2.1 (Hybrid Basic Conditions)**.** *Consider $\mathcal{H}$ in* (2.6)*. Suppose f_0, f_1, θ_0, θ_1, δ_0, and δ_1 are continuous and B is $\mathcal{C}^1$. Then, the system $\mathcal{H}$ in* (2.6) *satisfies the hybrid basic conditions* (A1)–(A3) *in Definition 1.2.*

Proof. The continuity of f and g follow directly their definitions and the continuity of f_0 and f_1, thereby satisfying (A2) and (A3). The gradient ∇B is continuous because B is $\mathcal{C}^1$, and the vector field f_0 is continuous by assumption, so the function $\dot{B}_0$ is continuous also. Thus, the sets $\mathcal{Z}_0$, $\mathcal{Z}_{0\mapsto 1}$, $\mathcal{Z}_1$, and $\mathcal{Z}_{1\mapsto 0}$ are closed because they can be written as finite unions and intersections of sublevel sets of continuous functions:

$$\begin{aligned}
\mathcal{Z}_0 &= \{z \in \mathbb{R}^n \mid \theta_0(z) - \dot{B}_1(z) \leq 0\} \cap \{z \in \mathbb{R}^n \mid \delta_0(z) - B(z) \leq 0\} \\
\mathcal{Z}_{0\mapsto 1} &= \{z \in \mathbb{R}^n \mid \dot{B}_1(z) - \theta_0(z) \leq 0\} \cup \{z \in \mathbb{R}^n \mid B(z) - \delta_0(z) \leq 0\} \\
\mathcal{Z}_1 &= \{z \in \mathbb{R}^n \mid \dot{B}_1(z) - \theta_1(z) \leq 0\} \cup \{z \in \mathbb{R}^n \mid B(z) - \delta_1(z) \leq 0\} \\
\mathcal{Z}_{1\mapsto 0} &= \{z \in \mathbb{R}^n \mid \theta_1(z) - \dot{B}_1(z) \leq 0\} \cap \{z \in \mathbb{R}^n \mid \delta_1(z) - B(z) \leq 0\}.
\end{aligned}$$

Thus, $C = (\mathcal{Z}_0 \times \{0\}) \cup (\mathcal{Z}_1 \times \{1\})$ and $D = (\mathcal{Z}_{0\mapsto 1} \times \{0\}) \cup (\mathcal{Z}_{1\mapsto 0} \times \{1\})$ are closed, satisfying (A1). □

In the following result, we assert (under appropriate assumptions) that each bounded solution to the closed–loop system $\mathcal{H}$ has a positive minimum dwell time between jumps. This result, combined with a proof that all maximal solutions are complete (in Theorem 2.2, below), allows us to conclude that maximal solutions exist for all time $t \geq 0$.

Lemma 2.2. *Suppose $B : \mathbb{R}^n \to \mathbb{R}$ is $\mathcal{C}^1$; the vector fields f_0 and f_1 are continuous; and the threshold functions δ_0, δ_1, θ_0, and θ_1 are continuous and satisfy the inequalities in* (2.7)*. For each solution φ to $\mathcal{H}$ in* (2.6)*, if φ is bounded, then there exists $\gamma > 0$ such that $t_{j+1} - t_j \geq \gamma$ for every pair of jump times t_j and t_{j+1} in* $\mathrm{dom}(\varphi)$*.*

Proof. To establish a lower bound on the time between jumps, we will show D and $g(D)$ are disjoint and apply [3, Prop. 2.34]. The sets D and $g(D)$ are disjoint if and

only if $\mathcal{Z}_{0\mapsto 1}$ and $\mathcal{Z}_{1\mapsto 0}$ are disjoint because

$$D_0 = \mathcal{Z}_{0\mapsto 1} \times \{0\} \quad \text{and} \quad D_1 = \mathcal{Z}_{1\mapsto 0} \times \{1\},$$

and

$$g(D_0) = \mathcal{Z}_{0\mapsto 1} \times \{1\} \quad \text{and} \quad g(D_1) = \mathcal{Z}_{1\mapsto 0} \times \{0\}.$$

Take any $z^0 \in \mathcal{Z}_{0\mapsto 1}$. By (2.7) and the definition of $\mathcal{Z}_{0\mapsto 1}$,

$$\dot{B}_1(z^0) \leq \theta_0(z^0) < \theta_1(z^0) \quad \text{or} \quad B(z^0) \leq \delta_0(z^0) < \delta_1(z^0),$$

thus $z^0 \notin \mathcal{Z}_{1\mapsto 0}$. Similarly, for every $z^1 \in \mathcal{Z}_{1\mapsto 0}$,

$$\dot{B}_1(z^1) \geq \theta_1(z^1) > \theta_0(z^1) \quad \text{and} \quad B(z^1) \geq \delta_1(z^1) > \delta_0(z^1),$$

so $z^1 \notin \mathcal{Z}_{0\mapsto 1}$. Therefore, $\mathcal{Z}_{0\mapsto 1}$ and $\mathcal{Z}_{1\mapsto 0}$ are disjoint and $D \cap g(D) = \varnothing$.

By Lemma 2.1, $\mathcal{H}$ satisfies the hybrid basic conditions Definition 1.2. Thus, the conclusion follows from [3, Prop. 2.34]. □

To ensure solutions to $\mathcal{H}$ exist for all $t \geq 0$, we require that all solutions to $\dot{z} = f_0(z)$ and $\dot{z} = f_1(z)$ do not exhibit "finite escape times." We say that $z : [t_0, T) \to \mathbb{R}^n$ with $t_0 < T$ has a *finite escape time* T if $\lim_{t \nearrow T} |z(t)| = \infty$.

Lemma 2.3 (Maximal Solutions are Complete)**.** *Suppose B is a $\mathcal{C}^1$ barrier function of K for $\dot{z} = f_0(z)$; the functions f_0, f_1, θ_0, θ_1, δ_0, and δ_1 are continuous; for each $q \in \{0, 1\}$, no solution to*

$$\dot{z} = f_q(z) \quad z \in \mathcal{Z}_q$$

has a finite escape time; and the threshold functions θ_0, θ_1, δ_0, and δ_1 are continuous and satisfy the inequalities in (2.7)*. Then, every maximal solution φ to $\mathcal{H}$ in* (2.6) *is complete. If, additionally, φ is bounded, then $\sup_t \operatorname{dom}(\varphi) = \infty$ (that is, φ is defined for all ordinary time $t \geq 0$).*

Proof. Our proof uses [3, Prop. 2.34]. For a hybrid system $\mathcal{H}' = (C', f', D', g')$, a point $x_0 \in C' \cup D'$ is said to satisfy the *viability condition* (VC) if there exists a neighborhood U of x_0 such that $f'(x) \in T_{C'}(x)$ for every $x \in U \cap C'$.

To show that VC holds for $\mathcal{H}$ at every point in $C \setminus D$, we use the following fact: For any set $S \subset \mathbb{R}^n$, the tangent cone of S at an interior point $x \in \operatorname{int} S$ is the entire space. That is,

$$T_S(x) = T_{\mathbb{R}^n}(x) \qquad \forall x \in \operatorname{int} S.$$

Because, by assumption, B is $\mathcal{C}^1$ and f_0, f_1, δ_0, δ_1, θ_0, and θ_1 are continuous, Lemma 2.1 asserts that C is closed (A1). Because C is closed and $D = \overline{\mathcal{X} \setminus C}$, we have that $C \setminus D \subset \operatorname{int} C$. Thus, $T_C(x) = T_{\mathcal{X}}(x)$ for all $x \in C \setminus D$. It follows that for every $x_0 \in C \setminus D$, there exists a neighborhood $U \subset \operatorname{int} C$ of x_0 such that $f_{\mathrm{P}}(x) \in T_C(x)$ for all $x \in U \cap C$. Therefore, VC holds everywhere in $C \setminus D$.

Because $\mathcal{H}$ satisfies the hybrid basic conditions and VC holds at each point in $C \setminus D$, every maximal solution φ to $\mathcal{H}$ satisfies exactly one of the following conditions [3, Prop. 2.34]:

(M1) φ escapes to infinity in finite time.

(M2) φ leaves $C \cup D$.

(M3) φ is complete.

By assumption, every solution does not escape to infinity in finite time, ruling out (M1). Furthermore, $g(D) \subset C \cup D = \mathcal{X}$, so, per the note in [3, Prop. 2.34], (M2) does not occur. By elimination, only (M3) is possible, therefore every maximal solution is complete.

Let φ be a maximal solution to $\mathcal{H}$ and let $T := \sup_t \operatorname{dom}(\varphi)$ and $J := \sup_j \operatorname{dom}(\varphi)$. Because φ is complete, either $T = \infty$ or $J = \infty$ (or both). If $T = \infty$, then φ is defined for all ordinary time. Suppose $J = \infty$. Let $\{t_j\}_{j=0}^{\infty}$ be the sequence of jump times in φ. With the given assumptions, Lemma 2.2 asserts that there exists $\gamma > 0$ such that

$$t_{j+1} - t_j \geq \gamma \quad \text{for } j = 0, 1, \ldots.$$

The infinite sum of a positive constant diverges, so

$$T = \sum_{j=0}^{\infty} t_{j+1} - t_j \geq \sum_{j=0}^{\infty} \gamma = \infty. \qquad \square$$

Theorem 2.2 (Forward Invariance)**.** *Suppose B is a $\mathcal{C}^1$ barrier function of K for $\dot{z} = f_0(z)$; the vector fields f_0 and f_1 are continuous; the threshold functions δ_0, δ_1, θ_0, and θ_1 are continuous and satisfy the inequalities in* (2.7)*; and for each $q \in \{0, 1\}$, no solution to*

$$\dot{z} = f_q(z) \quad z \in \mathcal{Z}_q$$

has a finite escape time. Then, $K' := K \times \{0,1\}$ *is forward invariant for* $\mathcal{H}$ *and every maximal solution* φ *to* $\mathcal{H}$ *is complete. Furthermore, if* φ *is bounded, then* $\sup_t \mathrm{dom}(\varphi) = \infty$.

Proof. By Theorem 2.1, the set K' is forward pre-invariant for $\mathcal{H}$. Furthermore, Lemma 2.3 asserts that every maximal solution is complete. □

The "no finite escape time" assumption in Theorem 2.2 is satisfied if, for each $q \in \{0,1\}$, the vector field f_q is globally Lipschitz continuous or the set $\mathcal{Z}_q$ is bounded.

Remark 2.1. Under the assumptions of Theorem 2.2, $\mathcal{H}$ is well-posed because it satisfies the *hybrid basic conditions* in [4, Assumption 6.5]. Solutions to a well-posed hybrid system have (in a sense) continuous dependence on initial conditions, although the sense of continuity is weaker (upper semi-continuous instead of continuous) than it is for well-posed continuous-time systems [4, Chapter 6].

2.4 Unbounded Solutions Without Chattering

There are several practical difficulties with Lemma 2.2 and Theorem 2.2 that we address in this section. Notably, the minimum dwell time $\gamma > 0$ in Lemma 2.2 depends on the choice of solution, rather than being a uniform lower bound that applies to all solutions. This can cause problems if, for example, the minimum dwell time γ for a particular solution is shorter than the clock rate of the computer processor used to run the supervisor. Furthermore, if a solution is unbounded, then the time between switches may converge to zero, as shown in Example 2.3, below. To address these problems, Theorem 2.3 provides conditions for establishing a uniform lower bound on the time between jumps for all solutions to $\mathcal{H}$ (including unbounded solutions).

Example 2.3. One can construct $\mathcal{H}$ with $z \in \mathbb{R}^2$ and with

$$\begin{aligned} f_0(z) &= (z_1, -1), & \mathcal{Z}_{0\mapsto 1} &:= \{(z_1, z_2) \mid z_2 \leq 0\}, \\ f_1(z) &= (z_1, +1), & \mathcal{Z}_{1\mapsto 0} &:= \left\{(z_1, z_2) \;\middle|\; z_2 \geq \exp(-z_1^2)\right\}, \end{aligned}$$

such that $\mathcal{H}$ satisfies the assumptions of Theorem 2.2. Consider a maximal and complete solution φ that starts in the right-half plane. The z_1-component of φ grows exponentially, approaching $+\infty$ as $t + j \to \infty$, so φ is unbounded. Meanwhile, the

z_2-component of φ bounces between $\mathcal{Z}_{0\mapsto 1}$ and $\mathcal{Z}_{1\mapsto 0}$ as the distance between them approaches zero—causing the time between switches to also approach zero. $\diamond$

The following result, Lemma 2.4, establishes an upper bound on the distance that a solution to a differential equation can travel in a fixed amount of time, assuming a Lipschitz-continuous vector field.

Lemma 2.4. *Suppose $f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous with Lipschitz constant L. Then, for each $x_0 \in \mathbb{R}^n$ and each solution $t \mapsto \varphi(t)$ to $\dot{x} = f(x)$ with $\varphi(0) = x_0$, the distance between φ and x_0 satisfies*

$$|\varphi(t) - x_0| \leq t|f(x_0)| \exp(L_t) \qquad \forall t \geq 0.$$

Proof. Take any $x_0 \in \mathbb{R}^n$ and $\tau \geq 0$. Let $t \mapsto \varphi(t)$ be a solution to $\dot{x} = f(x)$, $x(0) = x_0$ and let $t \mapsto \psi(t) = x_0 + tf(x_0)$. For each $t \geq 0$,

$$\left| f(\psi(t)) - \frac{d\psi}{dt}(t) \right| = \Big| f\big(x_0 + tf(x_0)\big) - f(x_0) \Big|.$$

By Lipschitz continuity,

$$\Big| f\big(x_0 + tf(x_0)\big) - f(x_0) \Big| \leq L \Big| x_0 + tf(x_0) - x_0 \Big| = tL|f(x_0)|.$$

Therefore, by the Solution Comparison Theorem in [15] with $\varepsilon := tL|f(x_0)|$ implies that for all $t \geq 0$,

$$|\varphi(t) - \psi(t)| \leq t|f(x_0)|(\exp(Lt) - 1). \tag{2.12}$$

We substitute $x_0 = \psi(t) - tf(x_0)$ into $|\varphi(t) - x_0|$:

$$|\varphi(t) - x_0| = |\varphi(t) - \psi(t) + tf(x_0)|.$$

Using the triangle inequality, we find

$$|\varphi(t) - x_0| \leq |\varphi(t) - \psi(t)| + t|f(x_0)|.$$

Using inequality (2.12), we find that for all $t \geq 0$,

$$\begin{aligned} |\varphi(t) - x_0| &\leq t|f(x_0)|(\exp(Lt) - 1) + t|f(x_0)| \\ &= t|f(x_0)| \exp(Lt). \end{aligned}$$

□

Using the bound in Lemma 2.4, along with assumptions of Lipschitz continuity for f_0 and f_1, Lemma 2.5 provides conditions to establish a uniform lower bound on dwell times for all solutions to $\mathcal{H}$.

Lemma 2.5 (Uniform Lower Bound on Dwell Times)**.** *Consider $\mathcal{H}$ in* (2.6) *with data as in* (2.8)*. Suppose f_0 is Lipschitz continuous with Lipschitz constant L_0. Let $\tau > 0$ be fixed. If, for each $z_0 \in \mathcal{Z}_{0\mapsto 1}$, the distance from z_0 to $\mathcal{Z}_{1\mapsto 0}$ is greater than*

$$R_0(z_0, \tau) := \tau |f_0(z_0)| \exp(L_0 \tau),$$

then, every solution to $\mathcal{H}$ does not have a jump from $q = 0$ to $q = 1$ within a time τ after a jump from $q = 1$ to $q = 0$. Similarly, if, for each $z_1 \in \mathcal{Z}_{1\mapsto 0}$, the distance from z_1 to $\mathcal{Z}_{0\mapsto 1}$ is greater than

$$R_1(z_1, \tau) := \tau |f_1(z_1)| \exp(L_1 \tau),$$

then, every solution to $\mathcal{H}$ does not have a jump from $q = 1$ to $q = 0$ within a time τ after a jump from $q = 0$ to $q = 1$.

Proof. Take $\tau \geq 0$. Suppose that for each $z_0 \in \mathcal{Z}_{0\mapsto 1}$, the distance from z_0 to $\mathcal{Z}_{1\mapsto 0}$ is greater than $R_0(z_0, \tau)$. By Lemma 2.4 with $f := f_0$, the time it takes a solution to $\mathcal{H}$ to travel a distance $R_0(z_0, \tau)$ from any point z_0 in $\mathcal{Z}_{1\mapsto 0}$ is at least τ. By assumption, the distance from z_0 to $\mathcal{Z}_{0\mapsto 1}$ is greater than $R_0(z_0, \tau)$, so no solution from $\mathcal{Z}_{1\mapsto 0}$ can reach $\mathcal{Z}_{0\mapsto 1}$ in a time $t \leq \tau$.

After a switch to $q = 0$, a solution is in $\mathcal{Z}_{1\mapsto 0}$ and for a switch to $q = 1$ to occur, the solution must be in $\mathcal{Z}_{0\mapsto 1}$. Then, the solution must travel for a time greater than τ before it $\mathcal{Z}_{0\mapsto 1}$ where a switch to $q = 1$ may occur. Therefore, the time between a jump to $q = 0$ and a jump to $q = 1$ is greater than τ. □

Lemma 2.5 leads immediately to Theorem 2.3, which asserts a minimum time between all switches and thereby establishes that maximal solutions to $\mathcal{H}$ exist for all $t \geq 0$.

Theorem 2.3. *Suppose that B is a $\mathcal{C}^1$ barrier function of K for $\dot{z} = f_0(z)$; the vector fields f_0 and f_1 are globally Lipschitz continuous with Lipschitz constants L_0 and L_1; the threshold functions δ_0, δ_1, θ_0, and θ_1 are continuous and satisfy the inequalities in* (2.7)*; and there exists $\tau > 0$ such that for all $z^0 \in \mathcal{Z}_{0\mapsto 1}$ and $z^1 \in \mathcal{Z}_{1\mapsto 0}$, the following hold:*

$$|z^0 - z^1| \geq \tau |f_0(z^0)| \exp(L_0 \tau), \tag{2.13}$$

$$|z^0 - z^1| \geq \tau |f_1(z^1)| \exp(L_1 \tau). \tag{2.14}$$

Then, for every solution φ to $\mathcal{H}$ in (2.6), and each pair of jump times t_j and t_{j+1} in $\operatorname{dom}(\varphi)$, we have that $t_{j+1} - t_j \geq \tau$. Furthermore, if φ is a maximal solution, then $\sup_t \operatorname{dom}(\varphi) = \infty$.

Proof. Because f_0 and f_1 are globally Lipschitz continuous, maximal solutions to $\dot{z} = f_0(z)$ and $\dot{z} = f_1(z)$ exist for all $t \geq 0$, so no solutions have a finite escape time. By Lemma 2.3 with the given assumptions, every maximal solution to $\mathcal{H}$ is complete. By Lemma 2.5, every solution to $\mathcal{H}$ does not have a jump within a time τ after another jump. Therefore, every maximal solution φ to $\mathcal{H}$ exists for all $t \geq 0$. □

The following example illustrates Theorem 2.3 with a system that has unbounded solutions.

Example 2.4. Consider the plant

$$\dot{z} = f_{\text{P}}(z, u) := \begin{bmatrix} z_1 \\ u \end{bmatrix}, \quad z = (z_1, z_2) \in \mathbb{R}^2,\ u \in \mathbb{R}$$

with admissible set $K := \left\{ z \in \mathbb{R}^2 \;\middle|\; z_2 \leq 0 \right\}$, certified controller $\kappa_0(z) := -|z_1|$, barrier function $B(z) := z_2$, uncertified controller $\kappa_1(z) := |z_1|$, and threshold functions $\delta_0(z) := -2 - 2|z_1|$ and $\delta_1(z) := -1 - |z_1|$. The threshold functions θ_0 and θ_1 have no effect because $\dot{B}_1(z) = |z_1| \geq 0$. Thus, the switching sets are

$$\mathcal{Z}_{0\mapsto 1} = \{(z_1, z_2) \in \mathbb{R}^n \mid z_2 \leq -2 - 2|z_1|\},$$
$$\mathcal{Z}_{1\mapsto 0} = \{(z_1, z_2) \in \mathbb{R}^n \mid z_2 \geq -1 - |z_1|\}.$$

By Theorem 2.2, K is forward invariant for $\mathcal{H}$. We can apply Theorem 2.3 to show that solutions exist for all $t \geq 0$ and the time between every pair of jumps is longer than $\tau := 0.25\,\text{sec}$. The vector fields f_0 and f_1 are globally Lipschitz continuous with Lipschitz constants $L_0 = L_1 = 1$. Take any points $z^0 := (z_1^0,\ z_2^0) \in \mathcal{Z}_{0\mapsto 1}$ and $z^1 := (z_1^1,\ z_2^1) \in \mathcal{Z}_{1\mapsto 0}$. Using the geometry of $\mathcal{Z}_{0\mapsto 1}$ and $\mathcal{Z}_{1\mapsto 0}$, and the fact that $\tau \exp(L_0\tau) = \tau \exp(L_1\tau) = 0.25 \exp(0.25) < \frac{1}{3}$, we find

$$|z^0 - z^1| \geq \frac{|z_1^0| + 1}{\sqrt{5}} > \frac{1}{3}|z_1^0| > \tau |f_0(z^0)| \exp(L_0\tau),$$
$$|z^0 - z^1| \geq \frac{|z_1^1| + 1}{\sqrt{2}} > \frac{1}{3}|z_1^1| > \tau |f_1(z^1)| \exp(L_1\tau).$$

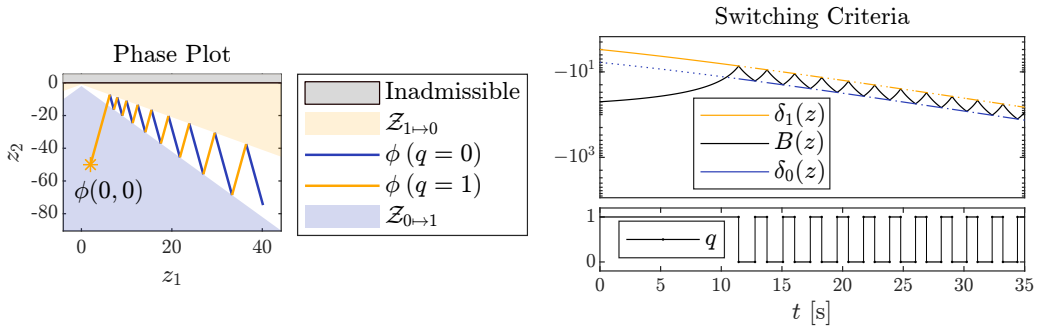

Figure 2.5. A solution φ to $\mathcal{H}$ in Example 2.4 (left) and the corresponding switching criteria for φ (right).

Take any $z^{(1)} = (z_1^{(1)}, z_2^{(1)}) \in \partial\mathcal{Z}_{0\mapsto 1}$. Thus, $B(z^{(1)}) = \delta_0(z^{(1)}) = -2 - 2|z_1^{(1)}|$. By the Lipschitz continuity of B and δ_1, for all $z' \in \mathbb{R}^2$ such that $|z' - z^{(1)}| \leq R(z^{(1)}, \tau)$,

$$|B(z') - B(z^{(1)})| \leq R(z^{(1)}, \tau)L_B < \frac{1}{3}|z_1^{(1)}|,$$
$$|\delta_1(z') - \delta_1(z^{(1)})| \leq R(z^{(1)}, \tau)L_{\delta_1} < \frac{1}{3}|z_1^{(1)}|.$$

Using triangle inequalities and substitution we find $B(z') < \delta_1(z')$, so z' is not in $\partial\mathcal{Z}_{1\mapsto 0}$. This shows that $B(z') < \delta_1(z')$, so z' is not in $\partial\mathcal{Z}_{1\mapsto 0}$. Therefore, (2.13) and (2.14) are satisfied, so Theorem 2.3 asserts that every solution to $\mathcal{H}$ exists for all time $t \geq 0$. A solution to $\mathcal{H}$ is shown in Figure 2.5 and the corresponding switching criteria are shown in Figure 2.5. ◇

It is important to note the effects of discrete sampling in the supervisor. If the supervisor only checks the switching conditions periodically (instead of continuously) with some sample time $T_s > 0$, then the set K is not, in general, forward invariant for $\mathcal{H}$. In particular, for Example 2.4, solutions that start with $\varphi(0,0)$ in $\partial K \times \{1\}$ will leave K due to the supervisor applying κ_1 over the interval $[0, T_s)$, before the first update. If, however, the threshold functions δ_0 and θ_0 are chosen such that the distance from $\mathcal{Z}_{0\mapsto 1}$ to $\mathbb{R}^n \setminus K$ is farther than the system can travel in time T_s, then solutions that start in $\mathcal{Z}_{0\mapsto 1}$ will never leave K.

Conclusion

In this chapter, we designed a supervisory hybrid control algorithm that switches between a given barrier-certified controller that renders a desired set forward invari-

ant and an uncertified controller that may not. The resulting hybrid control strategy guarantees forward invariance while preferentially using the uncertified controller. Our approach allows for advanced controllers, such as neural networks and MPC, to be safely used while avoiding the difficult task of constructing barrier functions for them.

The next chapter considers a similar system as was considered here, but with the goal of ensuring uniform global asymptotic stability. Chapter 4 provides results for extending the strategy in this chapter to hybrid plants with hybrid controllers, including systems with set-valued dynamics. Among other uses, that extension can be applied to use discontinuous static feedback controllers.

Chapter 3

Uniting Feedback for Asymptotic Stability with Static Controllers

In this chapter, we present a hybrid control strategy for rendering a set uniformly globally asymptotically stable using uniting feedback to switch between two static feedback controllers. One controller is Lyapunov-certified, which we exploit to allow us to opportunistically use the second controller, which is uncertified, while retaining the guarantee of asymptotic stability. This chapter introduces important ideas for ensuring convergence with uniting feedback and serves as a stepping stone to Chapter 4 which handles the general case of a hybrid plant with hybrid controllers.

As in Chapter 2, we consider a continuous time plant

$$\dot{z} = f_{\mathrm{P}}(z, u) \tag{3.1}$$

with state $z \in \mathbb{R}^n$ and input $u \in \mathbb{R}^m$. Let $\mathcal{A}_{\mathrm{P}} \subset \mathbb{R}^n$ be a given nonempty set we want to render globally asymptotically stable. Let $\kappa_0, \kappa_1 : \mathbb{R}^n \to \mathbb{R}^m$ be given static feedback controllers. For $q \in \{0, 1\}$, we write the closed-loop system using the feedback control law $u = \kappa_q(z)$ as $\dot{z} = f_q(z) := f_{\mathrm{P}}(z, \kappa_q(z))$. We assume that κ_0 renders $\mathcal{A}_{\mathrm{P}}$ globally asymptotically stable, as asserted by the existence of a differentiable Lyapunov function V_{P}, as formalized in Assumption 3.1, below.

Assumption 3.1 (Lyapunov Conditions)**.** For $\dot{z} = f_0(z)$ and a nonempty compact set $\mathcal{A}_{\mathrm{P}}$, there exists a differentiable function $V_{\mathrm{P}} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ such that

(L1) There exists $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ such that

$$\alpha_1(|z|_{\mathcal{A}}) \leq V_{\mathrm{P}}(z) \leq \alpha_2(|z|_{\mathcal{A}}) \quad \forall z \in \mathbb{R}^n.$$

(L2) There exists a lower semicontinuous function $\sigma : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ that is positive definite with respect to $\mathcal{A}_{\text{P}}$ such that

$$\langle \nabla V_{\text{P}}(z),\, f_0(z) \rangle \leq -\sigma(z) \quad \forall z \in \mathbb{R}^n. \qquad \diamond$$

The function V_{P} in Assumption 3.1 is called a *Lyapunov function*[1] of $\mathcal{A}_{\text{P}}$ for $\dot{z} = f_0(z)$. Condition (L2) is sufficient for $\mathcal{A}_{\text{P}}$ to be globally asymptotically stable for $\dot{z} = f_0(z)$. If $\mathcal{A}_{\text{P}}$ is compact, then Assumption 3.1 is sufficient for $\mathcal{A}_{\text{P}}$ to be *uniformly* globally asymptotically stable for $\dot{z} = f_0(z)$. If Assumption 3.1 is satisfied by $\dot{z} = f(z, \kappa_0(z))$ with Lyapunov function V_{P}, we say that κ_0 is a *Lyapunov-certified* controller.

In this chapter we design a supervisor for switching between κ_0 and κ_1, as shown in Figure 3.1, such that

1. $\mathcal{A}_{\text{P}}$ is uniformly globally asymptotically stable,
2. the time between jumps is lower bounded by a positive constant, and
3. the uncertified controller κ_1 is preferred over the certified controller κ_0.

The resulting closed-loop system is hybrid, which we model as in (1.3).

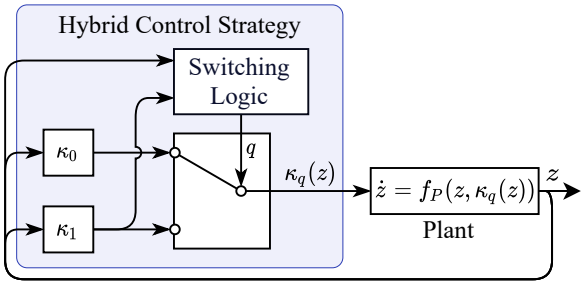

Figure 3.1. The switching logic passes q as an output to a switch, which determines whether κ_0 or κ_1 is applied to the plant.

3.1 Hybrid Control Strategy

Our hybrid control strategy uses the plant state $z \in \mathbb{R}^{n_{\text{P}}}$, the logic variable $q \in \{0, 1\}$ (described above), and an auxiliary variable $v \geq 0$. The purpose of each

[1]There are many varieties of Lyapunov function in the literature, so this should not be taken as one of many competing definitions.

variable is summarized here:

- $z \in \mathbb{R}^{n_{\text{P}}}$ is the state of the plant. Our goal is to steer z asymptotically to $\mathcal{A}_{\text{P}}$.
- $q \in \{0, 1\}$ determines the current feedback controller. When $q = 0$, then $u = \kappa_0(z)$ is used and when $q = 1$, then $u = \kappa_1(z)$ is used.
- $v \in \mathbb{R}_{\geq 0}$ is used to measure whether $V_{\text{P}}(z)$ is converging fast enough. When using the κ_1 controller, $V_{\text{P}}(z)$ may increase because κ_1 is not Lyapunov-certified, so we impose v as an upper bound on $V_{\text{P}}(z)$, thereby restricting how much $V_{\text{P}}(z)$ can grow (or fail to decrease) before triggering a switch to $q = 0$. During flows v decreases, converging to zero (per the dynamics of v designed in Section 3.2). Because v converges to zero, $V_{\text{P}}(z)$ will be squeezed to zero also.

The state of the closed-loop system is then

$$x := (z, v, q) \in \mathcal{X} := \mathbb{R}^{n_{\text{P}}} \times \mathbb{R}_{\geq 0} \times \{0, 1\}, \tag{3.2}$$

and the set that we want the closed-loop system to asymptotically stabilize is

$$\mathcal{A} := \{x \in \mathcal{X} \mid z \in \mathcal{A}_{\text{P}},\ v = 0\} = \mathcal{A}_{\text{P}} \times \{0\} \times \{0, 1\}. \tag{3.3}$$

The core idea of our hybrid control strategy is as follows:

1. The auxiliary variable v acts as an upper bound on $V_{\text{P}}(z)$. The dynamics of v chosen such that v converges to zero.
2. The system switches from $q = 1$ to $q = 0$—that is, from the uncertified controller κ_1 to the certified controller κ_0—when $V_{\text{P}}(z) \geq v$.[2]
3. The system switches from $q = 0$ (the certified controller κ_0) to $q = 1$ (the uncertified controller) when $V_{\text{P}}(z)$ is less than v with a sufficiently large gap or *buffer* between the values. The size of the buffer between $V_{\text{P}}(z)$ and v that is required to switch is defined by a function $z \mapsto \delta(z)$ such that $\delta(z) > 0$ for all $z \in \mathbb{R}^{n_{\text{P}}}$. We call δ a *buffer function*.

[2]Technically, the system is *permitted* to switch when $V_{\text{P}}(z) = v$ and *must* switch when $V_{\text{P}}(z) > v$ or $V_{\text{P}}(z) = v$ and $V_{\text{P}}(z)$ would increase relative to v. These technicalities are handled by the mathematical analysis, but unimportant in practice since $V_{\text{P}}(z) = v$ is numerically unlikely to occur, except at $V_{\text{P}}(z) = v = 0$.

At each switch, v is set to $v^+ = \max\{V_{\mathrm{P}}(z), v\}$, causing v to increase if $V_{\mathrm{P}}(z)$ is larger than v, otherwise v is not changed. Between switches, v evolves according to $\dot{v} = f_{\mathsf{v}}(z, v)$, where f_{v} is designed to force v to converge to zero. In particular,

$$\dot{v} = f_{\mathsf{v}}(z, v) := -\gamma \tanh(v)\sigma_0(z) - \mu\big(v - V_{\mathrm{P}}(z)\big), \tag{3.4}$$

where $\gamma \in (0, 1]$, $\mu > 0$, and σ_0 is the bound on the rate of change of V_{P} given in the Lyapunov condition (L2). The parameters γ and μ affect the rate at which v converges. The term $-\mu(v - V_{\mathrm{P}}(z))$ pulls v toward $V_{\mathrm{P}}(z)$, which helps v to "catch up" if $V_{\mathrm{P}}(z)$ has dropped quickly, or causes v to grow if it is initialized less than $V_{\mathrm{P}}(z)$. The term $-\gamma \tanh(v)\sigma_0(z)$ pulls v toward 0, which ensures that v converges to 0 if $V_{\mathrm{P}}(z)$ stagnates at some nonzero value. When $v = 0$, $f_{\mathsf{v}}(z, v)$ must be nonnegative, to avoid pushing v below zero, so the hyperbolic tangent $\tanh(v)$ is used as a sigmoid function that goes to zero as $v \to 0$. The specific criteria for switching are defined, as follows.

- While the feedback controller κ_0 is applied to the plant, due to q being equal to 0, we monitor $V_{\mathrm{P}}(z)$, $\delta(z)$, and v. If ever $V_{\mathrm{P}}(z) + \delta(z) \leq v$, we allow a switch from $q = 0$ to $q = 1$, since there is a sufficient buffer to safely use the uncertified controller. Conversely, if $V_{\mathrm{P}}(z) + \delta(z) \geq v$, then the system holds $q = 0$ since there is not enough of a buffer.

- While the feedback controller κ_1 is applied, due to q being equal to 1, the values of $V_{\mathrm{P}}(z)$ and v are monitored. The system switches from $q = 1$ to $q = 0$ if ever $V_{\mathrm{P}}(z) \geq v$, since this indicates that κ_1 has eliminated the buffer between $V_{\mathrm{P}}(z)$ and v. While $q = 1$ and either $V_{\mathrm{P}}(z) \leq v$, the system flows, holding $q = 1$, meaning the system continues to use κ_1.

After each switch to $q = 1$, a subsequent switch back to $q = 0$ indicates that κ_1 caused $V_{\mathrm{P}}(z)$ to either increase or decrease less quickly than v over an interval of time. The buffer between $V_{\mathrm{P}}(z)$ and v upon a switch to $q = 1$ permits the uncertified controller to increase the value of $V_{\mathrm{P}}(z)$ briefly. The assumptions on f_{v}, however, ensure that v converges to zero, so $V_{\mathrm{P}}(z)$ is either squeezed to zero or a switch to $q = 0$ is eventually triggered when $V_{\mathrm{P}}(z) = v$ occurs.

Before we formulate the hybrid closed-loop system, we demonstrate the switching logic with a toy example.

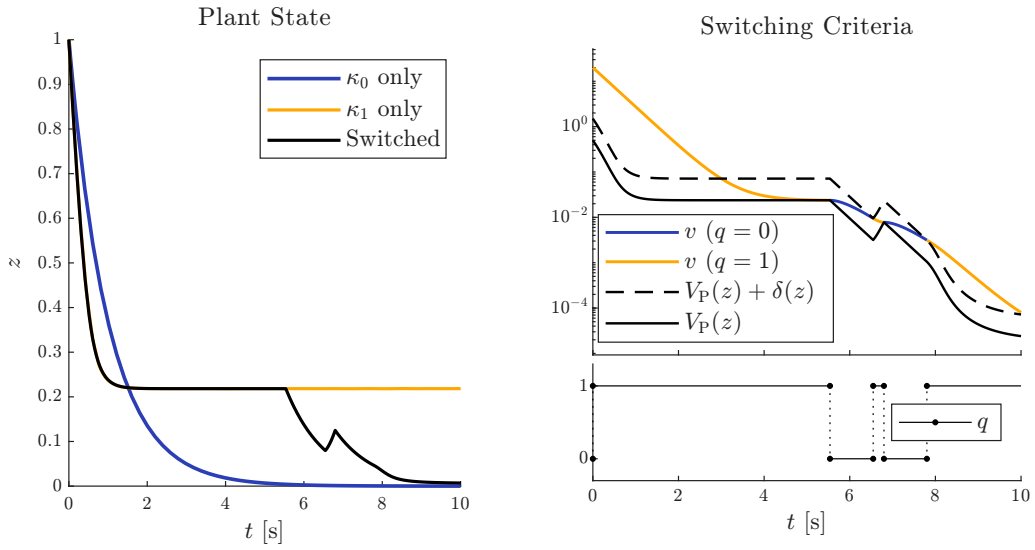

Figure 3.2. Trajectories of z (left) from Example 3.1 using κ_0 only, κ_1, and opportunistic switching between κ_0 and κ_1, and trajectories of q, v, $V_P(z) + \delta(z)$, and $V_P(z)$ (right), which determine switches.

Example 3.1. Consider the plant $\dot{z} = u$ with $z, u \in \mathbb{R}$ and controllers $\kappa_0(z) := -z$ and $\kappa_1(z) := -2z \sin(1/(z + 0.1))$. The controller κ_0 is certified to render the set $\mathcal{A}_P := \{0\}$ UGAS by he Lyapunov function $z \mapsto V_P(z) := \frac{1}{2}z^2$ with $\sigma(z) := z^2$. For the supervisor parameters, let $\gamma := 1/2$, $\mu := 2$, and $\delta(z) := |z|^2$. Figure 3.2 shows solutions to $\dot{z} = \kappa_0(z)$ and $\dot{z} = \kappa_1(z)$ (without switching), and to $\dot{z} = \kappa_q(z)$ with q switching according to our hybrid control strategy.[3] Initially, the solution with the feedback κ_1 decreases quickly, but it fails to converge to zero, instead becoming stuck above $z = 0.2$. On the other hand, the solution with the certified feedback κ_0 converges to the origin. For the solution with switching, v reaches $V_P(z)$ around $t = 6\,\text{sec}$, which triggers a switch to κ_0. After $t = 6\,\text{sec}$, $V_P(z)$ decreases quickly until the buffer between V_P and v is larger than $\delta(z)$, triggering a switch back to κ_1. ◇

[3] Simulations are computed in MATLAB with the *HyEQ Toolbox* [14].

3.2 Construction of the Closed-Loop System with Static Feedback

We are now ready to define the hybrid closed-loop system. If $q = 0$, jumps occur when $x = (z, v, q)$ is in

$$D_{0\mapsto 1} := \{(z, v, 0) \in \mathcal{X} \mid V_{\mathrm{P}}(z) + \delta(z) \leq v\} \tag{3.5}$$

and flows occur when x is in

$$C_0 := \{(z, v, 0) \in \mathcal{X} \mid V_{\mathrm{P}}(z) + \delta(z) \geq v\}. \tag{3.6}$$

Similarly, if $q = 1$, then the system jumps when x is in

$$D_{1\mapsto 0} := \{(z, v, 1) \in \mathcal{X} \mid V_{\mathrm{P}}(z) \geq v\} \tag{3.7}$$

and flows when x is in

$$C_1 := \{(z, v, 1) \in \mathcal{X} \mid V_{\mathrm{P}}(z) \leq v\} \tag{3.8}$$

The jump set is then defined as $D := D_{0\mapsto 1} \cup D_{1\mapsto 0}$ and the flow set is $C := C_0 \cup C_1$. Note that the flow set is the closed complement of the jump set, i.e., $C = \overline{\mathcal{X} \setminus D}$.

At each jump, z is constant, since the plant state is continuous in time; v is set to $v^+ = \max\{V_{\mathrm{P}}(z), v\}$ to record the of $V_{\mathrm{P}}(z)$ if it is larger than v. Due to the design of $D_{0\mapsto 1}$ and $D_{1\mapsto 0}$, the value of v after a switch to $q = 0$ is always $V_{\mathrm{P}}(z)$ and the value of v does not change at a switch to $q = 1$. The mode q is toggled to the opposite value in $\{0, 1\}$. During flows, z evolves according to $\dot{z} = f_{\mathrm{P}}(z, \kappa_q(z))$, the auxiliary variable v evolves according to $\dot{v} = f_{\mathsf{v}}(z, v)$, and the logic variable q is held constant ($\dot{q} = 0$).

The construction above leads to the hybrid closed-loop system $\mathcal{H} = (C, f, D, g)$ with state $x = (z, v, q) \in \mathcal{X}$ and data given by

$$\begin{cases} f(x) := \begin{bmatrix} f_q(z) \\ f_{\mathsf{v}}(x) \\ 0 \end{bmatrix} & \forall x \in C := C_0 \cup C_1 \\ g(x) := \begin{bmatrix} z \\ \max\{V_{\mathrm{P}}(z), v\} \\ 1 - q \end{bmatrix} & \forall x \in D := D_{0\mapsto 1} \cup D_{1\mapsto 0} \end{cases} \tag{3.9}$$

Proposition 3.1. *Suppose that*

1. f_{P}, κ_0, κ_1, *and* δ *are continuous;*
2. f_{P} *and* κ_0 *satisfy Assumption 3.1 for* $\mathcal{A}_{\mathrm{P}}$ *with Lyapunov function* V_{P}, *and*
3. $\delta(z) > 0$ *for all* $z \in \mathbb{R}$, $\gamma > 0$, *and* $\mu > 0$

Then, the set $\mathcal{A}$ *in* (3.3) *is UGAS for* $\mathcal{H}$ *given in* (3.9) *and for each solution* φ *to* $\mathcal{H}$,

$$\sup_t \operatorname{dom}(\varphi) = \infty,$$

and there exists a minimum dwell time $d_{\min} > 0$ *between switches.*

The proof Proposition 3.1 is conducted by constructing a new Lyapunov function for the closed-loop system, namely

$$V(x) := \max\{V_{\mathrm{P}}(z),\, v\}.$$

We defer the proof until Chapter 4 where we consider a more general case of the control strategy in this chapter.

Example 3.2 (LQR)**.** Consider the nonlinear

$$\dot{z} = \left(\underbrace{\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}}_{=:A_1} + \min\{|z|, 1\} \underbrace{\begin{bmatrix} -2 & 0 \\ 4 & -1 \end{bmatrix}}_{=:A_2} \right) z + \underbrace{\begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}}_{=:B} u, \tag{3.10}$$

and the set $\mathcal{A}_{\mathrm{P}} := \{0\}$. This system behaves like $\dot{z} = A_1 z + Bu$ near the origin and like $\dot{z} = (A_1 + A_2)z + Bu$ far from it. The origin of (3.10) is UGAS for $\kappa_0(z) := \left[\begin{smallmatrix} -5 & 0 \\ 0 & -6 \end{smallmatrix}\right] z$. For κ_1, we linearize (3.10) about the origin and use the linear quadratic regulator (LQR) feedback that solves the following infinite-horizon optimal control problem:

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \int_0^\infty |z(t)|^2 + |u(t)|^2 \, dt \\ \text{subject to} \quad & \dot{z} = A_1 z + Bu. \end{aligned} \tag{3.11}$$

The LQR feedback is

$$\kappa_1(z) := -\begin{bmatrix} -1.82 & -0.50 \\ -0.61 & -0.87 \end{bmatrix} z.$$

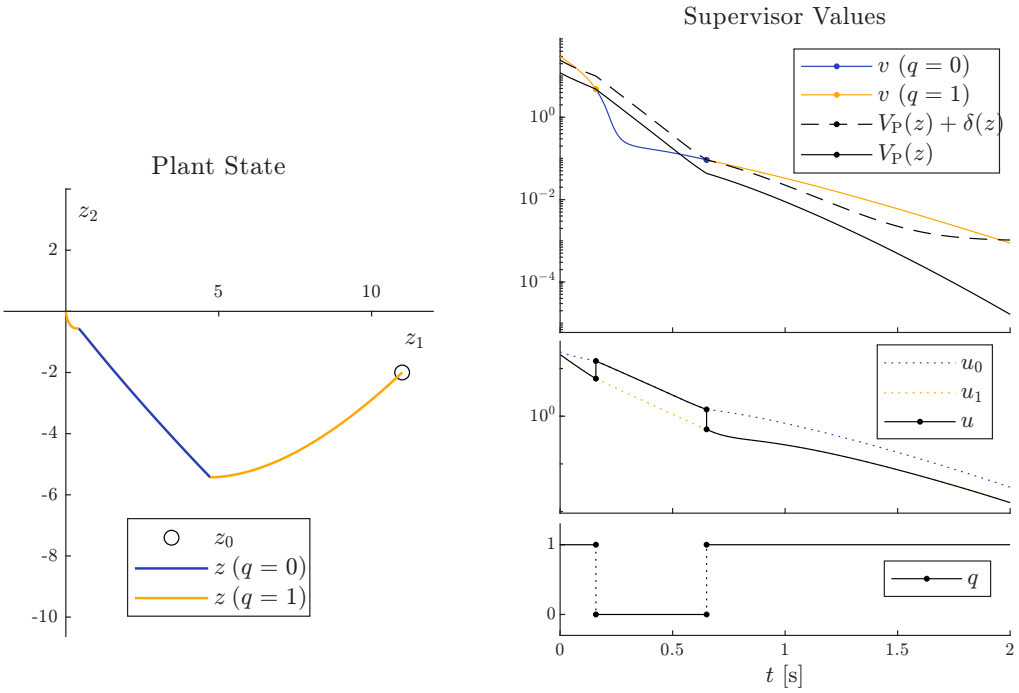

Figure 3.3. Trajectories for the plant and supervisor states in Example 3.2 starting from $z_0 := (11, 2)$ and $v_0 := 30$.

Figure 3.3 shows a solution to the hybrid closed-loop system using f_{v} from (3.4) with $\gamma = 2$ and $\mu = 4$, and $\delta(z) = \frac{1}{10}|z|^2 + 10^{-3}$. The switching logic uses κ_1 near the origin, reducing $|u|$. The buffer between v and $V_{\mathrm{P}}(z)$ allows $V_{\mathrm{P}}(z)$ to briefly slow without triggering a switch to $q = 0$. The switch is followed by a spike in control effort, a period of faster convergence, and a subsequent switch back to $q = 1$. ◇

Chapter 4

Uniting Feedback with Hybrid Controllers and a Hybrid Plant

In this chapter, we extend the hybrid control strategy that allows for uniting feedback between a certified controller and one that is uncertified (in some sense) to the case of systems with hybrid controllers and plants with set-valued dynamics. Specifically, we extend the strategy in Chapter 3 for uniting feedback to achieve asymptotic stability from the case of pure feedback ("memoryless") controllers, to a strategy for plants and controllers modeled as hybrid systems with inputs, including set-valued dynamics given by differential inclusions and difference inclusions.

4.1 Uniting Feedback for Hybrid Plant with Hybrid Controllers

In this section, we consider the case where the plant, certified controller, and uncertified controller are all hybrid inclusions with inputs, as described in [3]. The closed-loop system $\mathcal{H}$, depicted in Figure 4.1, is composed a plant $\mathcal{H}_{\mathrm{P}}$, controllers $\mathcal{H}_{\mathrm{K}_0}$ and $\mathcal{H}_{\mathrm{K}_1}$, a supervisor $\mathcal{H}_{\mathrm{S}}$, and a switch. The system is designed under the assumption that $\mathcal{H}_{\mathrm{K}_0}$ is certified to achieve some property whereas $\mathcal{H}_{\mathrm{K}_1}$ is uncertified. The state of the closed-loop system is $x := (z, \eta_0, \eta_1, v, q)$ where $z \in \mathbb{R}^{n_{\mathrm{P}}}$, $\eta_0 \in \mathbb{R}^{n_0}$, $\eta_1 \in \mathbb{R}^{n_1}$, and $(v, q) \in \mathbb{R}^{n_{\mathrm{S}}} \times \{0, 1\}$ are the states of the respective subsystems $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, and $\mathcal{H}_{\mathrm{S}}$. To allow for restricted state spaces for each subsystem, we define $\mathcal{E}_{\mathrm{P},0} \subset \mathbb{R}^{n_{\mathrm{P}}} \times \mathbb{R}^{n_0}$, $\mathcal{E}_1 \subset \mathbb{R}^{n_1}$, and $\mathcal{V} \subset \mathbb{R}^{n_{\mathrm{S}}}$ such that $(z, \eta_0) \in \mathcal{E}_{\mathrm{P},0}$, $\eta_1 \in \mathcal{E}_1$, and $v \in \mathcal{V}$. The closed-loop state x has dimension $n := n_{\mathrm{P}} + n_0 + n_1 + n_{\mathrm{S}} + 1$ and belongs

to a set $\mathcal{X} \subset \mathbb{R}^n$ defined as

$$\mathcal{X} := \mathcal{E}_{\mathrm{P},0} \times \mathcal{E}_1 \times \mathcal{V} \times \{0,1\}. \tag{4.1}$$

To facilitate discussion of the separate modes, let

$$\mathcal{X}_0 := \{(z, \eta_0, \eta_1, v, q) \in \mathcal{X} \mid q = 0\} \quad \text{and} \quad \mathcal{X}_1 := \{(z, \eta_0, \eta_1, v, q) \in \mathcal{X} \mid q = 1\}.$$

The subsystems are written as

$$\mathcal{H}_{\mathrm{P}} : \begin{cases} \dot{z} \in F_{\mathrm{P}}(z,u) \quad (z,u) \in C_{\mathrm{P}} & \text{input: } u \\ z^+ \in G_{\mathrm{P}}(z,u) \quad (z,u) \in D_{\mathrm{P}}. & \text{state: } z \end{cases} \tag{4.2a}$$

$$\mathcal{H}_{\mathrm{K}_0} : \begin{cases} \dot{\eta}_0 \in F_{\mathrm{K}_0}(z,\eta_0) \quad (z,\eta_0) \in C_{\mathrm{K}_0} & \text{input: } z \\ \eta_0^+ \in G_{\mathrm{K}_0}(z,\eta_0) \quad (z,\eta_0) \in D_{\mathrm{K}_0} & \text{state: } \eta_0 \\ u_0 = \kappa_0(z,\eta_0). & \text{output: } u_0 \end{cases} \tag{4.2b}$$

$$\mathcal{H}_{\mathrm{K}_1} : \begin{cases} \dot{\eta}_1 \in F_{\mathrm{K}_1}(x) \quad x \in C_{\mathrm{K}_1} & \text{inputs: } z,\ \eta_0,\ v,\ q \\ \eta_1^+ \in G_{\mathrm{K}_1}(x) \quad x \in D_{\mathrm{K}_1} & \text{state: } \eta_1 \\ u_1 = \kappa_1(x). & \text{output: } u_1 \end{cases} \tag{4.2c}$$

$$\mathcal{H}_{\mathrm{S}} : \begin{cases} \begin{bmatrix} \dot{v} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} f_{\mathsf{v}}(x) \\ 0 \end{bmatrix} \quad x \in C_{\mathrm{S}} & \text{inputs: } z,\ \eta_0,\ \eta_1 \\ \begin{bmatrix} v^+ \\ q^+ \end{bmatrix} = \begin{bmatrix} g_{\mathsf{v}}(x) \\ 1-q \end{bmatrix} \quad x \in D_{\mathrm{S}} & \text{state: } v,\ q. \end{cases} \tag{4.2d}$$

The output $q \in \{0,1\}$ from $\mathcal{H}_{\mathrm{S}}$ determines whether the plant $\mathcal{H}_{\mathrm{P}}$ receives its input from $\mathcal{H}_{\mathrm{K}_0}$ or $\mathcal{H}_{\mathrm{K}_1}$. The supervisor's flow and jump sets are defined as

$$C_{\mathrm{S}} := (\mathcal{S}_0 \times \{0\}) \cup (\mathcal{S}_1 \times \{1\}) \quad \text{and} \quad D_{\mathrm{S}} := (\mathcal{S}_{0\mapsto 1} \times \{0\}) \cup (\mathcal{S}_{1\mapsto 0} \times \{1\}),$$

where $\mathcal{S}_{0\mapsto 1}$ and $\mathcal{S}_{1\mapsto 0}$ are called the *switching sets* of $\mathcal{H}_{\mathrm{S}}$ and $\mathcal{S}_0$ and $\mathcal{S}_1$ are called the *hold sets* of $\mathcal{H}_{\mathrm{S}}$. The switching set $\mathcal{S}_{0\mapsto 1}$ and $\mathcal{S}_{1\mapsto 0}$ determine where q switches from 0 to 1, and from 1 to 0, respectively. The hold set $\mathcal{S}_0$ determines when the supervisor is allowed to continue using $q = 0$, if q is already 0, whereas $\mathcal{S}_1$ determines where $\mathcal{H}_{\mathrm{S}}$ will continue using $q = 1$. The functions f_{v} and g_{v} define the dynamics of the supervisor's auxiliary variable, v. One choice of supervisor to achieve uniform global asymptotic stability is given in Section 4.2.

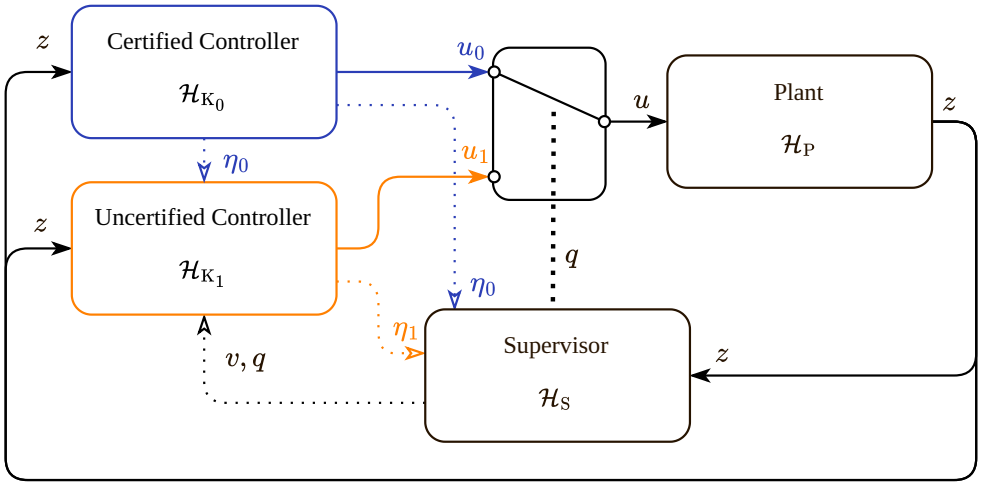

Figure 4.1. Feedback diagram for the hybrid closed-loop system $\mathcal{H}$ using a hybrid plant $\mathcal{H}_P$, hybrid controllers $\mathcal{H}_{K_0}$ and $\mathcal{H}_{K_1}$, and supervisor $\mathcal{H}_S$.

The controller $\mathcal{H}_{K_0}$ will be called the *certified* controller because we assume the existence of a (barrier or Lyapunov) certificate function for the closed-loop system formed from $\mathcal{H}_P$ and $\mathcal{H}_{K_0}$. On the other hand, $\mathcal{H}_{K_1}$ will be called the *uncertified* controller, because no such assumption is made.

By passing (z, η_0, v, q) to $\mathcal{H}_{K_1}$, we permit the uncertified controller to be designed with it full knowledge of the state of all of the subsystems. In practice, the inputs used by $\mathcal{H}_{K_1}$ will typically be limited to z, but allowing $\mathcal{H}_{K_1}$ access to the full state of the closed-loop system allows designers to exploit knowledge of other subsystems. We have two examples of how this can be useful. If the uncertified controller is a learning-based controller, then the certified controller can be used as an expert demonstration to perform online training of the uncertified controller while $q = 0$. Alternatively, if the uncertified controller is computationally expensive, then it may be advantageous to stop the computations of the control values while $q = 0$ (when the certified controller is used) and only run expensive computations when $q = 1$.

Since each subsystem $\mathcal{H}_P$, $\mathcal{H}_{K_0}$, $\mathcal{H}_{K_1}$, and $\mathcal{H}_S$ has different sized inputs, states, and outputs, it is notationally convenient to standardize the dimensions of the data for each subsystem. To this end, we define $\kappa : \mathcal{X} \to \mathcal{U}$ such that $\kappa(x)$ gives the control value at any $x := (z, \eta_0, \eta_1, v, q) \in \mathcal{X}$, selecting the control from $\mathcal{H}_{K_0}$ and $\mathcal{H}_{K_1}$

depending on the value of q according to

$$\kappa(x) := \begin{cases} \kappa_0(z, \eta_0) & \text{if } q = 0 \\ \kappa_1(x) & \text{if } q = 1. \end{cases} \tag{4.3}$$

For each subsystem, we define the closed-loop (CL) flow maps $F_{\mathrm{P}}^{\mathrm{CL}} : \mathcal{X} \rightrightarrows \mathbb{R}^{n_{\mathrm{P}}}$, $F_{\mathrm{K}_0}^{\mathrm{CL}} : \mathcal{X} \rightrightarrows \mathbb{R}^{n_0}$, $F_{\mathrm{K}_1}^{\mathrm{CL}} : \mathcal{X} \rightrightarrows \mathbb{R}^{n_1}$, and $F_{\mathrm{S}}^{\mathrm{CL}} : \mathcal{X} \rightrightarrows \mathbb{R}^{n_{\mathrm{S}}} \times \{0\}$ as

$$F_{\mathrm{P}}^{\mathrm{CL}}(x) := F_{\mathrm{P}}\big(z, \kappa(x)\big), \qquad F_{\mathrm{K}_0}^{\mathrm{CL}}(x) := F_{\mathrm{K}_0}(z, \eta_0), \tag{4.4}$$

$$F_{\mathrm{K}_1}^{\mathrm{CL}}(x) := F_{\mathrm{K}_1}(x), \qquad F_{\mathrm{S}}^{\mathrm{CL}}(x) := \begin{bmatrix} f_{\mathsf{v}}(x) \\ 0 \end{bmatrix}. \tag{4.5}$$

Similarly, we define

$$G_{\mathrm{P}}^{\mathrm{CL}}(x) := \begin{bmatrix} G_{\mathrm{P}}(z, \kappa(x)) \\ \eta_0 \\ \eta_1 \\ v \\ q \end{bmatrix}, \qquad G_{\mathrm{K}_0}^{\mathrm{CL}}(x) := \begin{bmatrix} z \\ G_{\mathrm{K}_0}(z, \eta_0) \\ \eta_1 \\ v \\ q \end{bmatrix}, \tag{4.6}$$

$$G_{\mathrm{K}_1}^{\mathrm{CL}}(x) := \begin{bmatrix} z \\ \eta_0 \\ G_{\mathrm{K}_1}(x) \\ v \\ q \end{bmatrix}, \qquad G_{\mathrm{S}}^{\mathrm{CL}}(x) := \begin{bmatrix} z \\ \eta_0 \\ \eta_1 \\ g_{\mathsf{v}}(x) \\ 1 - q \end{bmatrix} \tag{4.7}$$

To simplify the construction of the closed-loop system, we will impose that

$$\operatorname{dom}(F_{\mathrm{P}}) = C_{\mathrm{P}} \quad \text{and} \quad \operatorname{dom}(G_{\mathrm{P}}) = D_{\mathrm{P}},$$

which be achieved by restricting the domain of given functions. As a result of this assumption,

$$F_{\mathrm{P}}(z, u) \neq \varnothing \iff (z, u) \in C_{\mathrm{P}}.$$

For $x \notin \operatorname{dom}(f_{\mathsf{v}})$, we define $F_{\mathrm{S}}^{\mathrm{CL}}(x) = \varnothing$ and for $x \notin \operatorname{dom}(g_{\mathsf{v}})$, we define $G_{\mathrm{S}}^{\mathrm{CL}}(x) = \varnothing$. It is worth noting that vector produced by each $F_{\star}^{\mathrm{CL}}$ matches the state dimension of the subsystem $\star$, whereas the output of each $G_{\star}^{\mathrm{CL}}$ matches the dimension n of the entire closed-loop system. This difference arises from the fact that the state of all of the subsystems flow at the same time, but jump individually, so, for example, $F_{\mathrm{P}}^{\mathrm{CL}}$ only defines the flow for the plant $\mathcal{H}_{\mathrm{P}}$ as $\dot{z} \in F_{\mathrm{P}}^{\mathrm{CL}}(x)$, whereas $G_{\mathrm{P}}^{\mathrm{CL}}$ defines how all of

the subsystem states jump, according to $z^+ \in G_{\mathrm{P}}(x)$, but all of the components in x are unchanged except z.

The closed-loop subsystem flow and jump sets are given by

$$\begin{aligned}
C_{\mathrm{P}}^{\mathrm{CL}} &:= \overline{\{x \in \mathcal{X} \mid (z, \kappa(x)) \in C_{\mathrm{P}}\}}, & C_{\mathrm{K}_0}^{\mathrm{CL}} &:= C_{\mathrm{K}_0} \times \mathcal{E}_1 \times \mathcal{V} \times \{0,1\}, \\
D_{\mathrm{P}}^{\mathrm{CL}} &:= \overline{\big\{x \in \mathcal{X} \bigm| \big(z, \kappa(x)\big) \in D_{\mathrm{P}}\big\}}, & D_{\mathrm{K}_0}^{\mathrm{CL}} &:= D_{\mathrm{K}_0} \times \mathcal{E}_1 \times \mathcal{V} \times \{0,1\}, \\
C_{\mathrm{K}_1}^{\mathrm{CL}} &:= C_{\mathrm{K}_1}, & C_{\mathrm{S}}^{\mathrm{CL}} &:= C_{\mathrm{S}}, \\
D_{\mathrm{K}_1}^{\mathrm{CL}} &:= D_{\mathrm{K}_1}, & D_{\mathrm{S}}^{\mathrm{CL}} &:= D_{\mathrm{S}}.
\end{aligned} \tag{4.8}$$

The closed-loop system is

$$\mathcal{H}: \begin{cases} C := C_{\mathrm{P}}^{\mathrm{CL}} \cap C_{\mathrm{K}_0}^{\mathrm{CL}} \cap C_{\mathrm{K}_1}^{\mathrm{CL}} \cap C_{\mathrm{S}}^{\mathrm{CL}}, & F(x) := F_{\mathrm{P}}^{\mathrm{CL}}(x) \times F_{\mathrm{K}_0}^{\mathrm{CL}}(x) \times F_{\mathrm{K}_1}^{\mathrm{CL}}(x) \times F_{\mathrm{S}}^{\mathrm{CL}}(x), \\ D := D_{\mathrm{P}}^{\mathrm{CL}} \cup D_{\mathrm{K}_0}^{\mathrm{CL}} \cup D_{\mathrm{K}_1}^{\mathrm{CL}} \cup D_{\mathrm{S}}^{\mathrm{CL}}, & G(x) := G_{\mathrm{P}}^{\mathrm{CL}}(x) \cup G_{\mathrm{K}_0}^{\mathrm{CL}}(x) \cup G_{\mathrm{K}_1}^{\mathrm{CL}}(x) \cup G_{\mathrm{S}}^{\mathrm{CL}}(x). \end{cases} \tag{4.9}$$

The closures in the definitions of $C_{\mathrm{P}}^{\mathrm{CL}}$ and $D_{\mathrm{P}}^{\mathrm{CL}}$ are necessary to ensure the sets are closed, since κ_1 is not assumed to be continuous. Compare the set

$$\big\{x \in \mathcal{X} \bigm| \big(z, \kappa(x)\big) \in D_{\mathrm{P}}\big\},$$

which may not include all of its boundary points (causing it to not be closed), even if C_{P} is closed, due to discontinuities in κ_1. By the construction of each $G_\star^{\mathrm{CL}}$, along with (B3) in Assumption 4.1 (below), we have that $\mathrm{dom}(G_\star^{\mathrm{CL}}) = D_\star^{\mathrm{CL}}$ for each $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$. Thus, the state $\eta_\star$ of subsystem $\star$ only jumps according to $\eta_\star^+ \in G_\star^{\mathrm{CL}}(x)$ if $x \in D_\star^{\mathrm{CL}}$. More compactly, let $\mathcal{S} := \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$. Then,

$$F(x) := \prod_{\star \in \mathcal{S}} F_\star^{\mathrm{CL}}(x), \quad C(x) := \bigcap_{\star \in \mathcal{S}} C_\star^{\mathrm{CL}}, \quad G(x) := \bigcup_{\star \in \mathcal{S}} G_\star^{\mathrm{CL}}(x), \quad D(x) := \bigcup_{\star \in \mathcal{S}} D_\star^{\mathrm{CL}}.$$

It is convenient to define the closed-loop system composed of only the plant $\mathcal{H}_{\mathrm{P}}$ and the certified controller $\mathcal{H}_{\mathrm{K}_0}$, since $\mathcal{H}_{\mathrm{K}_0}$ is, by assumption, certified to produce some property in this system.

$$F_{\mathrm{P}\times 0}(z, \eta_0) := \begin{bmatrix} F_{\mathrm{P}}\big(z, \kappa_0(z, \eta_0)\big) \\ F_{\mathrm{K}_0}(z, \eta_0) \end{bmatrix} \tag{4.10}$$

$$G_{\mathrm{P}\times 0}(z, \eta_0) := \begin{bmatrix} G_{\mathrm{P}}\big(z, \kappa_0(z, \eta_0)\big) \\ \eta_0 \end{bmatrix} \cup \begin{bmatrix} z \\ G_{\mathrm{K}_0}(z, \eta_0) \end{bmatrix} \tag{4.11}$$

$$C_{\mathrm{P}\times 0} := \big\{(z, \eta_0) \bigm| \big(z, \kappa_0(z, \eta_0)\big) \in C_{\mathrm{P}},\ (z, \eta_0) \in C_{\mathrm{K}_0}\big\} \tag{4.12}$$

$$D_{\mathrm{P}\times 0} := \big\{(z, \eta_0) \bigm| \big(z, \kappa_0(z, \eta_0)\big) \in D_{\mathrm{P}} \text{ or } (z, \eta_0) \in D_{\mathrm{K}_0}\big\}. \tag{4.13}$$

The closed loop system formed from $\mathcal{H}_{\mathrm{P}}$ and $\mathcal{H}_{\mathrm{K}_0}$ (without any switching between controllers) is

$$\mathcal{H}_{\mathrm{P}\times 0} : \begin{cases} \left(\dot{z},\ \dot{\eta}_0\right) \in F_{\mathrm{P}\times 0}(z,\eta_0) & (z,\eta_0) \in C_{\mathrm{P}\times 0} \\ \left(z^+,\ \eta_0^+\right) \in G_{\mathrm{P}\times 0}(z,\eta_0) & (z,\eta_0) \in D_{\mathrm{P}\times 0}. \end{cases} \tag{4.14}$$

Example 4.1 (Systems in Chapter 3)**.** We can model the closed-loop system presented Chapter 3 using the model presented above. The plant in Chapter 3 is a continuous-time system and the controllers are static feedback controllers, which we write as

$$\mathcal{H}_{\mathrm{P}}\colon \begin{cases} \dot{z} = F_{\mathrm{P}}(z,u) := f_{\mathrm{P}}(z,u), \end{cases} \quad \mathcal{H}_{\mathrm{K}_0}\colon \begin{cases} u_0 = \kappa_0(z) \end{cases} \quad \text{and} \quad \mathcal{H}_{\mathrm{K}_1}\colon \begin{cases} u_1 = \kappa_1(z). \end{cases}$$

The dynamics for $\mathcal{H}_{\mathrm{K}_0}$ and $\mathcal{H}_{\mathrm{K}_1}$ are omitted because $n_0 = n_1 = 0$. The supervisor Chapter 3 has an internal state, and would be written as in (4.2d) with f_{v} given in (3.4), $g_{\mathsf{v}}(z,v) := \max\{V_{\mathrm{P}}(z), v\}$ and

$$\begin{aligned} \mathcal{S}_{0\mapsto 1} &= \{(z,v) \mid V_{\mathrm{P}}(z) + \delta(z) \le v\} & \mathcal{S}_{1\mapsto 0} &= \{(z,v) \mid V_{\mathrm{P}}(z) \ge v\} \\ \mathcal{S}_0 &= \{(z,v) \mid V_{\mathrm{P}}(z) + \delta(z) \ge v\} & \mathcal{S}_1 &= \{(z,v) \mid V_{\mathrm{P}}(z) \le v\}. \end{aligned}$$

The certified closed-loop system $\mathcal{H}_{\mathrm{P}\times 0}$ in (4.14) reduces to

$$\mathcal{H}_{\mathrm{P}\times 0} : \begin{cases} \dot{z} = F_{\mathrm{P}\times 0}(z) := f_{\mathrm{P}}(z, \kappa_0(z)). \end{cases} \qquad \diamond$$

The next several sections establish certain desirable properties for the closed-loop system $\mathcal{H}$ in the case of a generic supervisor. In Section 4.1.1, $\mathcal{H}$ is regularized to produce a system $\widehat{\mathcal{H}}$ that satisfies the hybrid basic conditions under given assumptions. Sections 4.1.2 and 4.1.3 establish the existence of solutions for all $t \ge 0$. By establishing these fundamental properties in the generic case, we can utilize them for specific choices of supervisors, as we do in Section 4.2 where we design a supervisor for global asymptotic stability.

4.1.1 Regularity of the Closed-loop System

We want to keep the assumptions on $\mathcal{H}_{\mathrm{K}_1}$ as weak as possible, so we do not impose any sort of continuity assumptions on κ_1, F_{K_1} and G_{K_1}. As a result, $\mathcal{H}$ may violate the hybrid basic conditions (Definition 1.2). Analyzing systems without

the hybrid basic conditions is difficult, however. To mitigate, we perform system regularization to construct a new system $\widehat{\mathcal{H}}$ such that $\widehat{\mathcal{H}}$ satisfies the hybrid basic conditions, and every solution to $\mathcal{H}$ is also a solution to $\widehat{\mathcal{H}}$. The second point is important because it allows us to prove certain properties for $\widehat{\mathcal{H}}$ (e.g., global asymptotic stability) and infer that $\mathcal{H}$ has the same property.

In particular, we define the regularized system as $\widehat{\mathcal{H}} := (C, \widehat{F}, D, \widehat{G})$ with $\widehat{F}$ and $\widehat{G}$ defined identically to F and G except that $F_{\mathrm{P}}^{\mathrm{CL}}$, $G_{\mathrm{P}}^{\mathrm{CL}}$, $F_{\mathrm{K}_1}^{\mathrm{CL}}$, $G_{\mathrm{K}_1}^{\mathrm{CL}}$, are replaced by the following regularization:

$$\widehat{F_{\mathrm{P}}^{\mathrm{CL}}}(x) = \begin{cases} F_{\mathrm{P}}^{\mathrm{CL}}(x) & \text{if } q = 0 \\ \bigcap_{\delta>0} \overline{\mathrm{conv}}\big(F_{\mathrm{P}}^{\mathrm{CL}}(x + \delta\mathbb{B})\big) & \text{if } q = 1, \end{cases} \tag{4.15a}$$

$$\widehat{G_{\mathrm{P}}^{\mathrm{CL}}}(x) = \begin{cases} G_{\mathrm{P}}^{\mathrm{CL}}(x) & \text{if } q = 0 \\ \bigcap_{\delta>0} \overline{\big(G_{\mathrm{P}}^{\mathrm{CL}}(x + \delta\mathbb{B})\big)} & \text{if } q = 1, \end{cases} \tag{4.15b}$$

$$\widehat{F_{\mathrm{K}_1}^{\mathrm{CL}}}(x) = \bigcap_{\delta>0} \overline{\mathrm{conv}}\big(F_{\mathrm{K}_1}^{\mathrm{CL}}(x + \delta\mathbb{B})\big), \tag{4.15c}$$

$$\widehat{G_{\mathrm{K}_1}^{\mathrm{CL}}}(x) = \bigcap_{\delta>0} \overline{\big(G_{\mathrm{K}_1}^{\mathrm{CL}}(x + \delta\mathbb{B})\big)}. \tag{4.15d}$$

To allow us to use a uniform notation for all of the functions and sets, we also write $\widehat{F_{\star}^{\mathrm{CL}}} := F_{\star}^{\mathrm{CL}}$ and $\widehat{G_{\star}^{\mathrm{CL}}} := G_{\star}^{\mathrm{CL}}$ for each $\star \in \{\mathrm{K}_0, \mathrm{S}\}$. The functions $\widehat{F_{\mathrm{P}}^{\mathrm{CL}}}$, $\widehat{F_{\mathrm{K}_1}^{\mathrm{CL}}}$, $\widehat{G_{\mathrm{P}}^{\mathrm{CL}}}$, and $\widehat{G_{\mathrm{K}_1}^{\mathrm{CL}}}$ are OSC (see [4, Lemma 5.16]), and $\widehat{F_{\mathrm{P}}^{\mathrm{CL}}}(x)$ and $\widehat{F_{\mathrm{K}_1}^{\mathrm{CL}}}(x)$ are convex for each $x \in \mathcal{X}$. The necessary properties of the remaining functions and sets are achieved via assumptions. In particular, we impose the following assumptions on $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, and $\mathcal{H}_{\mathrm{K}_1}$ to ensure that $\widehat{\mathcal{H}}$ satisfies the hybrid basic conditions (Definition 1.2).

Assumption 4.1. The subsystems $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, and $\mathcal{H}_{\mathrm{S}}$ in (4.2) satisfy the following.

(B1) (*Ensure $C \cup D = \mathcal{X}$*):

$$\big(z, \kappa(x)\big) \in C_{\mathrm{P}} \cup D_{\mathrm{P}} \qquad \forall x := (z, \eta_0, \eta_1, v, q) \in \mathcal{X}$$
$$C_{\mathrm{K}_0} \cup D_{\mathrm{K}_0} = \mathcal{E}_{\mathrm{P},0}; \quad C_{\mathrm{K}_1} \cup D_{\mathrm{K}_1} = \mathcal{X}; \quad \text{and} \quad C_{\mathrm{S}} \cup D_{\mathrm{S}} = \mathcal{X}.$$

(B2) (*Ensure $\widehat{C}$ and $\widehat{D}$ closed*): The sets $\mathcal{E}_1$, $\mathcal{V}$, $C_\star$, and $D_\star$ are closed for each $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$.

(B3) (*Ensure* $\operatorname{dom}(\widehat{F}) = C$ *and* $\operatorname{dom}(\widehat{G}) = D$):

$$\begin{aligned}
&\operatorname{dom}(\kappa_0) = \mathcal{E}_{\mathrm{P},0}, \quad \operatorname{dom}(\kappa_1) = \mathcal{X}, \\
&\operatorname{dom}(F_{\mathrm{P}}) = C_{\mathrm{P}}, \quad \operatorname{dom}(F_{\mathrm{K}_0}) = C_{\mathrm{K}_0}, \quad \operatorname{dom}(F_{\mathrm{K}_1}) = C_{\mathrm{K}_1}, \quad \operatorname{dom}(f_{\mathsf{v}}) = C_{\mathrm{S}}, \\
&\operatorname{dom}(G_{\mathrm{P}}) = D_{\mathrm{P}}, \quad \operatorname{dom}(G_{\mathrm{K}_0}) = D_{\mathrm{K}_0}, \quad \operatorname{dom}(G_{\mathrm{K}_1}) = D_{\mathrm{K}_1}, \quad \operatorname{dom}(g_{\mathsf{v}}) = D_{\mathrm{S}}.
\end{aligned}$$

(B4) (*Ensure* $\widehat{F}$ *and* $\widehat{G}$ *are OSC*): The functions f_{v}, and g_{v} are continuous, and the maps $(z, \eta_0) \mapsto F_{\mathrm{P}}(z, \kappa_0(z, \eta_0))$, $(z, \eta_0) \mapsto G_{\mathrm{P}}(z, \kappa_0(z, \eta_0))$, F_{K_0}, and G_{K_0} are OSC.

(B5) (*Ensure* $\widehat{F}$ *and* $\widehat{G}$ *are locally bounded*): The functions κ_1, $F_\star$, and $G_\star$ are locally bounded for each $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1\}$.

(B6) (*Ensure* $\widehat{F}$ *is pointwise–convex*): The set $F_{\mathrm{P}}(z, u)$ is convex for each $(z, u) \in C_{\mathrm{P}}$, and $F_{\mathrm{K}_0}(z, \eta_0)$ is convex for each $(z, \eta_0) \in C_{\mathrm{K}_0}$. ◇

Under the assumptions of Assumption 4.1, we assert basic properties of the closed-loop system.

Lemma 4.1. *Suppose* $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, *and* $\mathcal{H}_{\mathrm{S}}$ *satisfy Assumption 4.1. Then,* $\widehat{\mathcal{H}}$ *satisfies the hybrid basic conditions (Definition 1.2).*

The proof of Lemma 4.1 is in Section B.1.1.

4.1.2 Existence of Solutions

In this section we prove three results to establish the existence of solutions from each point in $\mathcal{X}$. Specifically, we prove $\widehat{G}(D) \subset C \cup D = \mathcal{X}$ (Lemma 4.2), and $\widehat{F}(x) \cap T_C(x)$ for all $x \in C \setminus D$ (Lemma 4.3). The following assumption ensures that solutions to $\mathcal{H}$ cannot jump out of $\mathcal{X}$.

Assumption 4.2. For each $x = (z, \eta_0, \eta_1, v, q) \in \mathcal{X}$,

$$(g_z, \eta_0) \in C_{\mathrm{K}_0} \cup D_{\mathrm{K}_0} \qquad \forall g_z \in G_{\mathrm{P}}(z, \kappa(x)), \tag{4.16}$$

$$(z, g_{\eta_0}) \in C_{\mathrm{K}_0} \cup D_{\mathrm{K}_0} \qquad \forall g_{\eta_0} \in G_{\mathrm{K}_0}(z, \eta_0), \tag{4.17}$$

$$\big(z, \eta_0, g_{\eta_1}, v, q\big) \in C_{\mathrm{K}_1} \cup D_{\mathrm{K}_1} \qquad \forall g_{\eta_1} \in G_{\mathrm{K}_1}(x), \tag{4.18}$$

$$\big(z, \eta_0, \eta_1, g_{\mathsf{v}}(x), 1 - q\big) \in C_{\mathrm{S}} \cup D_{\mathrm{S}}. \tag{4.19}$$

Lemma 4.2. *Suppose $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, and $\mathcal{H}_{\mathrm{S}}$ satisfy Assumptions 4.1 and 4.2. Then,*

$$C \cup D = \mathcal{X} \quad \text{and} \quad \widehat{G}(D) \subset \mathcal{X}.$$

The proof of Lemma 4.2 is in Section B.1.3.

To ensure that flows are viable for $\mathcal{H}$ at all points in $C \setminus D$, we impose the following conditions on the subsystems.

Assumption 4.3. The subsystems $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, and $\mathcal{H}_{\mathrm{S}}$ satisfy the following.

(V1) (*Ensure viability of (z, η_0) when $q = 0$*): For all $(z, \eta_0) \in C_{\mathrm{K}_0} \setminus D_{\mathrm{K}_0}$,

$$F_{\mathrm{P}\times 0}(z, \eta_0) \cap T_{C_{\mathrm{K}_0}}(z, \eta_0) \neq \varnothing;$$

(V2) (*Ensure viability of (z, η_0) when $q = 1$*): For all $(z,\eta_0,\eta_1,v,1) \in C_{\mathrm{K}_1} \setminus \big(\mathcal{S}_{1\mapsto 0} \times \{1\}\big)$,

$$(z, \eta_0) \in \operatorname{int}(C_{\mathrm{K}_0});$$

(V3) (*Ensure viability of η_1*): For all $x \in C_{\mathrm{K}_1} \setminus D_{\mathrm{K}_1}$,

$$F_{\mathrm{K}_1}(x) \cap T_{C_{\mathrm{K}_1}}(\eta_1) \neq \varnothing;$$

(V4) (*Ensure viability of v*): For all $x \in C_{\mathrm{S}} \setminus D_{\mathrm{S}}$,

$$f_{\mathsf{v}}(x) \in T_{C_{\mathrm{S}}}(x);$$

(V5) (*Ensure $T_C(x)$ can be split*): For all $x = (z, \eta_0, \eta_1, v, q) \in C \setminus D$,

$$T_C(x) = T_{C_{\mathrm{K}_0}}(z, \eta_0) \times T_{C_{\mathrm{K}_1}}(\eta_1) \times T_{C_{\mathrm{S}}}(v) \times \{0\}. \tag{4.20}$$

Assumption (V5) allows us to separate the tangent cones of C_{K_0}, C_{K_1}, and C_{S} allowing us to handle them individually. One might be surprised that $T_C(x)$ could be written without reference to C_{P}. The reason is that (4.20) is only assumed to hold for $x \in C \setminus D$. If x is in the boundary of $C_{\mathrm{P}}^{\mathrm{CL}}$, then it must be in the boundary of C (not in $\partial D_{\mathrm{P}}^{\mathrm{CL}}$) and $T_{C_{\mathrm{P}}^{\mathrm{CL}}}(x) = T_C(x)$.

Lemma 4.3 (Viability condition)**.** *Suppose $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, and $\mathcal{H}_{\mathrm{S}}$ satisfy Assumption 4.1 and Assumption 4.3. Then, for each $x \in C \setminus D$, there exists an open neighborhood U of x such that*

$$F(x') \cap T_C(x') \neq \varnothing \qquad \forall x' \in U \cap C.$$

Proof. Take any $x := (z, \eta_0, \eta_1, v, q) \in C \setminus D$. The set D is closed (Lemma 4.1), so there exists an open neighborhood U of x that is disjoint from D; that is, $U \cap D = \varnothing$. Take any $x' \in U \cap C$. By (V5),

$$T_C(x) = T_{C_{\mathrm{K}_0}}(z, \eta_0) \times T_{C_{\mathrm{K}_1}}(\eta_1) \times T_{C_{\mathrm{S}}}(v) \times \{0\}.$$

Substituting the definition of F and distributing the intersection operation produces

$$\begin{aligned}\big(F(x) \cap T_C(x)\big) = &\left(\begin{bmatrix} F_{\mathrm{P}}(z, \kappa(x)) \\ F_{\mathrm{K}_0}(z, \eta_0) \end{bmatrix} \cap T_{C_{\mathrm{K}_0}}(z, \eta_0)\right) \\ &\times \big(F_{\mathrm{K}_1}(x) \cap T_{C_{\mathrm{K}_1}}(\eta_1)\big) \\ &\times \big(\{f_{\mathsf{v}}(x)\} \cap T_{C_{\mathrm{S}}}(v)\big) \\ &\times \{0\}.\end{aligned} \tag{4.21}$$

By (V3) and (V4), $F_{\mathrm{K}_1}(x) \cap T_{C_{\mathrm{K}_1}}(\eta_1)$ and $\{f_{\mathsf{v}}(x)\} \cap T_{C_{\mathrm{S}}}(v)$ are nonempty. Thus, all that remains is to show that

$$\begin{bmatrix} F_{\mathrm{P}}(z, \kappa(x)) \\ F_{\mathrm{K}_0}(z, \eta_0) \end{bmatrix} \cap T_{C_{\mathrm{K}_0}}(z, \eta_0)$$

is nonempty. We consider $q = 0$ and $q = 1$ by cases.

Case 1. Suppose $q = 0$. Then

$$\begin{bmatrix} F_{\mathrm{P}}(z, \kappa(x)) \\ F_{\mathrm{K}_0}(z, \eta_0) \end{bmatrix} = \begin{bmatrix} F_{\mathrm{P}}(z, \kappa_0(z, \eta_0)) \\ F_{\mathrm{K}_0}(z, \eta_0) \end{bmatrix} = F_{\mathrm{P}\times 0}(z, \eta_0).$$

By (V1),

$$F_{\mathrm{P}\times 0}(z, \eta_0) \cap T_{C_{\mathrm{K}_0}}(z, \eta_0) \neq \varnothing,$$

completing this case.

Case 2. Suppose $q = 1$. We picked x to not be in D, so $x \notin \mathcal{S}_{1\mapsto 0} \times \{1\}$ (since $\mathcal{S}_{1\mapsto 0} \subset D_{\mathrm{S}} \subset D$). Thus, $(z, \eta_0) \in \mathrm{int}(C_{\mathrm{K}_0})$ by (V2). But the tangent cone at an interior point of a set is the full tangent space, namely

$$T_{C_{\mathrm{K}_0}}(z, \eta_0) = \mathbb{R}^{n_{\mathrm{P}}} \times \mathbb{R}^{n_0}.$$

Therefore,

$$\begin{bmatrix} F_{\mathrm{P}}(z, \kappa(x)) \\ F_{\mathrm{K}_0}(z, \eta_0) \end{bmatrix} \cap T_{C_{\mathrm{K}_0}}(z, \eta_0) = \begin{bmatrix} F_{\mathrm{P}}(z, \kappa_1(x)) \\ F_{\mathrm{K}_0}(z, \eta_0) \end{bmatrix} \cap \big(\mathbb{R}^{n_{\mathrm{P}}} \times \mathbb{R}^{n_0}\big) = \begin{bmatrix} F_{\mathrm{P}}(z, \kappa_1(x)) \\ F_{\mathrm{K}_0}(z, \eta_0) \end{bmatrix}.$$

The set $F_{\mathrm{P}}(z, \kappa_1(x)) = F_{\mathrm{P}}^{\mathrm{CL}}(x)$ is nonempty because $x \in C_{\mathrm{P}}^{\mathrm{CL}} = \mathrm{dom}(F_{\mathrm{P}}^{\mathrm{CL}})$. Similarly, the set $F_{\mathrm{K}_0}(z, \eta_0)$ is nonempty because $(z, \eta_0) \in C_{\mathrm{K}_0} = \mathrm{dom}(F_{\mathrm{K}_0})$ per (B3). □

By combining Lemmas 4.1–4.3—along with an assumption that solutions do not have finite escape time during flows—we are able to establish the existence of solutions and prove that all maximal solutions are complete. Recall that say that $z : [t_0, T) \to \mathbb{R}^n$ with $t_0 < T$ has a *finite escape time* T if $\lim_{t \nearrow T} |z(t)| = \infty$.

Lemma 4.4 (Maximal Solutions are Complete)**.** *Suppose $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, and $\mathcal{H}_{\mathrm{S}}$ satisfy Assumptions 4.1–4.3, and that no solution to*

$$\dot{x} \in \widehat{F}(x) \quad x \in C$$

has a finite escape time, where $\widehat{F}$ is given in (4.15)*. Then, for each $x_0 \in C \cup D$, there exists a non-trivial solution to $\widehat{\mathcal{H}}$ in* (4.15) *starting at x_0 and every maximal solution to $\widehat{\mathcal{H}}$ is complete.*

Proof. Our proof uses [3, Prop. 2.34]. By Assumption 4.3, we have that $\mathcal{H}$ satisfies the viability condition (VC) of [3, Prop. 2.34] for each $x \in C \setminus D$. Since $\widehat{F}(x) \supset F(x)$, we immediately have that $\widehat{\mathcal{H}}$ also satisfies (VC) for each $x \in C \setminus D$.

Because $\widehat{\mathcal{H}}$ satisfies the hybrid basic conditions (Lemma 4.1) and (VC) holds at each point in $C \setminus D$, every maximal solution φ to $\mathcal{H}$ satisfies exactly one of the following cases [3, Prop. 2.34]:

(M1) φ escapes to infinity in finite time.

(M2) φ leaves $C \cup D$.

(M3) φ is complete.

By assumption, (M1) does not occur. Furthermore, it was shown in Lemma 4.2 that $\widehat{G}(D) \subset C \cup D$, so (M2) does not occur. By elimination, only (M3) is possible, therefore every maximal solution is complete. □

4.1.3 Ensuring Minimum Dwell Times

In Lemma 4.4, we showed that maximal solutions are complete, but this means that $t + j \to \infty$ in the domain of solutions—there is no guarantee, yet, that the solution exists for all ordinary time, since j may approach infinity on its own. For

our chosen hybrid system model to be practical, however, we must ensure that there is a positive minimum dwell time between jumps, ensuring that the solution does not chatter and exists for all ordinary time $t \geq 0$. The following assumption introduces conditions that prevent the subsystems from jumping too frequently. Before we state the assumption, however, we must introduce additional notation. To allow us to map back from vectors or sets in $\mathcal{X}$ to the subsystem states, we define a projection map $\pi_\star$ for each $\star \in \{\text{“}z\text{”}, \text{“}\eta_0\text{”}, \text{“}\eta_1\text{”}, \text{“}v\text{”}, \text{“}q\text{”}\}$ as

$$(z, \eta_0, \eta_1, v, q) \mapsto \pi_\star(z, \eta_0, \eta_1, v, q) := \star.$$

For a set $\mathcal{B} \subset \mathcal{X}$, the image of $\mathcal{B}$ under $\pi_\star$ is $\pi_\star(\mathcal{B}) := \{\star \mid (z, \eta_0, \eta_1, v, q) \in \mathcal{B}\}$.

Assumption 4.4. There exists $d_{\min} > 0$ such that

$$\operatorname{dist}(\pi_z(D_{\mathrm{P}}),\, G_{\mathrm{P}}(D_{\mathrm{P}})) \geq d_{\min}, \tag{4.22}$$

$$\operatorname{dist}(\pi_{\eta_0}(D_{\mathrm{K}_0}),\, G_{\mathrm{K}_0}(D_{\mathrm{K}_0})) \geq d_{\min}, \tag{4.23}$$

$$\operatorname{dist}(\pi_{\eta_1}(D_{\mathrm{K}_1}),\, G_{\mathrm{K}_1}(D_{\mathrm{K}_1})) \geq d_{\min}. \tag{4.24}$$

In the following result, given a solution φ to $\mathcal{H}$ and $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$, we say that $(t, j) \in \operatorname{dom}(\varphi)$ is[1] a *$\mathcal{H}_\star$-jump time* if

$$\varphi(t, j) \in D_\star^{\mathrm{CL}} \quad \text{and} \quad \varphi(t, j+1) \in G_\star^{\mathrm{CL}}(\varphi(t, j)).$$

Lemma 4.5. *Suppose $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, and $\mathcal{H}_{\mathrm{S}}$ satisfy Assumptions 4.1–4.4 and*

$$D_{\mathrm{S}} \cap G_{\mathrm{S}}^{\mathrm{CL}}(D_{\mathrm{S}}) = \varnothing.$$

Then, for each compact set $K \subset \mathcal{X}$, there exists $\Delta_T > 0$ such that for every solution φ to $\widehat{\mathcal{H}}$ in (4.15) with $\operatorname{range}(\varphi) \subset K$, the following hold:

1. *For each $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$ and every pair of distinct $\mathcal{H}_\star$-jump times (t_1, j_1) and (t_2, j_2) in $\operatorname{dom}(\varphi)$,*

$$|t_2 - t_1| \geq \Delta_T.$$

(Informally: Δ_T is a minimum dwell time between jumps for each subsystem of $\widehat{\mathcal{H}}$.)

[1] A minor technical note: We have defined our closed-loop system so that one subsystem at a time jumps. Nevertheless, it is possible for a time $(t, j) \in \operatorname{dom}(\varphi)$ to be a $\mathcal{H}_\star$-jump for multiple $\star$'s. How? Consider $x \mapsto G_{\mathrm{P}}^{\mathrm{CL}}(x) = \{x\}$, $x \mapsto G_{\mathrm{K}_0}^{\mathrm{CL}}(x) = \{x\}$, and a $\mathcal{H}_{\mathrm{P}}$-jump from $x_0 \in D_{\mathrm{P}}^{\mathrm{CL}} \cap D_{\mathrm{K}_0}^{\mathrm{CL}}$.

2. For all (t_1, j_1) and (t_2, j_2) in $\mathrm{dom}(\varphi)$,

$$|t_2 - t_1| \leq \Delta_T \implies |j_2 - j_1| \leq \Delta_J := 4.$$

Proof. Let $\bar{f} := \sup\Big\{|f| : f \in \widehat{F}(K)\Big\}$. We have that $\bar{f}$ is finite because $\widehat{F}$ is locally bounded (Lemma 4.1) and K is compact.

Take $d_{\min} > 0$ from Assumption 4.4. The set $D_{\mathrm{S}} \cap K$ is compact and $\widehat{G^{\mathrm{CL}}_{\mathrm{S}}}$ is continuous (since $\widehat{G^{\mathrm{CL}}_{\mathrm{S}}} = G^{\mathrm{CL}}_{\mathrm{S}}$, by definition), so $\widehat{G^{\mathrm{CL}}_{\mathrm{S}}}(D_{\mathrm{S}} \cap K)$ is compact. Furthermore, $\widehat{G^{\mathrm{CL}}_{\mathrm{S}}}(D_{\mathrm{S}} \cap K)$ and $D_{\mathrm{S}} \cap K$ are disjoint, by assumption, so

$$d_s := \mathrm{dist}(D_{\mathrm{S}} \cap K,\, \widehat{G^{\mathrm{CL}}_{\mathrm{S}}}(D_{\mathrm{S}} \cap K))$$

is positive. Let $d := \min\{d_s, d_{\min}\}$ and $\Delta_T := d/\bar{f}$, which are also both positive.

We want to show that distance between $D^{\mathrm{CL}}_{\mathrm{P}}$ and $\widehat{G^{\mathrm{CL}}_{\mathrm{P}}}(D^{\mathrm{CL}}_{\mathrm{P}})$ is at least $d_{\min}$. Take any

$$x := (z, \eta_0, \eta_1, v, q) \in D^{\mathrm{CL}}_{\mathrm{P}} \quad \text{and} \quad g := (g_z, \eta_0, \eta_1, v, q) \in \widehat{G^{\mathrm{CL}}_{\mathrm{P}}}(D^{\mathrm{CL}}_{\mathrm{P}}).$$

We have $z \in \pi_z(\overline{D_{\mathrm{P}}})$ and $g_z \in \overline{G_{\mathrm{P}}(D_{\mathrm{P}})}$, so

$$|x - g| = |z - g_z| \geq \mathrm{dist}(\overline{D_{\mathrm{P}}},\, \overline{G_{\mathrm{P}}(D_{\mathrm{P}})}) = \mathrm{dist}(D_{\mathrm{P}},\, G_{\mathrm{P}}(D_{\mathrm{P}})) \geq d_{\min}.$$

Since this holds for all $x \in D^{\mathrm{CL}}_{\mathrm{P}}$ and $g \in \widehat{G^{\mathrm{CL}}_{\mathrm{P}}}(D^{\mathrm{CL}}_{\mathrm{P}})$, we have that

$$\mathrm{dist}(D^{\mathrm{CL}}_{\mathrm{P}},\, \widehat{G^{\mathrm{CL}}_{\mathrm{P}}}(D^{\mathrm{CL}}_{\mathrm{P}})) \geq d_{\min}.$$

Similarly, we can find $\mathrm{dist}(D^{\mathrm{CL}}_{\mathrm{K}q},\, \widehat{G^{\mathrm{CL}}_{\mathrm{K}q}q}(D^{\mathrm{CL}}_{\mathrm{K}q})) \geq d_{\min}$ for $q \in \{0, 1\}$. Thus, for each $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$,

$$\mathrm{dist}(D^{\mathrm{CL}}_{\star} \cap K,\, \widehat{G^{\mathrm{CL}}_{\star}}(D^{\mathrm{CL}}_{\star} \cap K)) \geq d.$$

Take any solution φ to $\mathcal{H}$ with $\mathrm{range}(\varphi) \subset K$ and any $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$. For each jump time $(t, j) \in \mathrm{dom}(\varphi)$, the subsystem state of $\mathcal{H}_\star$ jumps (is non-constant) only if (t, j) is an $\mathcal{H}_\star$-jump time. Thus, between any distinct pair of $\mathcal{H}_\star$-jump times (t_1, j_1) and (t_2, j_2), the solution must flow a distance of at least d at a velocity no more than $\bar{f}$, therefore,

$$|t_2 - t_1| \geq \Delta_T.$$

Thus concluding the proof of Item 1.

Item 2 follows directly. Per Item 1, each subsystem can jump at most once within any interval of length Δ_T, so solutions to $\widehat{\mathcal{H}}$ can jump at most $\Delta_J := 4$ times (once per subsystem) in the same interval. □

4.2 Supervisor Design for Global Asymptotic Stability

We now return to the problem of rendering a nonempty compact set $\mathcal{A}_{\mathrm{P}}$ to be UGAS. Our goal is to design a hybrid supervisor $\mathcal{H}_{\mathrm{S}}$ that switches between $\mathcal{H}_{\mathrm{K}_0}$ and $\mathcal{H}_{\mathrm{K}_1}$ while ensuring that $\mathcal{A}_{\mathrm{P}}$ remains UGAS despite sometimes exploiting $\mathcal{H}_{\mathrm{K}_1}$. To do so, we will extend the design of the supervisor developed in Chapter 3 to the more general setting of this chapter. We assume the existence of a Lyapunov function V_{P} that certifies, via Corollary 1.1, that $\mathcal{A}_{\mathrm{P}}$ is UGpAS for $\mathcal{H}_{\mathrm{P}\times 0}$ (Assumption 4.5). Additionally, sufficient conditions are assumed to ensure that all maximal solutions to $\mathcal{H}_{\mathrm{P}\times 0}$ are complete and switching is not too frequent.

In contrast to Section 3.1 where $\mathcal{A}_{\mathrm{P}}$ was taken as a subset of the plant's state space, we now want to account for the state η_0 of $\mathcal{H}_{\mathrm{K}_0}$ since it may be necessary to ensure that the joint state (z, η_0) of $\mathcal{H}_{\mathrm{P}}$ and $\mathcal{H}_{\mathrm{K}_0}$ converge to a given set. Thus, we let $\mathcal{A}_{\mathrm{P}}$ be a subset of $C_{\mathrm{K}_0} \cup D_{\mathrm{K}_0} \subset \mathbb{R}^{n_{\mathrm{P}}} \times \mathbb{R}$ instead of just $\mathbb{R}^{n_{\mathrm{P}}}$. The following assumption gives Lyapunov conditions asserting that $\mathcal{H}_{\mathrm{K}_0}$ is a Lyapunov-certified controller for the plant $\mathcal{H}_{\mathrm{P}}$.

Assumption 4.5. The set $\mathcal{A}_{\mathrm{P}} \subset \mathcal{E}_{\mathrm{P},0}$ and the closed-loop system $\mathcal{H}_{\mathrm{P}\times 0}$ in (4.14) satisfy the following:

(L1) $\mathcal{A}_{\mathrm{P}}$ is compact and nonempty;

(L2) There exists a continuous function $V_{\mathrm{P}} : \mathbb{R}^{n_{\mathrm{P}}} \times \mathbb{R}^{n_0} \to \mathbb{R}_{\geq 0}$ such that

- V_{P} is $\mathcal{C}^1$ on an open neighborhood of C_{K_0}
- V_{P} is positive definite on $\mathcal{E}_{\mathrm{P},0}$ with respect to $\mathcal{A}_{\mathrm{P}}$;

(L3) there exists $\alpha \in \mathcal{K}_\infty$ such that $\alpha(|(z, \eta_0)|_{\mathcal{A}_{\mathrm{P}}}) \leq V_{\mathrm{P}}(z, \eta_0)$ for all $(z, \eta_0) \in \mathcal{E}_{\mathrm{P},0}$;

(L4) there exists a continuous function $\sigma_0 : \mathcal{E}_{\mathrm{P},0} \to \mathbb{R}_{\geq 0}$ that is positive definite on $\mathcal{E}_{\mathrm{P},0}$ with respect to $\mathcal{A}_{\mathrm{P}}$, and

$$\sup \mathcal{L}_{F_{\mathrm{P}\times 0}} V_{\mathrm{P}}(z, \eta_0) \leq -\sigma_0(z, \eta_0) \qquad \forall (z, \eta_0) \in C_{\mathrm{P}\times 0};$$

(L5) For all $(z, \eta_0) \in \mathcal{E}_{\mathrm{P},0}$ and $(z, u) \in D_{\mathrm{P}}$,

$$V_{\mathrm{P}}(g_z, \eta_0) \leq V_{\mathrm{P}}(z, \eta_0) \qquad \forall g_z \in G_{\mathrm{P}}(z, u).$$

For all $(z, \eta_0) \in D_{\mathrm{K}_0}$,

$$V_{\mathrm{P}}(z, g_{\eta_0}) \leq V_{\mathrm{P}}(z, \eta_0) \qquad \forall g_{\eta_0} \in G_{\mathrm{K}_0}(z, \eta_0). \qquad \diamond$$

Remark 4.1. Condition (L5) imposes that V_{P} is nonincreasing at jumps for any input value u (regardless of whether u is produced by κ_0), and V_{P} is also nonincreasing for jumps in the value η_0 according to the dynamics of $\mathcal{H}_{\mathrm{K}_0}$.

The supervisor's state variable v is assumed to be any nonnegative real number, so the state space is $\mathcal{V} := \mathbb{R}_{\geq 0}$. When $q = 0$ and there is a sufficient buffer between $V_{\mathrm{P}}(z, \eta_0)$ and v, then the supervisor will trigger a switch from $q = 0$ to $q = 1$. When $q = 1$, if $V_{\mathrm{P}}(z, \eta_0) \geq v$ ever holds, then the supervisor switches to $q = 0$. In particular, a continuous positive function $(z, \eta_0) \mapsto \delta(z, \eta_0) > 0$ defines a state-dependent buffer required between $V_{\mathrm{P}}(z, \eta_0)$ and an auxiliary variable v that is required before a switch to $q = 1$ is permitted. The supervisor switching and hold sets must satisfy the following inclusions:

$$\mathcal{S}_0 \supset \{(z, \eta_0, \eta_1, v) \mid V_{\mathrm{P}}(z, \eta_0) + \delta(z, \eta_0) \geq v\}, \tag{4.25a}$$

$$\mathcal{S}_{0 \mapsto 1} \subset \{(z, \eta_0, \eta_1, v) \mid V_{\mathrm{P}}(z, \eta_0) + \delta(z, \eta_0) \leq v\}, \tag{4.25b}$$

$$\mathcal{S}_1 \subset \{(z, \eta_0, \eta_1, v) \mid V_{\mathrm{P}}(z, \eta_0) \leq v\}, \tag{4.25c}$$

$$\mathcal{S}_{1 \mapsto 0} \supset \{(z, \eta_0, \eta_1, v) \mid V_{\mathrm{P}}(z, \eta_0) \geq v\}. \tag{4.25d}$$

You may sometimes simply choose the switching and hold sets using equality in (4.25), but having flexibility may be of use in the following ways:

1. Picking a restricted switching set $\mathcal{S}_{0 \mapsto 1}$ (and larger $\mathcal{S}_0$) may be desirable to avoid switching to $\mathcal{H}_{\mathrm{K}_1}$ at times when $\mathcal{H}_{\mathrm{K}_1}$ is not expected to perform well. Alternatively, one may wish to ensure that there is a minimum dwell time when $q = 0$ before switching to the uncertified controller by introducing an auxiliary timer variable $\tau \geq 0$ to the system with $\dot{\tau} = 1$, we can impose a minimum dwell time $T > 0$ by picking

$$\mathcal{S}_{0 \mapsto 1} := \{(z, \eta_0, \eta_1, v, \tau) \mid V_{\mathrm{P}}(z, \eta_0) + \delta(z, \eta_0) \leq v,\ \tau \geq T\}.$$

2. Picking an expanded switching set $\mathcal{S}_{1 \mapsto 0}$ (and smaller $\mathcal{S}_1$) can be necessary to force the system back to the certified controller. For example, if $\mathcal{H}_{\mathrm{K}_1}$ would

cause (z, η_0) to move out of $\mathcal{E}_{\mathrm{P},0}$, we can ensure viability by choosing $\mathcal{S}_{1\mapsto 0}$ to cover the boundary. Specifically, $\mathcal{S}_{1\mapsto 0}$ should be chosen to satisfy (V2) in Assumption 4.3.

To define the dynamics of v, we define f_{v} and g_{v} for all $(z, \eta_0, v) \in \mathcal{E}_{\mathrm{P},0} \times \mathbb{R}_{\geq 0}$ as

$$f_{\mathsf{v}}(z, \eta_0, v) := -\gamma \tanh(v)\sigma_0(z, \eta_0) - \mu\big(v - V_{\mathrm{P}}(z, \eta_0)\big) \tag{4.26}$$

$$g_{\mathsf{v}}(z, \eta_0, v) := \max\{V_{\mathrm{P}}(z, \eta_0),\ v\}. \tag{4.27}$$

The term $-\mu(v - V_{\mathrm{P}}(z, \eta_0))$ pulls v toward $V_{\mathrm{P}}(z, \eta_0)$, which helps v to "catch up" if $V_{\mathrm{P}}(z, \eta_0)$ has dropped quickly, and allows v to grow if v is initialized less than $V_{\mathrm{P}}(z, \eta_0)$. The term $-r \tanh(v)\sigma_0(z, \eta_0)$ pushes v toward zero to prevent v from stopping if $V_{\mathrm{P}}(z, \eta_0) = v \neq 0$. Since $-\sigma_0(z, \eta_0) < 0$ for $(z, \eta_0) \notin \mathcal{A}_{\mathrm{P}}$, we multiply $\sigma(z, \eta_0)$ by $\tanh(v)$ so that $f_{\mathsf{v}}(z, \eta_0, v)$ is nonnegative when $V_{\mathrm{P}}(z, \eta_0) = v \neq 0$. This ensures $\dot{v} \geq 0$ when $v = 0$ is already on the boundary of $\mathbb{R}_{\geq 0}$. One important property of f_{v} is that it is nonnegative for $v = 0$, ensuring that $f_{\mathsf{v}}(z, \eta_0, 0) \in T_{\mathcal{V}}(0)$:

$$f_{\mathsf{v}}(z, \eta_0, 0) \geq 0 \qquad \forall (z, \eta_0) \in \mathcal{E}_{\mathrm{P},0}. \tag{4.28}$$

For the closed-loop system, we want to ensure that $(z, \eta_0) \to \mathcal{A}_{\mathrm{P}}$ and $v \to 0$, which is to say that x converges to the set

$$\mathcal{A} := \mathcal{A}_{\mathrm{P}} \times \mathcal{E}_1 \times \{0\} \times \{0, 1\} \subset \mathcal{X}.$$

To this end, we define a Lyapunov function candidate V of $\mathcal{A}$. Let $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$ be defined for all $x = (z, \eta_0, \eta_1, v, q) \in \mathcal{X}$ by

$$V(x) := \max\{V_{\mathrm{P}}(z, \eta_0),\ v\}. \tag{4.29}$$

The following results establish properties on V that are required by Corollary 1.1 so that we can apply that result to prove that $\mathcal{A}$ is UGpAS. In Lemma B.1, we assert that V is a Lyapunov function candidate with respect to $\mathcal{A}$ for $\mathcal{H}$. We then construct a function σ in (4.30) that is LSC and positive definite on C with respect to $\mathcal{A}$ (Lemma 4.6) such that $-\sigma(x)$ is an upper bound on the rate of change of V along flows in $\mathcal{H}$ (Lemma 4.7). In Lemma 4.8, we show that V is nondecreasing across jumps for solutions to $\mathcal{H}$. Altogether, these results give us that V is nondecreasing at jumps and strictly decreasing during flows outside of $\mathcal{A}$. To show use these results

to prove $\mathcal{A}$ is UGpAS, we must show that jumps do not occur too often, so the strict decrease during flows has sufficient time to produce convergence. Establishing a bound on the frequency of jumps is left until Section 4.1.3.

Assumption 4.6 (UGAS Supervisor)**.** The supervisor $\mathcal{H}_{\mathrm{S}}$ satisfies $\gamma \in (0, 1]$, $\mu > 0$, and δ is continuous and strictly positive. $\diamond$

Let $x \mapsto \sigma(x)$ be defined for all $x \in \mathcal{X}$ by

$$\sigma(x) := \begin{cases} \sigma_0(z, \eta_0) & \text{if } V_{\mathrm{P}}(z, \eta_0) > v \\ -f_{\mathsf{v}}(z, \eta_0, v) & \text{if } V_{\mathrm{P}}(z, \eta_0) \leq v. \end{cases} \tag{4.30}$$

We will show that σ is LSC, positive definite with respect to $\mathcal{A}$, and

$$\sup \mathcal{L}_F(x) \leq -\sigma(x).$$

Lemma 4.6. *Suppose that $\mathcal{A}_{\mathrm{P}}$ and $\mathcal{H}_{\mathrm{P}\times 0}$ satisfy Assumptions 4.5 and 4.6. Then, σ is LSC and positive definite with respect to $\mathcal{A}$ on C.*

Proof. Since σ_0 and f_{v} are continuous, the only points where σ can discontinuities is where $V_{\mathrm{P}}(z, \eta_0) = v$. Take any $x := (z, \eta_0, \eta_1, v, q) \in \mathcal{X}$ such that $V_{\mathrm{P}}(z, \eta_0) = v$. We have that

$$\sigma(x) = -f_{\mathsf{v}}(z, \eta_0, v) = \gamma \tanh(v)\sigma_0(z, \eta_0) \leq \sigma_0(z, \eta_0),$$

where the inequality holds because $0 \leq \gamma \tanh(v) < 1$ for each $v \geq 0$ because $\gamma \in (0, 1]$ and $\tanh(v) \in [0, 1)$. Thus, $\liminf_{x' \to x} \sigma(x') = \sigma(x)$, so σ is LSC at x and since this holds for all x, we have that σ is LSC everywhere.

Next we prove the positive definiteness of σ. Take any $x := (z, \eta_0, \eta_1, v, q) \in C$. If $x \in \mathcal{A}$, then $V_{\mathrm{P}}(z, \eta_0) = 0$ and $v = 0$, so, by construction of f_{v} in (4.26),

$$\sigma(x) = -f_{\mathsf{v}}(z, \eta_0, v) = 0.$$

Thus, $\sigma(x) = 0$.

Alternatively, suppose $x \notin \mathcal{A}$ and consider the following cases.

Case 1. Suppose $V_{\mathrm{P}}(z, \eta_0) > v$. Since V_{P} is assumed to be positive definite with respect to $\mathcal{A}_{\mathrm{P}}$, we have $(z, \eta_0) \notin \mathcal{A}_{\mathrm{P}}$. Thus, $\sigma(x) = \sigma_0(z, \eta_0) > 0$, since σ_0 is positive definite with respect to $\mathcal{A}_{\mathrm{P}}$.

Case 2. Suppose $V_{\mathrm{P}}(z,\eta_0) < v$. Then,

$$\sigma(x) = -f_{\mathsf{v}}(z,\eta_0,v) = \gamma\tanh(v)\sigma_0(z,\eta_0) + \mu\big(v - V_{\mathrm{P}}(z,\eta_0)\big).$$

The first term, $\gamma\tanh(v)\sigma_0(z,\eta_0)$ is nonnegative, whereas $\mu\big(v - V_{\mathrm{P}}\big) > 0$. Thus, $\sigma(x) > 0$.

Case 3. Suppose $V_{\mathrm{P}}(z,\eta_0) = v$. Since $x \notin \mathcal{A}$, we must have $V_{\mathrm{P}}(z,\eta_0) = v > 0$, so $(z,\eta_0) \notin \mathcal{A}_{\mathrm{P}}$ and $\sigma_0(z,\eta_0) > 0$, by positive definiteness. Thus, $\gamma\tanh(v)\sigma_0(z,\eta_0) > 0$ and $\mu\big(v - V_{\mathrm{P}}(z,\eta_0)\big) = 0$.
Therefore, σ is positive definite with respect to $\mathcal{A}$ on C. □

The next result establishes that for each solution φ to $\widehat{\mathcal{H}}$, the function $(t,j) \mapsto V_{\mathrm{P}}(\varphi(t,j))$ decreases along each interval of flow in $\mathrm{dom}(\varphi)$ while φ is outside $\mathcal{A}$.

Lemma 4.7. *Suppose that $\mathcal{A}_{\mathrm{P}}$ and $\mathcal{H}_{\mathrm{P}\times 0}$ satisfy Assumptions 4.5 and 4.6. Then, the Lyapunov function candidate V in* (4.29) *and the regularized flow map $\widehat{F}$ defined in Section 4.1.1 satisfy*

$$\sup \mathcal{L}_{\widehat{F}\cap T_C} V(x) \le -\sigma(x) \qquad \forall x \in C. \tag{4.31}$$

Proof. Take any $x := (z,\eta_0,\eta_1,v,q) \in C$. If $\widehat{F}(x) \cap T_C(x)$ is empty, then

$$\mathcal{L}_{\widehat{F}\cap T_C} V(x) = -\infty,$$

so (4.31) holds. Suppose, instead, that $\widehat{F}(x) \cap T_C(x) \neq \varnothing$. We want to show that $\langle \zeta,\, f\rangle \le -\sigma(x)$ for each $\zeta \in \partial^\circ V(x)$ and each $f \in \widehat{F}(x) \cap T_C(x)$. Take any such ζ and f. We write the components of f as $f := (f_z, f_{\eta_0}, f_{\eta_1}, f_v, 0)$, where

$$f_z \in \widehat{F_{\mathrm{P}}^{\mathrm{CL}}}(x), \quad f_{\eta_0} \in \widehat{F_{\mathrm{K}0}^{\mathrm{CL}}}(x) = F_{\mathrm{K}_0}(z,\eta_0), \quad f_{\eta_1} \in \widehat{F_{\mathrm{K}1}^{\mathrm{CL}}}(x), \quad \text{and} \quad f_v := f_{\mathsf{v}}(z,\eta_0,v).$$

Evaluating the generalized gradient of V, we find

$$\partial^\circ V(x) = \begin{cases} \Big\{\begin{bmatrix} 0_{n_{\mathrm{P}}}^\top & 0_{n0}^\top & 0_{n_1}^\top & 1 & 0\end{bmatrix}^\top\Big\} & \text{if } V_{\mathrm{P}}(z,\eta_0) < v \\ \mathrm{conv}\Big\{\begin{bmatrix} 0_{n_{\mathrm{P}}}^\top & 0_{n0}^\top & 0_{n_1}^\top & 1 & 0\end{bmatrix}^\top, \begin{bmatrix} \nabla V_{\mathrm{P}}^\top(z,\eta_0) & 0_{n_1}^\top & 0 & 0\end{bmatrix}^\top\Big\} & \text{if } V_{\mathrm{P}}(z,\eta_0) = v \\ \Big\{\begin{bmatrix} \nabla V_{\mathrm{P}}^\top(z,\eta_0) & 0_{n_1}^\top & 0 & 0\end{bmatrix}^\top\Big\} & \text{if } V_{\mathrm{P}}(z,\eta_0) > v. \end{cases} \tag{4.32}$$

(The gradient ∇V_{P} exists because V_{P} is assumed to be $\mathcal{C}^1$.)

Consider the following three cases.

Case 1. Suppose $V_{\mathrm{P}}(z,\eta_0) < v$. Then, $\zeta = \begin{bmatrix} 0_{n_{\mathrm{P}}}^\top & 0_{n_0}^\top & 0_{n_1}^\top & 1 & 0 \end{bmatrix}^\top$ is the unique element of $\partial^\circ V(x)$, so

$$\langle \zeta,\, f\rangle = \langle 1,\, f_{\mathsf{v}}(z,\eta_0,v)\rangle = f_{\mathsf{v}}(z,\eta_0,v).$$

Since $V_{\mathrm{P}}(z,\eta_0) < v$, we have that $f_{\mathsf{v}}(z,\eta_0,v) = -\sigma(x)$, per the definition of σ in (4.30).

Case 2. Suppose $V_{\mathrm{P}}(z,\eta_0) > v$. Thus, $\zeta = \begin{bmatrix} \nabla V_{\mathrm{P}}^\top(z,\eta_0) & 0_{n_1}^\top & 0 & 0 \end{bmatrix}^\top$ is the unique element of $\partial^\circ V(x)$ and

$$\langle \zeta,\, f\rangle = \langle \nabla V_{\mathrm{P}}(z,\eta_0),\, (f_z, f_{\eta_0})\rangle.$$

We have that $V_{\mathrm{P}}(z,\eta_0) > v$, so $x \notin C_1$ (recall $C_1 = \{x \mid V_{\mathrm{P}}(z,\eta_0) \le v\}$). Since $x \in C_{\mathrm{S}} \subset C$, we must have $x \in C_0$ and $q = 0$. Thus, $\widehat{F_{\mathrm{P}}^{\mathrm{CL}}}(x) = F_{\mathrm{P}}(z,\kappa_0(z,\eta_0))$, so $f_{\mathrm{P}\times 0} := (f_z, f_{\eta_0}) \in F_{\mathrm{P}\times 0}(z,\eta_0)$ and $(z,\eta_0) \in C_{\mathrm{P}\times 0}$. From the definition of $\mathcal{L}_{F_{\mathrm{P}\times 0}} V_{\mathrm{P}}$,

$$\big\langle \nabla V_{\mathrm{P}}(z,\eta_0),\, (f_z, f_{\eta_0})\big\rangle \in \mathcal{L}_{F_{\mathrm{P}\times 0}} V_{\mathrm{P}}(z,\eta_0).$$

By the continuous-time Lyapunov condition (L4) in Assumption 4.5,

$$\sup \mathcal{L}_{F_{\mathrm{P}\times 0}} V_{\mathrm{P}}(z,\eta_0) \le -\sigma_0(z,\eta_0).$$

Finally, since $\sigma(x) = \sigma_0(z,\eta_0)$ for $V_{\mathrm{P}}(z,\eta_0) > v$, we arrive at the desired inequality:

$$\langle f,\, \zeta\rangle \le \sup \mathcal{L}_{F_{\mathrm{P}\times 0}} V_{\mathrm{P}}(z,\eta_0) \le -\sigma_0(z,\eta_0) = -\sigma(x).$$

Case 3. Suppose $V_{\mathrm{P}}(z,\eta_0) = v$. This case is more difficult because $\partial^\circ V(x)$ is not a singleton. We have that

$$\zeta \in \partial^\circ V(x) = \operatorname{conv}\Big\{ \begin{bmatrix} 0_{n_{\mathrm{P}}}^\top & 0_{n_0}^\top & 0_{n_1}^\top & 1 & 0 \end{bmatrix}^\top,\ \begin{bmatrix} \nabla V_{\mathrm{P}}^\top(z,\eta_0) & 0_{n_1}^\top & 0 & 0 \end{bmatrix}^\top \Big\}. \tag{4.33}$$

Since $\langle \zeta,\, f\rangle$ is linear in ζ, the maximum value of $\langle \zeta,\, f\rangle$ is attained at one of the endpoints of the convex hull in (4.33):

$$\begin{aligned} \langle \zeta,\, f\rangle &\le \max\Big\{ \Big\langle \begin{bmatrix} 0_{n_{\mathrm{P}}}^\top & 0_{n_0}^\top & 0_{n_1}^\top & 1 & 0 \end{bmatrix}^\top, f \Big\rangle,\ \Big\langle \begin{bmatrix} \nabla V_{\mathrm{P}}^\top(z,\eta_0) & 0_{n_1}^\top & 0 & 0 \end{bmatrix}^\top, f \Big\rangle \Big\} \\ &= \max\Big\{ \big\langle 1,\, f_{\mathsf{v}}(z,\eta_0,v)\big\rangle,\ \big\langle \nabla V_{\mathrm{P}}(z,\eta_0),\, (f_z, f_{\eta_0})\big\rangle \Big\}. \end{aligned} \tag{4.34}$$

From the definition of σ, we see

$$\langle 1,\, f_{\mathsf{v}}(z,\eta_0,v)\rangle = f_{\mathsf{v}}(z,\eta_0,v) = -\sigma(x).$$

To show $\langle \nabla V_{\mathrm{P}}(z, \eta_0),\, (f_z, f_{\eta_0}) \rangle \leq -\sigma(x)$, we will consider $q = 0$ and $q = 1$ separately. Suppose $q = 0$. Then, $(f_z, f_{\eta_0}) \in F_{\mathrm{P}\times 0}(z, \eta_0)$, so by (L4),

$$\left\langle \nabla V_{\mathrm{P}}(z, \eta_0),\, (f_z, f_{\eta_0}) \right\rangle \leq -\sigma_0(z, \eta_0) \leq f_{\mathsf{v}}(z, \eta_0, v) = -\sigma(x),$$

since $-f_{\mathsf{v}}(z, \eta_0, v) = \gamma \tanh(v)\sigma_0(z, \eta_0) \leq \sigma_0(z, \eta_0)$.

Suppose $q = 1$. For this case, we must use the fact that $f \in T_C(x)$. Calculating $T_C(x)$ directly is difficult but unnecessary. Instead, we find the tangent cone to C_{S}, which contains $T_C(x)$ because $C(x) \subset C_{\mathrm{S}}$. Since $q = 1$, we have $x \in C_1$. Furthermore, since C_0 and C_1 are separate components of C_{S} (i.e., separated by a positive distance), we have that $T_{C_{\mathrm{S}}}(x) = T_{C_1}(x)$.

The set C_1 can be written as

$$C_1 = \{(z, \eta_0, \eta_1, v) \mid V_{\mathrm{P}}(z, \eta_0) - v \leq 0\} \times \{1\}.$$

Thus, for every $w := (w_z, w_{\eta_0}, w_{\eta_1}, w_v, 0) \in T_{C_1}(x)$,

$$\langle \nabla_x (V_{\mathrm{P}}(z, \eta_0) - v),\, w \rangle = \langle (\nabla V_{\mathrm{P}}(z, \eta_0), 0, -1, 0),\, w \rangle \leq 0. \tag{4.35}$$

Rewriting (4.35) as separate inner products, we get

$$\langle \nabla V_{\mathrm{P}}(z, \eta_0),\, (w_z, w_{\eta_0}) \rangle - \langle 1,\, w_v \rangle = \langle \nabla V_{\mathrm{P}}(z, \eta_0),\, (w_z, w_{\eta_0}) \rangle - w_v \leq 0. \tag{4.36}$$

Since $f \in T_{C_1}(x)$, we can pick $w_z := f_z$, $w_{\eta_0} := f_{\eta_0}$, and $w_v := f_{\mathsf{v}}(z, \eta_0, v)$ and substitute in (4.36),

$$\langle \nabla V_{\mathrm{P}}(z, \eta_0),\, (f_z, f_{\eta_0}) \rangle - f_{\mathsf{v}}(z, \eta_0, v) \leq 0.$$

Rearranging the prior equation, we find

$$\langle \nabla V_{\mathrm{P}}(z, \eta_0),\, (f_z, f_{\eta_0}) \rangle \leq f_{\mathsf{v}}(z, \eta_0, v) = -\sigma(x).$$

Thus, $\langle \zeta,\, f \rangle \leq -\sigma(x)$, completing the proof. □

Next, we prove an analogous result for jumps, namely that $V(x)$ is nonincreasing at jumps.

Lemma 4.8. *Suppose that $\mathcal{A}_{\mathrm{P}}$ and $\mathcal{H}_{\mathrm{P}\times 0}$ satisfy Assumptions 4.5 and 4.6. Then, the Lyapunov function candidate V in* (4.29) *and the regularized jump map $\widehat{G}$ defined in Section 4.1.1 satisfy*

$$V(g) \leq V(x) \qquad \forall x \in D,\ \forall g \in \widehat{G}(x)$$

Proof. By Lemma B.1, V is a Lyapunov function candidate with respect to $\mathcal{A}$ for $\mathcal{H}$. Take any $x = (z, \eta_0, \eta_1, v, q) \in D$ and $g := (g_z, g_{\eta_0}, g_{\eta_1}, g_{\mathsf{v}}, g_q) \in \widehat{G}(x)$. We want to show $V(g) \leq V(x)$. Since $g \in \widehat{G}(x)$, we must have that $x \in D_\star^{\mathrm{CL}}$ and $g \in \widehat{G_\star^{\mathrm{CL}}}(x)$ for some $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$. Thus, we consider, individually, each of the four cases given by $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$.

Case 1. Suppose $x \in D_{\mathrm{P}}^{\mathrm{CL}}$ and $g \in \widehat{G_{\mathrm{P}}^{\mathrm{CL}}}(x)$. From the definitions of $D_{\mathrm{P}}^{\mathrm{CL}}$ and $\widehat{G_{\mathrm{P}}^{\mathrm{CL}}}$, we have that $g_z \in G_{\mathrm{P}}(z, u)$ for some $(z, u) \in D_{\mathrm{P}}$. By (L5),

$$V_{\mathrm{P}}(g_z, \eta_0) \leq V_{\mathrm{P}}(z, \eta_0).$$

Since g_{η_0} and v are constant when z jumps (i.e., $g_{\eta_0} = \eta_0$ and $g_{\mathsf{v}} = v$), we have that

$$V(g) = \max\{V_{\mathrm{P}}(g_z, g_{\eta_0}),\ g_{\mathsf{v}}\} = \max\{V_{\mathrm{P}}(g_z, \eta_0),\ v\} \leq \max\{V_{\mathrm{P}}(z, \eta_0),\ v\} = V(x).$$

Case 2. Suppose $x \in D_{\mathrm{K}_0}^{\mathrm{CL}}$ and $g \in \widehat{G_{\mathrm{K}_0}^{\mathrm{CL}}}(x)$. From the definitions of $D_{\mathrm{K}_0}^{\mathrm{CL}}$ and $\widehat{G_{\mathrm{K}_0}^{\mathrm{CL}}}$, we have $(z, \eta_0) \in D_{\mathrm{K}_0}$, $g_z = z$, $g_{\eta_0} \in G_{\mathrm{K}_0}(z, \eta_0)$, $g_{\eta_1} = \eta_1$, and $g_{\mathsf{v}} = v$. By (L5), $V_{\mathrm{P}}(z, g_{\eta_0}) \leq V_{\mathrm{P}}(z, \eta_0)$. Thus,

$$V(g) = \min\{V_{\mathrm{P}}(g_z, g_{\eta_0}), g_{\mathsf{v}}\} = \min\{V_{\mathrm{P}}(z, g_{\eta_0}), v\} \leq \min\{V_{\mathrm{P}}(z, \eta_0), v\} = V(x).$$

Case 3. Suppose $x \in D_{\mathrm{K}_1}^{\mathrm{CL}}$ and $g \in \widehat{G_{\mathrm{K}_1}^{\mathrm{CL}}}(x)$. From the definitions, we have $x \in D_{\mathrm{K}_0}$, $g_z = z$, $g_{\eta_0} = \eta_0$, and $g_{\mathsf{v}} = v$. Thus, $V(g) = V(x)$, since V only depends on z, η_0, and v, which are not changed.

Case 4. Suppose $x \in D_{\mathrm{S}}$ and $g \in \widehat{G_{\mathrm{S}}^{\mathrm{CL}}}(x)$. Then,

$$g_z = z, \quad g_{\eta_0} = \eta_0, \quad \text{and} \quad g_{\mathsf{v}} = \max\{V_{\mathrm{P}}(z, \eta_0), v\}.$$

By substitution, we find

$$\begin{aligned} V(g) &= \max\{V_{\mathrm{P}}(g_z, g_{\eta_0}),\ g_{\mathsf{v}}\} \\ &= \max\big\{V_{\mathrm{P}}(z, \eta_0),\ \max\{V_{\mathrm{P}}(z, \eta_0),\ v\}\big\} \\ &= \max\{V_{\mathrm{P}}(z, \eta_0),\ v\} \\ &= V(x). \end{aligned}$$

Therefore, $V(g) \leq V(x)$. □

Combining the prior results, we can prove $\mathcal{A}$ is UGAS for $\widehat{\mathcal{H}}$, which immediately implies that $\mathcal{A}$ is UGAS for $\mathcal{H}$, since every solution to $\widehat{\mathcal{H}}$ is a solution to $\mathcal{H}$.

Theorem 4.1 (UGAS)**.** *Suppose that $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, and $\mathcal{H}_{\mathrm{S}}$ satisfy Assumptions 4.1–4.4, and that $D_{\mathrm{S}} \cap G_{\mathrm{S}}^{\mathrm{CL}}(D_{\mathrm{S}}) = \varnothing$ and $\mathcal{E}_1$ is compact and nonempty. Suppose, furthermore, that $\mathcal{A}_{\mathrm{P}}$ and $\mathcal{H}_{\mathrm{P}\times 0}$ satisfy Assumptions 4.5 and 4.6. Then, the set $\mathcal{A}$ is UGAS for $\widehat{\mathcal{H}}$.*

Proof. The set $\mathcal{A}$ is compact and nonempty because $\mathcal{A}_{\mathrm{P}}$ and $\mathcal{E}_1$ are compact and nonempty by assumption Assumption 4.6. By Lemmas B.1, B.3, and 4.6–4.8, we have that V in (4.29) is a Lyapunov function candidate with respect to $\mathcal{A}$ for $\mathcal{H}$; the function σ in (4.30) is LSC and positive definite on C with respect to $\mathcal{A}$; the function α in (B.1) is class-$\mathcal{K}_\infty$; and conditions (1.8a)–(1.8c) in Theorem 5.1 are satisfied. A bound on jump frequency as required by (1.9) in Theorem 5.1 remains to be shown.

Take any $r > 0$. We want to show that there exists $\Delta_T > 0$ and $\Delta_J > 0$ such that for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \leq r$, and for all (t_0, j_0) and (t_1, j_1) in $\operatorname{dom}(\varphi)$ such that $|t_1 - t_0| \leq \Delta_T$, we have that $|j_1 - j_0| \leq \Delta_J$. To apply Lemma 4.5, we must construct a compact set $K \subset \mathcal{X}$ such that K is forward pre-invariant for $\mathcal{H}$ and $\mathcal{A} + r\mathbb{B} \subset K$. Consider

$$K := \big\{ x \in \mathcal{X} \;\big|\; V(x) \leq \sup V(\mathcal{A} + r\mathbb{B}) \big\}, \tag{4.37}$$

where $V(\mathcal{A} + r\mathbb{B})$ is a set in $\mathbb{R}$, so $\sup V(\mathcal{A} + r\mathbb{B})$ is the supremum of that set. The set $\mathcal{A} + r\mathbb{B}$ is compact because $\mathcal{A}$ is compact. Let $V^* := \sup V(\mathcal{A} + r\mathbb{B})$. Since V is continuous, $V(\mathcal{A} + r\mathbb{B})$ is bounded, so V^* is finite. Thus, K is the V^*-sublevel set of V. The sublevel set of a continuous function is closed, so K is closed. Every sublevel set of V is forward pre-invariant, as a consequence of (1.8b) and (1.8c) (which imply that V is nonincreasing along flows and jumps, respectively), so K is forward pre-invariant (Proposition 5.7). Finally, to show that K is bounded, we use (1.8a) which gives that for all $x \in K$,

$$\alpha(|x|_{\mathcal{A}}) \leq V(x) \leq V^*.$$

Then, using the fact that α is invertible and α^{-1} is strictly increasing, we have that $|x|_{\mathcal{A}} \leq \alpha^{-1}(V^*)$. In other words, K is contained within the compact set $\mathcal{A} + \alpha^{-1}(V^*)\mathbb{B}$ and (as shown earlier) is closed, so K is compact.

Per Lemma 4.5, there exists $\Delta_T > 0$ such that for each solution φ with $\varphi(0,0) \in K$ and each $(t_1, j_2) \in \mathrm{dom}(\varphi)$ and $(t_2, j_2) \in \mathrm{dom}(\varphi)$,

$$|t_2 - t_1| \leq \Delta_T \quad \Longrightarrow \quad |j_2 - j_1| \leq \Delta_J := 4.$$

Then, it follows from Proposition 5.5 that (5.18) in Theorem 5.1 holds. Therefore, by Theorem 5.1, the set $\mathcal{A}$ is UGpAS for $\mathcal{H}$.

To move from UGpAS to UGAS, we will show that all maximal solutions are complete by applying Lemma 4.4. We need to show that no solutions to $(\widehat{F}, C)$ have a finite escape time, but this follows immediately from the fact that $\mathcal{A}$ is compact and UGpAS. Therefore, by Lemma 4.4, every maximal solution to $\mathcal{H}$ is complete and $\mathcal{A}$ is UGAS for $\widehat{\mathcal{H}}$. □

Chapter 5

Relaxed Lyapunov Conditions for Hybrid Systems

During the development of the results in Chapters 3 and 4, we found that existing Lyapunov theorems for hybrid systems were insufficient for our needs. In particular, we tried to use [3, Thm.3.19(b)] to prove that our uniting feedback strategies rendered $\mathcal{A}_{\text{P}}$ to be UGAS. For our purposes, however, [3, Thm.3.19(b)] was difficult to apply, however, because it required the construction of a continuous, positive definite function ρ such that $-\rho(|x|_{\mathcal{A}})$ is an upper bound on $\sup \mathcal{L}_F V(x)$. In particular, we found that our analysis would be benefit by relaxing the continuity assumption to merely lower semicontinuity and by writing the bound as a function x directly, instead of $|x|_{\mathcal{A}}$ the distance from x to $\mathcal{A}$. To alleviate these difficulties, we developed the results presented in this chapter, which have also been published in [16].

5.1 Introduction

In several Lyapunov-like theorems found in the control theory literature, assumptions are imposed on a function $h : \mathbb{R}^n \to \mathbb{R}$ in the form

$$h(x) \leq -\rho(|x|_{\mathcal{A}}) \qquad \forall x \in \mathbb{R}^n, \tag{5.1}$$

where $|x|_{\mathcal{A}}$ is the distance from $x \in \mathbb{R}^n$ to a set $\mathcal{A}$, and $\rho : [0, \infty) \to [0, \infty)$ is continuous and positive definite. E.g., for a system $\dot{x} = f(x)$ with a differentiable Lyapunov function V, we would use $h := \dot{V}$, where $\dot{V}(x) := \langle \nabla V(x),\, f(x) \rangle$. Examples of assumptions in the form (5.1) include the hybrid Lyapunov theorem [3, Thm. 3.19(3)],

(robust) control Lyapunov functions [3, Defs. 10.2 and 10.14], and input-to-state stability (ISS) Lyapunov functions [17], [18]. In some results, such as Theorems 4.1 and 4.9 in [19], assumptions are given without using the distance function in the form

$$h(x) \leq -\sigma(x) \quad \forall x \in \mathbb{R}^n, \tag{5.2}$$

where $\sigma : \mathbb{R}^n \to [0, \infty)$ is continuous and positive definite with respect to $\mathcal{A}$, but such existing results assume $\mathcal{A} = \{0\}$.

In this chapter, we relax the assumptions on Lyapunov functions for the case where $\mathcal{A}$ is compact. This work builds upon the hybrid Lyapunov theorems [4, Thm. 3.18] and [3, Thm. 3.19]. In particular, [3, Thm. 3.19] asserts that a given set $\mathcal{A}$ is uniformly globally asymptotically stable with respect to a given hybrid system $\mathcal{H}$ under given assumptions. In this chapter, we relax the assumptions of [3, Thm. 3.19] by i) relaxing bounds on the rate of change of V that are given as a function of the distance from $\mathcal{A}$, as in (5.1), to only a function of the state, as in (5.2), ii) allowing for bounds to be lower semicontinuous instead of continuous, iii) relaxing the typical $\mathcal{K}_\infty$ upper-bound on V, and iv) simplifying conditions on hybrid time domains when V is merely nondecreasing during flows or across jumps. We prove our results in the context of hybrid dynamical systems, with the results for discrete-time and continuous-time systems following as special cases. Along the way, we also prove several auxiliary results relating to finding lower bounds for positive definite lower semi-continuous functions that may be useful in other contexts. Throughout this chapter, we use the notation "ρ" to denote positive definite functions on $\mathbb{R}_{\geq 0}$ and "σ" to denote positive definite functions on $\mathbb{R}^n$ with respect to $\mathcal{A}$ (i.e., $\rho \in \mathcal{PD}(0)$ and $\sigma \in \mathcal{PD}(\mathcal{A})$).

The remainder of this chapter is structured as follows. Section 5.2 contains insertion theorems that assert the existence of functions between constraints. Section 5.3 presents our main result, a Lyapunov theorem to show that compact sets are UGAS for hybrid systems, which relaxes results in [3], [4]. In Section 5.3.1, simplified conditions are provided for establishing bounds on the amount of flow versus the number of jumps in a hybrid time domain. Section 5.3.3 presents corollaries of our hybrid Lyapunov theorem for the special cases of continuous-time and discrete-time systems. Section 5.3.2 provides results for establishing bounds on solutions via sublevel sets of Lyapunov functions.

5.2 Insertion Theorems

In the field of topology, an *insertion theorem* asserts the ability to insert a function between two other functions. An example is the Katětov–Tong insertion theorem [20], which allows for the insertion of a continuous function between any USC function $\ell : \mathbb{R} \to \mathbb{R}$ and LSC function $u : \mathbb{R} \to \mathbb{R}$ such that $\ell \leq u$. In this section, we introduce results for inserting positive definite functions between zero and another positive definite function. These results are used, in Section 5.3, to relax conditions such as (5.1) and (5.2).

5.2.1 Insertion Theorems for Positive Definite Functions

Our first result shows that given any LSC function $\sigma_{\text{LSC}} \in \mathcal{PD}(\mathcal{A})$, we can construct a Lipschitz continuous function $\sigma_{\text{C}} \in \mathcal{PD}(\mathcal{A})$ such that $\sigma_{\text{C}} \leq \sigma_{\text{LSC}}$.

Proposition 5.1. *Consider a closed set $\mathcal{X} \subset \mathbb{R}^n$, a compact set $\mathcal{A} \subset \mathcal{X}$, a function $\sigma_{\text{LSC}} : \mathcal{X} \to \mathbb{R}_{\geq 0}$, and any $\ell > 0$. Let $\sigma_{\text{C}} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ be defined by*

$$\sigma_{\text{C}}(x) := \inf_{x^* \in \mathcal{X}} \big(\ell |x^* - x| + \sigma_{\text{LSC}}(x^*)\big) \qquad \forall x \in \mathcal{X}. \tag{5.3}$$

If σ_{LSC} is in $\mathcal{PD}(\mathcal{A})$ and LSC, then σ_{C} is in $\mathcal{PD}(\mathcal{A})$, ℓ–Lipschitz continuous, and

$$\sigma_{\text{C}}(x) \leq \sigma_{\text{LSC}}(x) \qquad \forall x \in \mathcal{X}. \tag{5.4}$$

Proof. First, we will establish that for each $x \in \mathcal{X}$, the objective function $x^* \mapsto \ell |x^* - x| + \sigma_{\text{LSC}}(x^*)$ attains a minimum for some $x^* \in \mathcal{X}$ and that the minimum point is within a ball with radius $\sigma_{\text{LSC}}(x_0)$ centered at x_0. Take any $x_0 \in \mathcal{X}$ and let $K := x_0 + \sigma_{\text{LSC}}(x_0)\mathbb{B}$, which is compact, and take any $x_1 \in \mathcal{X} \setminus K$. Since x_1 is not in K,

$$\ell |x_1 - x_0| > \sigma_{\text{LSC}}(x_0).$$

Using the fact that $\sigma_{\text{LSC}}(x_1) \geq 0$, we find

$$\ell |x_1 - x_0| + \sigma_{\text{LSC}}(x_1) \geq \ell |x_1 - x_0| > \sigma_{\text{LSC}}(x_0).$$

Thus, for all $x_0 \in \mathcal{X}$ and $x_1 \in \mathcal{X} \setminus K$,

$$\ell |x_1 - x_0| + \sigma_{\text{LSC}}(x_1) > \sigma_{\text{LSC}}(x_0),$$

so excluding all such points x_1 from the domain of the infimum does not change the value of the infimum. In other words, restricting the domain of the infimum to K does not change the value of $\sigma_{\mathrm{C}}(x_0)$:

$$\sigma_{\mathrm{C}}(x_0) = \inf_{x^* \in \mathcal{X}} \big(\ell|x_0 - x^*| + \sigma_{\mathrm{LSC}}(x^*)\big) = \inf_{x^* \in \mathcal{X} \cap K} \big(\ell|x_0 - x^*| + \sigma_{\mathrm{LSC}}(x^*)\big).$$

An LSC function restricted to a compact set always attains a minimum, so there exists $x^* \in \mathcal{X} \cap K$ such that

$$\sigma_{\mathrm{C}}(x_0) = \ell|x_0 - x^*| + \sigma_{\mathrm{LSC}}(x^*). \tag{5.5}$$

Furthermore, since x^* is a minimum point, the objective function cannot be smaller at x_0, so we establish that (5.4) holds:

$$\sigma_{\mathrm{C}}(x_0) = \ell|x_0 - x^*| + \sigma_{\mathrm{LSC}}(x^*) \leq \big(\ell|x_0 - x_0| + \sigma_{\mathrm{LSC}}(x_0)\big) = \sigma_{\mathrm{LSC}}(x_0).$$

Next we establish that σ_{C} is positive definite with respect to $\mathcal{A}$. If $x_0 \in \mathcal{A}$, then $\sigma_{\mathrm{C}}(x_0) = \sigma_{\mathrm{LSC}}(x_0) = 0$. Suppose, instead, that $x_0 \in \mathcal{A}$. Then, x^* in (5.5) must either be x_0 or not x_0. If $x_* = x_0$, then

$$\sigma_{\mathrm{C}}(x_0) = \underbrace{\ell|x_0 - x^*|}_{=0} + \underbrace{\sigma_{\mathrm{LSC}}(x^*)}_{>0} > 0.$$

If $x^* \neq x_0$, then

$$\sigma_{\mathrm{C}}(x_0) = \underbrace{\ell|x_0 - x^*|}_{>0} + \underbrace{\sigma_{\mathrm{LSC}}(x^*)}_{\geq 0} > 0.$$

Therefore, $\sigma_{\mathrm{C}}(x_0) > 0$ for all $x_0 \notin \mathcal{A}$, thereby proving σ_{C} is positive definite with respect to $\mathcal{A}$.

Next, we will prove σ_{C} is ℓ–Lipschitz. Take any $x_0 \in \mathcal{X}$ and $x_1 \in \mathcal{X}$. Because the minimum is attained on $\mathcal{X} \cap K$, there exists $x_0^* \in \mathcal{X} \cap K$ such that

$$\sigma_{\mathrm{C}}(x_0) = \ell|x_0^* - x_0| + \sigma_{\mathrm{LSC}}(x_0^*).$$

Additionally,

$$\sigma_{\mathrm{C}}(x_1) \leq \ell|x_0^* - x_1| + \sigma_{\mathrm{LSC}}(x_0^*).$$

Therefore,

$$\sigma_{\mathrm{C}}(x_1) - \sigma_{\mathrm{C}}(x_0) \leq \ell|x_0^* - x_1| + \sigma_{\mathrm{LSC}}(x_0^*) - \big(\ell|x_0^* - x_0| + \sigma_{\mathrm{LSC}}(x_0^*)\big) = \ell|x_0^* - x_1| - \ell|x_0^* - x_0|.$$

By the inverse triangle inequality

$$|x_0^* - x_1| - |x_0^* - x_0| \leq \big||x_0^* - x_1| - |x_0^* - x_0|\big| \leq \big|(x_0^* - x_1) - (x_0^* - x_0)\big| = |x_1 - x_0|.$$

Therefore,

$$\sigma_{\mathrm{C}}(x_1) - \sigma_{\mathrm{C}}(x_0) \leq \ell |x_1 - x_0|.$$

The values x_0 and x_1 were arbitrarily chosen from $\mathcal{X}$, so we can switch them, producing

$$\sigma_{\mathrm{C}}(x_0) - \sigma_{\mathrm{C}}(x_1) = -(\sigma_{\mathrm{C}}(x_1) - \sigma_{\mathrm{C}}(x_0)) \leq \ell |x_0 - x_1| = \ell |x_1 - x_0|.$$

Therefore, we have established that σ_{C} is Lipschitz continuous with Lipschitz constant ℓ, since

$$|\sigma_{\mathrm{C}}(x_1) - \sigma_{\mathrm{C}}(x_0)| \leq \ell |x_1 - x_0| \qquad \forall x_0, x_1 \in \mathcal{X}. \qquad \square$$

Example 5.1. To see why σ_{LSC} is assumed to be LSC in Proposition 5.1, consider $\sigma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ defined for all $x \geq 0$ by

$$\sigma(x) := \begin{cases} x(1-x) & \text{if } x \in [0, 1) \\ 1 & \text{if } x \geq 1. \end{cases}$$

Although σ is positive definite with respect to $\mathcal{A} := \{0\}$, it cannot be lower bound by a continuous function in $\mathcal{PD}(\mathcal{A})$ because σ is not LSC at $x = 1$ and

$$\liminf_{x \to 1} \sigma(x) = 0.$$

In particular, for any continuous function $\sigma_{\mathrm{C}} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that $\sigma_{\mathrm{C}}(x) \leq \sigma(x)$, it must be that $\sigma_{\mathrm{C}}(1) = 0$ because σ_{C} is squeezed to zero by σ as x approaches 1 from the left. $\diamond$

The next result allows us to weaken assumptions in the form of (5.1), with ρ continuous, into an inequality in the form of (5.2) with σ LSC.

Lemma 5.1. *Consider a continuous function $\sigma_{\mathrm{C}} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and a compact set $\mathcal{A} \subset \mathbb{R}^n$. If $\sigma_{\mathrm{C}} \in \mathcal{PD}(\mathcal{A})$, then*

$$r \mapsto \rho_{\mathrm{LSC}}(r) := \inf\big\{\sigma_{\mathrm{C}}(x) : |x|_{\mathcal{A}} = r\big\} \qquad \forall r \geq 0 \tag{5.6}$$

is LSC, positive definite, and satisfies

$$\rho_{\mathrm{LSC}}(|x|_{\mathcal{A}}) \leq \sigma_{\mathrm{C}}(x) \qquad \forall x \in \mathbb{R}^n. \tag{5.7}$$

Proof. Let $\mathcal{D} : \mathbb{R}_{\geq 0} \rightrightarrows \mathbb{R}^n$ be defined by

$$r \mapsto \mathcal{D}(r) := \left\{ x \in \mathbb{R}^n : |x|_{\mathcal{A}} = r \right\} \qquad \forall r \geq 0.$$

Since $\mathcal{A}$ is compact, the set $\mathcal{D}(r)$ is nonempty and compact for all $r \geq 0$.[1] Furthermore, since σ_{C} is continuous and $\mathcal{D}$ has compact values for each $r \geq 0$, the restriction of σ_{C} to $\mathcal{D}(r)$ attains a minimum value, therefore

$$\rho_{\mathrm{LSC}}(r) = \min_{x \in \mathcal{D}(r)} \sigma_{\mathrm{C}}(x) \qquad \forall r \geq 0.$$

Next, we prove ρ_{LSC} is positive definite and satisfies (5.7). Take any $r \geq 0$. If $r = 0$, then $|x|_{\mathcal{A}} = r$ implies $x \in \mathcal{A}$, so $\rho_{\mathrm{LSC}}(0) = \min\{\sigma_{\mathrm{C}}(x) \mid x \in \mathcal{A}\} = 0$, since σ_{C} is identically zero on $\mathcal{A}$. Suppose, instead $r > 0$. Since σ_{C} is positive outside $\mathcal{A}$ and $\mathcal{D}(r)$ is disjoint from $\mathcal{A}$, we have that $\rho_{\mathrm{LSC}}(r) > 0$. Therefore, ρ_{LSC} is positive definite.

Take any $x \in \mathbb{R}^n$. Then, we find ρ_{LSC} satisfies (5.7):

$$\rho_{\mathrm{LSC}}(|x|_{\mathcal{A}}) = \min_{x' \in \mathcal{D}(|x|_{\mathcal{A}})} \sigma_{\mathrm{C}}(x') \leq \sigma_{\mathrm{C}}(x) \qquad \forall x \in \mathbb{R}^n. \tag{5.8}$$

Next, we prove ρ_{LSC} is LSC. For all $r \geq 0$, let

$$\mathcal{M}(r) = \{x_{\min} \in \mathcal{D}(r) \mid \sigma_{\mathrm{C}}(x_{\min}) = \rho_{\mathrm{LSC}}(r)\},$$

which is the (nonempty) set of points that minimize σ_{C} in $\mathcal{D}(r)$.

To prove lower semicontinuity at each $r_1 \geq 0$, we want to show that for all $\varepsilon > 0$, there exists $\delta > 0$ such that for all $r_2 \in \mathbb{R}_{\geq 0}$,

$$|r_1 - r_2| \leq \delta \implies \rho_{\mathrm{LSC}}(r_2) \geq \rho_{\mathrm{LSC}}(r_1) - \varepsilon.$$

Take any $r_1 \geq 0$ and let $K_r := \mathcal{A} + (r_1 + 42)\mathbb{B} \subset \mathbb{R}^n$. Since σ_{C} is continuous and K_r is compact, the restriction of σ_{C} to K_r is uniformly continuous. From the definition of uniform continuity, for all $\varepsilon > 0$, there exists $\delta > 0$ such that for all $x_1, x_2 \in K_r$,

$$|x_1 - x_2| \leq \delta \implies |\sigma_{\mathrm{C}}(x_1) - \sigma_{\mathrm{C}}(x_2)| \leq \varepsilon.$$

Without loss of generality, suppose $\delta < 42$ (so that $\mathcal{D}(r_1) + \delta\mathbb{B} \subset K_r$). Take any $r_2 \geq 0$ such that $|r_1 - r_2| \leq \delta$. We will show $\rho_{\mathrm{LSC}}(r_2) \geq \rho_{\mathrm{LSC}}(r_1) - \varepsilon$.

[1] To see why compactness is important, consider that if $\mathcal{A} := \mathbb{R}^n$, then $\mathcal{D}(1) = \varnothing$.

Take any $x_{\min} \in \mathcal{M}(r_1)$ and $x_r \in \mathcal{D}(r_1)$. From the definition of $\mathcal{M}(r_1)$ and (5.8),

$$\sigma_{\mathrm{C}}(x_{\min}) = \rho_{\mathrm{LSC}}(|x_{\min}|_{\mathcal{A}}) = \rho_{\mathrm{LSC}}(|x_r|_{\mathcal{A}}) \leq \sigma_{\mathrm{C}}(x_r).$$

Consider $\mathcal{D}(r_2)$. Since $|r_1 - r_2| \leq \delta$,

$$\mathcal{D}(r_2) \subset \mathcal{D}(r_1) + \delta\mathbb{B}.$$

Take any $x_2 \in \mathcal{D}(r_2)$. Because of the uniform continuity of σ_{C} on $\mathcal{D}(r_2) \subset \mathcal{D}(r_1) + \delta\mathbb{B} \subset K_r$, and because $\min_{x \in \mathcal{D}(r_1)} \sigma_{\mathrm{C}}(x) = \rho_{\mathrm{LSC}}(r_1)$,

$$\sigma_{\mathrm{C}}(x_2) \geq \rho_{\mathrm{LSC}}(r_1) - \varepsilon.$$

Since this holds for all $x_2 \in \mathcal{D}(r_2)$, we have that

$$\rho_{\mathrm{LSC}}(r_2) = \min_{x_2 \in \mathcal{D}(r_2)} \sigma_{\mathrm{C}}(x_2) \geq \rho_{\mathrm{LSC}}(r_1) - \varepsilon.$$

Therefore, ρ_{LSC} is LSC. □

Remark 5.1. One may suspect that we can relax that assumption in Lemma 5.1 that σ_{C} is continuous to merely lower semicontinuous. This remains an open question, however. The proof approach we have used necessitates continuity of σ_{C} because we use the fact that continuity on a compact set implies uniform continuity. The analogous claim for lower semicontinuity on a compact set is false.

The next example shows a case where the function ρ_{LSC} in Lemma 5.1 is merely LSC—not continuous.

Example 5.2. Consider $\mathcal{A} := \{-1, 1\} \subset \mathbb{R}$, and let $\sigma_{\mathrm{C}}(x) := |x^2 - 1|$ for all $x \in \mathbb{R}$. Then, for all $r \geq 0$,

$$\rho_{\mathrm{LSC}}(r) = \left\{|x^2 - 1| : |x|_{\mathcal{A}} = r\right\} = \begin{cases} r(2 - r) & \text{if } r \leq 1 \\ r(2 + r) & \text{if } r > 1, \end{cases}$$

so ρ_{LSC} jumps from $\rho_{\mathrm{LSC}}(1) = 1$ to $\rho_{\mathrm{LSC}}(1.001) > 2$. ◇

The following result asserts that for every LSC function $\sigma_{\mathrm{LSC}} \in \mathcal{PD}(\mathcal{A})$ with $\mathcal{A}$ compact, we can construct a continuous function $\rho_{\mathrm{C}} \in \mathcal{PD}(0)$ that—when composed with the distance from $\mathcal{A}$, as in (5.1)—is a lower bound on σ_{LSC}.

Proposition 5.2. *Consider a compact set $\mathcal{A} \subset \mathbb{R}^n$. For each LSC function $\sigma_{\text{LSC}} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ in $\mathcal{PD}(\mathcal{A})$ and each $\ell > 0$, there exists an ℓ-Lipschitz continuous and positive definite function $\rho_{\text{C}} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that*

$$\rho_{\text{C}}(|x|_{\mathcal{A}}) \leq \sigma_{\text{LSC}}(x) \quad \forall x \in \mathbb{R}^n. \tag{5.9}$$

Proof. Suppose $\sigma_{\text{LSC}} \in \mathcal{PD}(\mathcal{A})$ is LSC. By Proposition 5.1, there exists a continuous function $\sigma_{\text{C}} \in \mathcal{PD}(\mathcal{A})$ such that

$$\sigma_{\text{C}}(x) \leq \sigma_{\text{LSC}}(x) \quad \forall x \in \mathbb{R}^n.$$

By Lemma 5.1, there exists an LSC and positive definite function $\rho_{\text{LSC}} \in \mathcal{PD}(0)$ such that

$$\rho_{\text{LSC}}(|x|_{\mathcal{A}}) \leq \sigma_{\text{C}}(x) \quad \forall x \in \mathbb{R}^n.$$

Again, by Proposition 5.1, for any $\ell > 0$ there exists an ℓ-Lipschitz continuous function $\rho_{\text{C}} \in \mathcal{PD}(0)$ such that

$$\rho_{\text{C}}(r) \leq \rho_{\text{LSC}}(r) \qquad \forall r \geq 0.$$

Thus, for all $x \in \mathbb{R}^n$,

$$\rho_{\text{C}}(|x|_{\mathcal{A}}) \leq \rho_{\text{LSC}}(|x|_{\mathcal{A}}) \leq \sigma_{\text{C}}(x) \leq \sigma_{\text{LSC}}(x). \qquad \square$$

5.2.2 Insertion Theorems for Class $\mathcal{K}_\infty$ Functions

This section shows that for any nonempty compact set $\mathcal{A}$ and continuous function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, there exists $\alpha \in \mathcal{K}_\infty$ such that

$$V(x) \leq \alpha(|x|_{\mathcal{A}}) \quad \forall x \in \text{dom}(V). \tag{5.10}$$

The following lemma establishes that any continuous function ρ_{C} on $\mathbb{R}$ with $\rho_{\text{C}}(0) = 0$ can be upper bounded by a class-$\mathcal{K}_\infty$ function.

Lemma 5.2. *Consider a continuous function $\rho_{\text{C}} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. If $\rho_{\text{C}}(0)$=0, then there exists a smooth function $\alpha \in \mathcal{K}_\infty$ such that*

$$\rho_{\text{C}}(r) \leq \alpha(r) \quad \forall r \geq 0. \tag{5.11}$$

Proof. We define $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ for each $r \geq 0$ by

$$\alpha(r) := \sup_{r' \in [0,r]} |r - r'| + \rho_{\mathrm{C}}(r').$$

The function α is continuous by [21, Prop. 2.9] because $r' \mapsto |r - r'| + \rho_{\mathrm{C}}(r')$ is (single-valued) continuous and $r \mapsto [0, r]$ is (set-valued) continuous.

At zero, $\alpha(0) = \rho_{\mathrm{C}}(0) = 0$. To see that α is strictly increasing, take any $r_0 \geq 0$ and $r_1 > r_0$. Then, since $|r_1 - r_0| > 0$,

$$\alpha(r_1) \geq |r_1 - r_0| + \rho_{\mathrm{C}}(r_0) > \alpha(r_0).$$

Finally, we see that $\lim_{r \to \infty} \alpha(r) = \infty$ because for all $r \geq 0$,

$$\alpha(r) \geq |r - 0| + \rho_{\mathrm{C}}(0) = r.$$

Therefore, α is a class–$\mathcal{K}_\infty$ function. □

Lemma 5.2 leads naturally to the following lemma that we use to eliminate the $\mathcal{K}_\infty$ upper bound in Lyapunov theorems, such as [3, Thm. 3.19], when $\mathcal{A}$ is compact.

Lemma 5.3. *Consider a closed and nonempty set $\mathcal{X} \subset \mathbb{R}^n$, a compact and nonempty set $\mathcal{A} \subset \mathcal{X}$, and a continuous function $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$. If $V(x) = 0$ for all $x \in \mathcal{A}$, then there exists $\alpha \in \mathcal{K}_\infty$ such that $V(x) \leq \alpha(|x|_{\mathcal{A}})$ for all $x \in \mathcal{X}$.*

To see why the conclusion in Lemma 5.3 does not generally hold if $\mathcal{A}$ is unbounded, consider $\mathcal{A} := \mathbb{R} \times \{0\}$ and $(x_1, x_2) \mapsto V(x_1, x_2) = (|x_1| + 1)|x_2|$.

5.3 Lyapunov Theorems for Compact Sets

In this section, we present a Lyapunov theorem with relaxed assumptions for showing that a compact set is UGpAS for a hybrid system. The following definition establishes the class of functions permissible as Lyapunov functions.

Definition 5.1 ([3, Def. 3.17])**.** Consider a hybrid system $\mathcal{H} = (C, F, D, G)$ on $\mathbb{R}^n$ and a set $\mathcal{A} \subset \mathbb{R}^n$. A function $V : \mathrm{dom}(V) \subset \mathbb{R}^n \to \mathbb{R}$ is a *Lyapunov function candidate with respect to $\mathcal{A}$ for $\mathcal{H}$* if $\overline{C} \cup D \cup G(D) \subset \mathrm{dom}(V)$, V is positive definite on $C \cup D \cup G(D)$ with respect to $\mathcal{A}$, V is continuous, and V is locally Lipschitz on an open neighborhood of $\overline{C}$. ◇

A key part of any Lyapunov-like theorem is establishing an upper bound on the change in a Lyapunov function candidate V. For hybrid systems, the following functions provide upper bounds on the rate of V during flows and across jumps.

Definition 5.2. Consider a hybrid system $\mathcal{H} = (C, F, D, G)$ on $\mathbb{R}^n$, a nonempty set $\mathcal{A} \subset \mathbb{R}^n$, and a Lyapunov function candidate V with respect to $\mathcal{A}$ for $\mathcal{H}$. Recall that $T_C(x)$ is the contingent cone of C at x. We define

$$u_{\mathrm{C}}(x) := \sup_{f \in F(x) \cap T_C(x)} V^\circ(x, f) \qquad \forall x \in C \tag{5.12}$$

$$u_{\mathrm{D}}(x) := \sup_{g \in G(x)} V(g) - V(x) \qquad \forall x \in D. \tag{5.13}$$

The suprema are defined as $-\infty$ if the domains are empty (that is, if $F(x) \cap T_C(x) = \varnothing$, then $u_{\mathrm{C}}(x) = -\infty$). $\diamond$

For any solution φ to $\mathcal{H}$ and all $(t_1, j_1), (t_2, j_2) \in \mathrm{dom}(\varphi)$,

$$V\big(\varphi(t_2, j_2)\big) - V\big(\varphi(t_1, j_1)\big) \leq \int_{t_1}^{t_2} u_{\mathrm{C}}\big(\varphi(t, \underline{j}(t))\big)\, dt + \sum_{j=j_1}^{j_2-1} u_{\mathrm{D}}\big(\varphi(\underline{t}(j), j)\big),$$

where $\underline{j}$ and $\underline{t}$ are defined for all $(t, j) \in \mathrm{dom}(\varphi)$ by

$$j \mapsto \underline{t}(j) := \min\big\{t' \;\big|\; (t', j) \in \mathrm{dom}(\varphi)\big\} \quad \text{and} \quad t \mapsto \underline{j}(t) := \min\big\{j' \;\big|\; (t, j') \in \mathrm{dom}(\varphi)\big\}.$$

The main result of this chapter, which follows the structure of [3, Thm. 3.19(3)], is presented next. In particular, Theorem 5.1 provides sufficient conditions for a compact set to be UGpAS.

Theorem 5.1. *Consider a hybrid system $\mathcal{H} = (C, F, D, G)$ on $\mathbb{R}^n$, a nonempty compact set $\mathcal{A} \subset \mathbb{R}^n$, and a Lyapunov function candidate V with respect to $\mathcal{A}$ for $\mathcal{H}$. Suppose there exists $\alpha_1 \in \mathcal{K}_\infty$ such that*

$$\alpha_1(|x|_{\mathcal{A}}) \leq V(x) \quad \forall x \in C \cup D \cup G(D). \tag{5.14}$$

Then, the set $\mathcal{A}$ is UGpAS for $\mathcal{H}$ if any of the following conditions hold:

(a) Strict decrease during flows and jumps: *There exist LSC functions $\sigma_c, \sigma_d \in \mathcal{PD}(\mathcal{A})$ such that*

$$u_{\mathrm{C}}(x) \leq -\sigma_c(x) \qquad \forall x \in C \tag{5.15}$$

$$u_{\mathrm{D}}(x) \leq -\sigma_d(x) \qquad \forall x \in D. \tag{5.16}$$

(b) Strict decrease during flows, nonincreasing at jumps: *There exists an LSC function $\sigma_c \in \mathcal{PD}(\mathcal{A})$ such that*

$$u_{\mathrm{C}}(x) \leq -\sigma_c(x) \qquad \forall x \in C \qquad (*5.15)$$

$$u_{\mathrm{D}}(x) \leq 0 \qquad \forall x \in D, \qquad (5.17)$$

and, for each $r > 0$, there exist $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ such that for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$,

$$t \geq \gamma_r(t + j) - N_r \quad (t, j) \in \mathrm{dom}(\varphi). \qquad (5.18)$$

(c) Strict decrease at jumps, nonincreasing during flows: *There exists an LSC function $\sigma_d \in \mathcal{PD}(\mathcal{A})$ such that*

$$u_{\mathrm{C}}(x) \leq 0 \qquad \forall x \in C \qquad (5.19)$$

$$u_{\mathrm{D}}(x) \leq -\sigma_d(x) \qquad \forall x \in D, \qquad (*5.16)$$

and, for each $r > 0$, there exist $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ such that for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$,

$$j \geq \gamma_r(t + j) - N_r \quad \forall (t, j) \in \mathrm{dom}(\varphi). \qquad (5.20)$$

(d) *This item is skipped to keep the enumeration consistent with* [3, Thm. 3.19].

(e) Bounded flow time: *There exist $\lambda \in \mathbb{R}$ and an LSC function $\sigma_d \in \mathcal{PD}(\mathcal{A})$ such that*

$$u_{\mathrm{C}}(x) \leq \lambda V(x) \qquad \forall x \in C \qquad (5.21)$$

$$u_{\mathrm{D}}(x) \leq -\sigma_d(x) \qquad \forall x \in D, \qquad (*5.16)$$

and, for each $r > 0$, there exists $T_r \geq 0$ such that for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$,

$$\mathrm{dom}(\varphi) \subset [0, T_r] \times \mathbb{N}. \qquad (5.22)$$

(f) Finite number of jumps: *There exist an LSC function $\sigma_c \in \mathcal{PD}(\mathcal{A})$ and a continuous function $\lambda : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ with $\lambda(0) = 0$ such that*

$$u_{\mathrm{C}}(x) \leq -\sigma_c(x) \qquad \forall x \in C \qquad (*5.15)$$

$$V(g) \leq \lambda(V(x)) \qquad \forall x \in D,\ \forall g \in G(x), \qquad (5.23)$$

and, for each $r \geq 0$, there exists $J_r \in \mathbb{N}$ such that for every solution φ to $\mathcal{H}$,

$$\mathrm{dom}(\varphi) \subset \mathbb{R} \times \{0, 1, \ldots, J_r\}. \qquad (5.24)$$

Proof. We will apply [3, Thm. 3.19(3)] to prove each case. By assumption and Lemma 5.3 with $\mathcal{U} := C \cup D \cup G(D)$, there exist $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ satisfying [3, Eq. 3.26]:

$$\alpha_1(|x|_{\mathcal{A}}) \leq V(x) \leq \alpha_2(|x|_{\mathcal{A}}) \qquad \forall x \in C \cup D \cup G(D).$$

Suppose that there exists an LSC function $\sigma_c \in \mathcal{PD}(\mathcal{A})$ that satisfies (5.15). By Proposition 5.2, there exist a Lipschitz continuous function $\rho_c \in \mathcal{PD}(0)$ such that $\rho_c(|x|_{\mathcal{A}}) \leq \sigma_c(x)$ for all $x \in \mathbb{R}^n$. Therefore, under the assumption that there exists an LSC function $\sigma_c \in \mathcal{PD}(\mathcal{A})$ such that (5.15) holds, then there exists a continuous function $\rho_c \in \mathcal{PD}(0)$ such that [3, Eq. 3.27] holds:

$$u_C(x) \leq -\sigma_c(x) \leq -\rho_c(|x|_{\mathcal{A}}) \qquad \forall x \in C. \tag{5.25}$$

By a similar process, we find if there exists $\sigma_d \in \mathcal{PD}(\mathcal{A})$ such that (5.16) holds, then there exists a continuous function $\rho_d \in \mathcal{PD}(0)$ such that [3, Eq. 3.28] holds.

(a) By assumption, there exist LSC functions $\sigma_c, \sigma_d \in \mathcal{PD}(\mathcal{A})$ such that (5.15) and (5.16) hold. Then, as shown above, there exist continuous $\rho_c, \rho_d \in \mathcal{PD}(0)$ such that [3, Eqs. 3.27-3.28] hold. Therefore, by [3, Thm. 3.19(3a)], the set $\mathcal{A}$ is UGpAS.

(b) By assumption, there exists an LSC function $\sigma_c \in \mathcal{PD}(\mathcal{A})$ such that (5.15) holds. Then, as in part (a), there exists $\rho_c \in \mathcal{PD}(0)$ such that [3, Eq. 3.27] holds. By assumption, [3, Eq. 3.28] holds with $\rho_c \equiv 0$.

Additionally, by assumption, for each $r > 0$, there exist $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ such that for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$, and for all $(t, j) \in \mathrm{dom}(\varphi)$, $t \geq \gamma_r(t + j) - N_r$. By Lemma B.7 (in Section B.2.1), for all $T \geq 0$ and $(t, j) \in \mathrm{dom}(\varphi)$ with $t + j \geq T$,

$$t \geq \gamma_r(T) - N_r.$$

Therefore, by [3, Thm. 3.19(3b)], the set $\mathcal{A}$ is UGpAS.

(c) By assumption, there exists a lower semicontinuous function $\sigma_d \in \mathcal{PD}(\mathcal{A})$ such that (5.16) holds. Then, as in part (a), there exists $\rho_d \in \mathcal{PD}(0)$ such that [3, Eq. 3.28] holds. By assumption, [3, Eq. 3.27] holds with $\rho_c \equiv 0$.

Additionally, by assumption, for each $r > 0$, there exist $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ such that for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$, and for all $(t, j) \in$

$\operatorname{dom}(\varphi)$, $j \geq \gamma_r(t+j) - N_r$. By Lemma B.8 (in Section B.2.1), for all $J \geq 0$ and $(t,j) \in \operatorname{dom}(\varphi)$ with $t + j \geq J$,

$$j \geq \gamma_r(J) - N_r.$$

Therefore, by [3, Thm. 3.19(3c)], the set $\mathcal{A}$ is UGpAS.

(d) This case is omitted.

(e) The inequality [3, Eqs. 3.28] is satisfied, as shown above in (a). Equation (5.21) is satisfied by assumption. Take any $r > 0$ and take $T_r \geq 0$ such that for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$,

$$\operatorname{dom}(\varphi) \subset [0, T_r] \times \mathbb{N}.$$

Thus, for every solution φ to $\mathcal{H}$,

$$|\varphi(0,0)|_{\mathcal{A}} \in (0,r], (t,j) \in \operatorname{dom}(\varphi) \quad \implies \quad t \leq T_r,$$

thereby satisfying all of the assumptions of [3, Thm. 3.19(3e)].

(f) The rate inequalities in this case are either true by assumption or can be shown via the methods above. By Lemma B.9 in Section B.2.1, the given assumption on J_r is equivalent to the assumption on the existence of $\gamma \in \mathcal{K}$ and $J \geq 0$ in [3, Thm. 3.19(3f)]. Therefore, $\mathcal{A}$ is UGpAS for $\mathcal{H}$ per [3, Thm. 3.19(3f)]. □

The next example Theorem 5.1 can be applied to prove that a set is UGpAS without needing to construct a bound on u_{C} in the form of (5.1).

Example 5.3 (Bouncing Ball)**.** Consider a bouncing ball modeled as in [4, Ex. 3.19] with height $x_1 \geq 0$ and vertical velocity $x_2 \in \mathbb{R}$. The bouncing ball is modeled as the hybrid system $\mathcal{H} := (C, F, D, G)$ with state $x := (x_1, x_2) \in \mathbb{R}^2$ and dynamics given by

$$F(x) := \begin{bmatrix} x_2 \\ -\gamma \end{bmatrix} \qquad \forall x \in C := \left\{ x \in \mathbb{R}^2 \;\middle|\; x_1 > 0 \right\}$$

$$G(x) := \begin{bmatrix} 0 \\ -\lambda x_2 \end{bmatrix} \qquad \forall x \in D := \left\{ x \in \mathbb{R}^2 \;\middle|\; x_1 = 0, x_2 < 0 \right\},$$

where $\gamma > 0$ is acceleration due to gravity and $\lambda \in [0, 1)$ is the coefficient of restitution when the ball hits the floor. The sets C and D are not closed, so (A1) of the hybrid basic conditions is violated. To show that $\mathcal{A} := \{(0, 0)\}$ is UGpAS, we take the Lyapunov function candidate

$$x \mapsto V(x_1, x_2) := \big(1 + \theta \operatorname{atan}(x_2)\big)\big(x_2^2/2 + \gamma x_1\big)$$

where $\theta := \big(1 - \lambda^2\big)/\big(2 + 2\lambda^2\big)$. Equation (5.14) holds with

$$s \mapsto \alpha_1(s) := \tfrac{1}{1-\theta} \min\big\{s^2/4,\ \gamma s/\sqrt{2}\big\}.$$

Since V is continuously differentiable and F is single valued, we have that for all $x := (x_1, x_2) \in C$,

$$u_{\mathrm{C}}(x) = \langle \nabla V(x),\, F(x) \rangle = -\gamma\theta\left(x_2^2/2 + \gamma x_1\right)/\left(1 + x_2^2\right).$$

Thus, u_{C} is continuous and negative definite with respect to $\mathcal{A}$, and $\sigma_c := -u_{\mathrm{C}}$ satisfies (5.15). For each $x := (0, x_2) \in D$,

$$u_{\mathrm{D}}(x) = \Big[\lambda^2 - 1 - \theta\big(\operatorname{atan}(\lambda x_2)\lambda^2 + \operatorname{atan}(x_2)\big)\Big]\frac{x_2^2}{2},$$

which is continuous. For any $x \in D \setminus \mathcal{A}$, we have

$$\theta\big(\operatorname{atan}(-\lambda x_2)\lambda^2 - \operatorname{atan}(x_2)\big) \le \theta(\lambda^2 + 1) < \lambda^2 - 1 < 0,$$

so u_{D} is negative definite. Thus, $\sigma_d := -u_{\mathrm{D}}$ satisfies (5.16). Therefore, by Theorem 5.1, $(0, 0)$ is UGpAS for $\mathcal{H}$. ◇

As we saw in Examples 5.3 and 5.4, if u_{C} and u_{D} are negative definite and USC, then we can simply use the functions $\sigma_{\mathrm{C}} \equiv -u_{\mathrm{C}}$ and $\sigma_{\mathrm{C}} \equiv -u_{\mathrm{D}}$ for the assumptions in Theorem 5.1. This approach holds in general if we introduce additional (weak) assumptions on F and G, which are a subset of the hybrid basic conditions. In particular, F is assumed to be locally bounded and OSC, as in (A2), but not assumed to have convex values. The assumptions on G match (A3).

Proposition 5.3. *Consider a hybrid system $\mathcal{H} = (C, F, D, G)$ on $\mathbb{R}^n$, a compact set $\mathcal{A} \subset \mathbb{R}^n$, and a Lyapunov function candidate V with respect to $\mathcal{A}$ for $\mathcal{H}$. Suppose F is OSC and locally bounded, and u_{C} is negative definite with respect to $\mathcal{A}$. Then, $\sigma_c \equiv -u_{\mathrm{C}}$ is LSC and satisfies* (5.15)*, and there exists a continuous function $\rho \in \mathcal{PD}(0)$ such that $u_{\mathrm{C}}(x) \le -\rho(|x|_{\mathcal{A}})$ for all $x \in \mathbb{R}^n$.*

Proof. First, we will show u_{C} is (single valued) USC. By [22, Prop. 2.1.5], the Clarke generalized gradient $\partial^\circ V$ is USC, and, since it has closed values, it is OSC. The function F is (set-valued) USC because it is OSC and locally bounded. From the definition of u_{C}, the value of $u_{\mathrm{C}}(x)$ is the maximum of a continuous function (the inner product) over USC set-valued maps. Additionally, F and $\partial^\circ V$ have compact values. Therefore, u_{C} is USC [21, Prop. 2.9].

Let $\rho_{\mathrm{LSC}} := -u_{\mathrm{C}}$, which is positive definite and LSC. By Proposition 5.2, there exists $\sigma : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ that is continuous and positive definite such that

$$\rho_{\mathrm{C}}(|x|_{\mathcal{A}}) \leq -u_{\mathrm{C}}(x) \quad \forall x \in \mathbb{R}^n.$$

Therefore, flipping the signs, we find

$$u_{\mathrm{C}}(x) \leq -\rho_{\mathrm{C}}(|x|_{\mathcal{A}}) \quad \forall x \in \mathbb{R}^n. \qquad \square$$

Proposition 5.4. *Consider a hybrid system $\mathcal{H} = (C, F, D, G)$ on $\mathbb{R}^n$, a compact set $\mathcal{A} \subset \mathbb{R}^n$, and a Lyapunov function candidate V with respect to $\mathcal{A}$ for $\mathcal{H}$. Suppose that G is OSC and locally bounded, and u_{D} is negative definite with respect to $\mathcal{A}$. Then, $\sigma_d \equiv -u_{\mathrm{D}}$ is LSC and satisfies* (5.16)*, and there exists a continuous function $\rho \in \mathcal{PD}(0)$ such that $u_{\mathrm{D}}(x) \leq -\rho(|x|_{\mathcal{A}})$ for all $x \in \mathbb{R}^n$.*

5.3.1 Simplified Assumptions on Hybrid Time Domains

In Theorem 5.1, the conditions on the hybrid time domain of solutions given in (5.18) of case (b) and (5.20) of case (c) are rather non-intuitive and are often difficult to show. Thus, in Propositions 5.5 and 5.6, we provide sufficient conditions for (5.18) and (5.20), respectively, that are easier to check while remaining general enough to apply to most systems that satisfy (5.18) or (5.20).

Proposition 5.5. *Consider a hybrid system $\mathcal{H}$ and a nonempty closed set $\mathcal{A}$. Suppose that for each $r \geq 0$, there exist $\Delta_T > 0$ and $\Delta_J > 0$ such that for every solution φ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$ and for every $(t_0, j_0), (t_1, j_1) \in \mathrm{dom}(\varphi)$,*

$$|t_1 - t_0| \leq \Delta_T \implies |j_1 - j_0| \leq \Delta_J. \tag{5.26}$$

Then, for each $r \geq 0$, there exist $N_r \geq 0$ and $\gamma_r \in \mathcal{K}_\infty$ such that for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$,

$$t \geq \gamma_r(t + j) - N_r \qquad \forall (t, j) \in \mathrm{dom}(\varphi). \tag{*5.18}$$

Informally, the assumptions of Proposition 5.5 state that for every solution that starts within a given distance of $\mathcal{A}$, there exists a bound Δ_J on the number of jumps that can occur during any time interval a given length Δ_T.

Proof for Proposition 5.5. Take any $r \geq 0$. By assumption, there exists $\Delta_T > 0$ and $\Delta_J > 0$ such that (5.26) holds for every solution φ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$, and for every $(t_0, j_0), (t_1, j_1) \in \mathrm{dom}(\varphi)$.

Take any solution φ to $\mathcal{H}$ such that $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$. We will show that

$$j \leq \Delta_J + \frac{\Delta_J}{\Delta_T} t \quad \forall (t, j) \in \mathrm{dom}(\varphi). \tag{5.27}$$

Let $T := \sup_t \mathrm{dom}(\varphi)$. If $T = 0$, then (5.27) follows directly from (5.26) with $t_0 = t_1 = 0$. Suppose, instead, that $T > 0$. For each $i \in \mathbb{N}$, let $\tau_i := i\Delta_T$. Let $\mathcal{I} \subset \mathbb{N}$ be the set of all $i \in \mathbb{N}$ such that $\tau_i < T$. For each $i \in \mathcal{I}$, let

$$\bar{\jmath}_i := \sup\{j \mid (t, j) \in \mathrm{dom}(\varphi),\ t \in [\tau_i, \tau_{i+1}]\}. \tag{5.28}$$

Let $(t_0, j_0) := (0, 0) \in \mathrm{dom}(\varphi)$. By (5.26), $|j_1 - j_0| = j_1 \leq \Delta_J$ for all $(t_1, j_1) \in \mathrm{dom}(\varphi)$ such that $|t_1 - t_0| \leq \tau_1$ since $\tau_1 = \delta T$. Thus, $\bar{\jmath}_0 \leq \Delta_J$. For each $i \in \mathcal{I} \setminus \{0\}$, we have

$$\bar{\jmath}_{i-1} = \max\{j \mid (t, \tau_i) \in \mathrm{dom}(\varphi)\},$$

since j is nondecreasing relative to t for $(t, j) \in \mathrm{dom}(\varphi)$, and $\tau_i < T$ (from the definition of $\mathcal{I}$), so the supremum in (5.28) is over a compact set and a maximum is attained with $t = \tau_i$. Thus,

$$\bar{\jmath}_i - \bar{\jmath}_{i-1} \leq \Delta_J \qquad \forall i \in \mathcal{I} \setminus \{0\}. \tag{5.29}$$

Combining $\bar{\jmath}_0 \leq \Delta_J$ and (5.29), we find

$$\bar{\jmath}_i \leq \Delta_J + i\Delta_J \qquad \forall i \in \mathcal{I}. \tag{5.30}$$

For each $i \in \mathcal{I}$ and each $t \in [\tau_i, \tau_{i+1}]$, substituting $i\delta T$ for τ_i leads to $i \leq t/\delta T$. Thus, from (5.30), we find

$$\bar{\jmath}_i \leq \Delta_J + \frac{\Delta_J}{\Delta_T} t \qquad \forall i \in \mathcal{I},\ \forall t \in [\tau_i, \tau_{i+1}]. \tag{5.31}$$

For each $i \in \mathcal{I}$ and each $(t, j) \in \mathrm{dom}(\varphi)$ such that $t \leq \tau_{i+1}$, the definition of $\bar{\jmath}_i$ in (5.28) gives us that $j \leq \bar{\jmath}_i$. Since, for every $(t, j) \in \mathrm{dom}(\varphi)$ there exists $i \in \mathcal{I}$ such that $t \in [\tau_i, \tau_{i+1}]$, so $j \leq \bar{\jmath}_i \leq \Delta_J + t\Delta_J/\Delta_T$, thereby proving (5.27).

Adding t to both sides of (5.27) produces

$$t + j \leq t + \Delta_J + \frac{\Delta_J}{\Delta_T} t = \left(1 + \frac{\Delta_J}{\Delta_T}\right) t + \Delta_J \quad \forall (t, j) \in \mathrm{dom}(\varphi).$$

Therefore, via algebra,

$$t \geq \left(\frac{\Delta_T}{\Delta_T + \Delta_J}\right)(t + j) - \frac{\Delta_T \Delta_J}{\Delta_T + \Delta_J}.$$

Therefore, (5.18) holds with $N_r := \Delta_T \Delta_J / (\Delta_T + \Delta_J)$ and $\gamma_r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ defined by

$$s \mapsto \gamma_r(s) := \left(\frac{\Delta_T}{\Delta_T + \Delta_J}\right) s.$$

The function γ_r is linear with a positive coefficient (since $\Delta_T > 0$ and $\Delta_J > 0$), so $\gamma_r \in \mathcal{K}_\infty$. □

The analogous result with flows and jumps switched is presented next.

Proposition 5.6. *Consider a hybrid system $\mathcal{H}$ and a nonempty closed set $\mathcal{A}$. Suppose that for each $r \geq 0$, there exists $\Delta_T > 0$ and $\Delta_J > 0$ such that for every solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$ and for all $(t_0, j_0), (t_1, j_1) \in \mathrm{dom}(\varphi)$,*

$$|j_1 - j_0| \leq \Delta_J \implies |t_1 - t_0| \leq \Delta_T. \tag{5.32}$$

Then, for each $r > 0$, there exist $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ such that for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$,

$$j \geq \gamma_r(t + j) - N_r \quad \forall (t, j) \in \mathrm{dom}(\varphi). \tag{*5.20}$$

Proof. Take any $r \geq 0$. By assumption, there exists $\Delta_T > 0$ and $\Delta_J > 0$ such that (5.26) holds for every solution φ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$, and for every $(t_0, j_0) \in \mathrm{dom}(\varphi)$ and $(t_1, j_1) \in \mathrm{dom}(\varphi)$.

Take any solution φ to $\mathcal{H}$ such that $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$. We will show that

$$t \leq \Delta_T + \frac{\Delta_T}{\Delta_J} j \quad \forall (t, j) \in \mathrm{dom}(\varphi). \tag{5.33}$$

Let $J := \sup_j \mathrm{dom}(\varphi)$. If $J = 0$, then (5.33) follows directly from (5.32) with $j_0 = j_1 = 0$. Suppose, instead, that $J > 0$. For each $i \in \mathbb{N}$, let $m_i := i\Delta_J$. Let $\mathcal{I} \subset \mathbb{N}$ be the set of all $i \in \mathbb{N}$ such that $m_i < J$. For each $i \in \mathcal{I}$, let

$$\bar{t}_i := \sup\{j \mid (t, j) \in \mathrm{dom}(\varphi),\ t \in [m_i, m_{i+1}]\}. \tag{5.34}$$

Let $(t_0, j_0) := (0, 0) \in \mathrm{dom}(\varphi)$. By (5.32), $|t_1 - t_0| = t_1 \leq \Delta_T$ for all $(t_1, j_1) \in \mathrm{dom}(\varphi)$ such that $|j_1 - j_0| \leq m_1$ since $m_1 = \Delta_J$. Thus, $\bar{t}_0 \leq \Delta_T$. For each $i \in \mathcal{I} \setminus \{0\}$, we have

$$\bar{t}_{i-1} = \max\{t \mid (t, m_i) \in \mathrm{dom}(\varphi)\},$$

since t is nondecreasing relative to j for $(t, j) \in \mathrm{dom}(\varphi)$, and $m_i < J$ (from the definition of $\mathcal{I}$), so the supremum in (5.34) is over a compact set and a maximum is attained with $j = m_i$. Thus,

$$\bar{t}_i - \bar{t}_{i-1} \leq \Delta_J \qquad \forall i \in \mathcal{I} \setminus \{0\}. \tag{5.35}$$

Combining $\bar{t}_0 \leq \Delta_T$ and (5.35), we find

$$\bar{t}_i \leq \Delta_T + i\Delta_T \qquad \forall i \in \mathcal{I}. \tag{5.36}$$

For each $i \in \mathcal{I}$ and each $j \in [m_i, m_{i+1}]$, substituting $i\Delta_J$ for m_i leads to $i \leq j/\Delta_J$. Thus, from (5.36), we find

$$\bar{t}_i \leq \Delta_T + \frac{\Delta_T}{\Delta_J} j \qquad \forall i \in \mathcal{I},\ \forall j \in [m_i, m_{i+1}]. \tag{5.37}$$

For each $i \in \mathcal{I}$ and each $(t, j) \in \mathrm{dom}(\varphi)$ such that $j \leq m_{i+1}$, the definition of $\bar{t}_i$ in (5.34) gives us that $t \leq \bar{t}_i$. Since, for every $(t, j) \in \mathrm{dom}(\varphi)$ there exists $i \in \mathcal{I}$ such that $j \in [m_i, m_{i+1}]$, so $t \leq \bar{t}_i \leq \Delta_T + t\Delta_T/\Delta_J$, thereby proving (5.33).

Adding j to both sides of (5.33) produces

$$t + j \leq t + \Delta_J + \frac{\Delta_J}{\Delta_T} t = \left(1 + \frac{\Delta_J}{\Delta_T}\right) t + \Delta_J \quad \forall (t, j) \in \mathrm{dom}(\varphi).$$

Therefore, via algebra,

$$j \geq \left(\frac{\Delta_J}{\Delta_T + \Delta_J}\right)(t + j) - \frac{\Delta_T \Delta_J}{\Delta_T + \Delta_J}.$$

Therefore, (5.18) holds with $N_r := \Delta_T \Delta_J / (\Delta_T + \Delta_J)$ and $\gamma_r : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ defined by

$$s \mapsto \gamma_r(s) := \left(\frac{\Delta_J}{\Delta_T + \Delta_J}\right) s.$$

The function γ_r is linear with a positive coefficient (since $\Delta_T > 0$ and $\Delta_J > 0$), so $\gamma_r \in \mathcal{K}_\infty$. □

5.3.2 Bounded Solutions from Lyapunov Functions

Prior to applying Theorem 5.1 (or other related results, like [3, Thm. 3.19]) to establish uniform global pre-asymptotic stability, it is sometime useful to establish bounds on solutions. In this section, Lemma 5.4 provides sufficient conditions for the sublevel sets of a Lyapunov function candidate V to be forward pre-invariant.

Lemma 5.4. *Let $\mathcal{H} := (C, F, D, G)$ be a hybrid system on $\mathbb{R}^n$ with F and G set valued. Suppose that $\mathcal{H}$ satisfies the hybrid basic conditions (Definition 1.2) and $G(D) \subset C \cup D$. Let $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be a Lyapunov function candidate [3, Def. 3.17] for a closed set $\mathcal{A} \subset \mathbb{R}^n$ with respect to $\mathcal{H}$. Suppose that v_{C} and u_{D} defined in (5.12) and (5.13) satisfy*

$$v_{\mathrm{C}}(x) \leq 0 \quad \text{and} \quad u_{\mathrm{D}}(x) \leq 0 \quad \forall x \in \mathbb{R}^n. \tag{5.38}$$

Then, for each $r \geq 0$, the r-sublevel set of V,

$$L_V(r) := \{x \in \mathrm{dom}(V) \mid V(x) \leq r\}$$

is forward pre-invariant.

Proof. Take any $r \geq 0$. To show that $L_V(r)$ is forward invariant, we will use

$$x \mapsto B_r(x) := V(x) - r \quad \forall x \in \mathbb{R}^n$$

as a barrier function of $L_V(r)$ relative to $\mathcal{H}$ allowing us to apply [5, Thm. 4] to conclude $L_V(r)$ is forward invariant. The function B_r is a barrier function candidate [5, Def. 3] for $L_V(r)$ because $L_V(r) = \{x \in \mathbb{R}^n \mid B_r(x) \leq 0\}$.

We will show [5, Eq. 12] is satisfied. Take any $x \in D \cap L_V(r)$ and any $\eta \in G(x)$. By assumption, $u_{\mathrm{D}}(x) \leq 0$, so $V(\eta) - V(x) \leq 0$. Using the fact that $V(x) \leq r$ (because $x \in L_V(r)$), we find $V(\eta) \leq V(x) \leq r$, so $B_r(\eta) = V(\eta) - r \leq 0$. Therefore,

$$B_r(\eta) \leq 0 \quad \forall x \in D \cap L_V(r),\ \forall \eta \in G(x).$$

Next, we will show [5, Eq. 28] is satisfied. Take any $x \in C$. We want to show

$$\max_{\zeta \in \partial^\circ B(x)} \langle \zeta, \eta \rangle \leq 0 \quad \forall \eta \in F_T(x) := F(x) \cap T_C(x). \tag{5.39}$$

If $F_T(x)$ is empty, then (5.39) is vacuously true. Suppose, instead, that $F_T(x) \neq \varnothing$ and take any $\eta \in F_T(x)$. Since V and B_r differ by a constant, their generalized derivatives are identically equal: $\partial^\circ V \equiv \partial^\circ B_r$. Furthermore, $v_{\mathrm{C}}(x) \leq 0$ by assumption,

since $x \in C$, so

$$\max_{\zeta \in \partial^\circ B_r(x)} \langle \zeta, \eta \rangle = \max_{\zeta \in \partial^\circ V(x)} \langle \zeta, \eta \rangle \leq v_{\mathrm{C}}(x) \leq 0.$$

Therefore, (5.39) and, consequently, [5, Eq. 28] are satisfied.

Equation 13 in [5] is satisfied by assumption. Therefore, by [5, Thm. 4], $L_V(r)$ is pre-forward invariant. □

Although Lemma 5.4 asserts that sublevel sets of a Lyapunov function candidate are forward pre-invariant, under the given assumptions, it can still be the case that solutions are unbounded if $\mathcal{A}$ is unbounded. Under the assumption that $\mathcal{A}$ is compact, the following result establishes bounds on the growth of solutions that start within each fixed distance from $\mathcal{A}$.

Proposition 5.7. *Let $\mathcal{H} := (C, F, D, G)$ be a hybrid system on $\mathbb{R}^n$ with F and G set-valued maps. Suppose that $\mathcal{H}$ satisfies the hybrid basic conditions (Definition 1.2) and $G(D) \subset C \cup D$. Let $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be a Lyapunov function candidate [3, Def. 3.17] for a compact set $\mathcal{A} \subset \mathbb{R}^n$ with respect to $\mathcal{H}$. Suppose that v_{C} and u_{D} defined in* (5.12) *and* (5.13) *satisfy*

$$v_{\mathrm{C}}(x) \leq 0 \quad \text{and} \quad u_{\mathrm{D}}(x) \leq 0 \quad \forall x \in \mathbb{R}^n. \tag{5.40}$$

Then, for all $r \geq 0$, there exists $M_r \geq 0$ such that for every solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \leq r$,

$$|\varphi(t,j)|_{\mathcal{A}} \leq M_r \quad \forall (t,j) \in \mathrm{dom}(\varphi).$$

Proof. Take any $r \geq 0$. Let $\overline{V}_r := \max V(\mathcal{A} + r\mathbb{B})$. The set $\mathcal{A} + r\mathbb{B}$ is compact and V is continuous, so the maximum is well-defined. Let $L_V(\overline{V}_r) := \left\{ x \in \mathbb{R}^n \;\middle|\; V(x) \leq \overline{V}_r \right\}$ be the $\overline{V}_r$-sublevel set of V. For all $x_0 \in \mathcal{A} + r\mathbb{B}$, we have $V(x_0) \leq \overline{V}_r$, so $x_0 \in L_V(\overline{V}_r)$. Therefore,

$$\mathcal{A} + r\mathbb{B} \subset L_V(\overline{V}_r).$$

We will show that $L_V(\overline{V}_r)$ is bounded and forward pre-invariant relative to $\mathcal{H}$, thereby establishing our conclusion.

Let $M_r := \alpha_1^{-1}(\overline{V}_r)$ and take any $x_{\mathrm{L}} \in L_V(\overline{V}_r)$. Thus, $V(x_{\mathrm{L}}) \leq \overline{V}_r$. Furthermore, $\alpha_1(|x_{\mathrm{L}}|_{\mathcal{A}}) \leq V(x_{\mathrm{L}})$, so $\alpha_1(|x_{\mathrm{L}}|_{\mathcal{A}}) \leq \overline{V}_r$, which implies

$$|x_{\mathrm{L}}|_{\mathcal{A}} \leq \alpha_1^{-1}(\overline{V}_r) = M_r.$$

Since this holds for all $x_{\text{L}} \in L_V(\overline{V}_r)$,

$$L_V(\overline{V}_r) \subset \left\{x : |x|_{\mathcal{A}} \leq M_r\right\} = \mathcal{A} + M_r\mathbb{B}.$$

This implies $L_V(\overline{V}_r)$ is bounded because $\mathcal{A}$ is compact. The set $L_V(\overline{V}_r)$ is forward invariant by Lemma 5.4.

Let φ be any solution to $\mathcal{H}$ with $|\varphi|_{\mathcal{A}} \leq r$. We have that $V(\varphi(0,0)) \leq \overline{V}_r$, so $\varphi(0,0) \in L_V(\overline{V}_r)$. By the pre-forward invariance of $L_V(\overline{V}_r)$, the solution $\varphi(t,j)$ is in $L_V(\overline{V}_r)$ for all $(t,j) \in \text{dom}(\varphi)$. Since $L_V(\overline{V}_r) \subset \mathcal{A} + M_r\mathbb{B}$,

$$|\varphi(t,j)|_{\mathcal{A}} \leq M_r \qquad \forall (t,j) \in \text{dom}(\varphi). \qquad \square$$

5.3.3 Continuous-Time and Discrete-Time Systems

The following corollaries are special cases of Theorem 5.1 for continuous- and discrete-time systems.

Corollary 5.1 (Continuous-time Lyapunov Theorem)**.** *Consider compact set $\mathcal{A} \subset \mathbb{R}^n$, a continuous-time system on $C \subset \mathbb{R}^n$*

$$\dot{x} \in F(x) \quad x \in C, \tag{5.41}$$

and a Lyapunov function candidate V with respect to $\mathcal{A}$ for (5.41)*. Suppose there exists $\alpha_1 \in \mathcal{K}_\infty$ and an LSC function $\sigma_c \in \mathcal{PD}(\mathcal{A})$ such that*

$$\alpha_1(|x|_{\mathcal{A}}) \leq V(x) \quad \text{and} \quad u_{\text{C}}(x) \leq -\sigma_c(x) \quad \forall x \in C.$$

Then, $\mathcal{A}$ is UGpAS for (5.41)*.*

Corollary 5.2 (Discrete-time Lyapunov Theorem)**.** *Consider compact set $\mathcal{A} \subset \mathbb{R}^n$, a discrete-time system on $D \subset \mathbb{R}^n$*

$$x^+ \in G(x) \quad x \in D, \tag{5.42}$$

and Lyapunov function candidate V with respect to $\mathcal{A}$ for (5.42)*. Suppose there exists $\alpha_1 \in \mathcal{K}_\infty$ and an LSC function $\sigma_d \in \mathcal{PD}(\mathcal{A})$ such that*

$$\begin{aligned} \alpha_1(|x|_{\mathcal{A}}) &\leq V(x) && \forall x \in D \cup G(D) \\ u_{\text{D}}(x) &\leq -\sigma_d(x) && \forall x \in D. \end{aligned}$$

Then, $\mathcal{A}$ is UGpAS for (5.42)*.*

The next example illustrates how Theorem 5.1 can be used to show that a compact set is UGpAS for $\dot{x} = F(x)$ with F discontinuous.

Example 5.4. Consider the continuous-time system

$$\dot{x} = F(x) := -\lfloor x \rfloor \quad x \in C := \mathbb{R},$$

where $\lfloor x \rfloor$ is the largest integer m such that $m \leq x$ and $\lceil x \rceil$ is the smallest integer n such that $n \geq x$. The $\lfloor \cdot \rfloor$ function is USC and $\lceil \cdot \rceil$ is LSC. Let $\mathcal{A} := [0, 1]$ and consider $x \mapsto V(x) := |x|^2_{\mathcal{A}}$. We find that

$$u_{\mathrm{C}}(x) = \begin{cases} (-|x|_{\mathcal{A}})\lfloor x \rfloor & \text{if } x \geq 0 \\ (|x|_{\mathcal{A}})\lfloor x \rfloor & \text{if } x < 0, \end{cases}$$

which is neither LSC nor USC. Let

$$-\sigma_c(x) := \begin{cases} |x|_{\mathcal{A}}(1 - \lceil x \rceil) & \text{if } x \geq 0 \\ (|x|_{\mathcal{A}})\lfloor x \rfloor & \text{if } x < 0. \end{cases}$$

We see $-\sigma_c$ is USC, so σ_c is LSC. For $x \leq 0$, $u_{\mathrm{C}}(x) = -\sigma_c(x)$, and for $x \geq 0$, $-\lfloor x \rfloor \geq 1 - \lceil x \rceil$, so $u_{\mathrm{C}}(x) \leq -\sigma_c(x)$, thus (5.15) holds. It can be easily checked that $\sigma_c \in \mathcal{PD}(\mathcal{A})$. Therefore, $\mathcal{A}$ is UGpAS for $\dot{x} = F(x)$, by Corollary 5.1. $\diamond$

Chapter 6

Conical Transition Graph (CTG)

In Section 4.2 and Chapter 3 we considered methods for using a Lyapunov-certified controller as a backup controller, allowing one to safely deploy an uncertified controller. The advantage of this approach is that it allows one to construct a Lyapunov function once, perhaps for a simple, well understood controller and reuse to provide guaranteed properties for other controllers. The downside of this approach is that it still requires constructing one Lyapunov function, which can be difficult even for simple systems. In fact, one is often unsure whether the construction of a Lyapunov function is a futile endeavor, since it may be unclear whether the system is stable in the first place. In this chapter, we introduce a method for algorithmically checking stability and instability in a class of hybrid systems we call *conical hybrid systems* [23].

6.1 Introduction

For continuous- and discrete-time systems, local asymptotic stability can be determined by linearizing the system and checking the eigenvalues of the resulting Jacobian matrix. For hybrid systems, however, the same ease is currently unavailable. In the conical approximation of a hybrid system, the flow and jumps sets are approximated by tangent cones, and the flow and jump maps are approximated by constant or linear approximations [4, Ch. 9]. It was shown in [24, Thm. 3.3] that the conical approximation of a hybrid system can be used to determine if a point is pre-asymptotically stable. Namely, if a point is pre-asymptotically stable with respect to the conical approximation, then the center of the approximation in the original hybrid system is locally pre-asymptotically stable. (The prefix "pre-" indi-

cates that some maximal solutions may terminate in finite time due to the solution leaving the region of the state space where it is permitted to evolve.) The utility of [24, Thm. 3.3] is currently limited, however, by the fact that it is still generally difficult to show that the origin of a conical approximation is pre-asymptotically stable. The purpose of this chapter is to close this gap by introducing the *conical transition graph* (CTG) as a tool to determine asymptotic stability in conical approximations. Thereby, we can establish local asymptotic stability in non-conical hybrid systems.

A graph-based approach is used in [25] to determine Lyapunov and asymptotic stability of a class of hybrid systems called piecewise constant derivatives (PCD). In a PCD system, the state space is partitioned into polyhedral regions with a flow vector field that is constant within each region but not necessarily continuous on their boundaries. The class of systems considered in the present work is more general in that the hybrid systems permit jumps in the value of the state and transitions between modes.

While there are limited results for analyzing stability of hybrid systems via conical approximations, there are numerous other approaches for stability analysis in the literature [26, 27] and [4, Thm. 7.30]. Lyapunov functions are a powerful and flexible tool for proving many types of stability properties, including stability of sets, finite-time stability, Zeno stability, and input to state stability [15, 28]. For hybrid systems where asymptotic stability of a limit cycle is of interest rather asymptotic stability of an equilibrium point, Poincaré maps have been used in hybrid systems to prove convergence of solutions to limit cycles [29, 30, 31]. Discrete graphs[1] have been used to evaluate stability of switched dynamical systems including discrete-time linear systems [32], discrete-time nonlinear systems [33], and continuous-time linear systems [34]. In contrast to the existing methods for switched systems, the present work is (to the best of our knowledge) the first graph-theoretic approach to analyze asymptotic stability in non-switched hybrid systems (i.e., systems where components of the state vector may range over a continuum at jumps). In the context of reachability analysis, [35] introduced *conical abstractions* as a graph-based method to compute infinite-horizon reachable sets for linear hybrid automata. The biggest drawback of the Lyapunov function method is that Lyapunov functions

[1]Throughout, we use *graph* in the sense of *discrete graph*—that is, a set of vertices connected by edges or arrows.

are often difficult to construct. There have, however, been advances made for algorithmically constructing Lyapunov functions. For hybrid systems defined by polynomial functions, Lyapunov functions can be constructed numerically via sum-of-squares (SOS) programming [7, 36, 37, 38, 39, 40]. Lyapunov functions can also be generated for non-polynomial systems by modeling non-polynomial functions as polynomials plus a disturbance, as done in [7] for barrier certificates, or by transforming the system into a polynomial system as done in [38] for continuous-time systems. The SOS approach to constructing Lyapunov functions is powerful but suffers from two limitations. Firstly, SOS requires solving a semidefinite program (SDP) that grows quickly as the dimension of the hybrid system and the degrees of the polynomials increase. While there are efficient algorithms for solving SDP's, the size of the optimization problem can make them computationally expensive for high-dimensional hybrid systems. Secondly, since SOS is a numerical approach, it requires the hybrid system to be fully defined, numerically—it cannot have any unspecified parameters. This inhibits using SOS to reason about parameters, limiting its utility for, e.g., designing an asymptotically stabilizing feedback law.

An alternative algorithmic approach to determine stability-like properties is via reachability analysis. The idea behind this approach is to use numerical reachability tools for hybrid systems [35, 41, 42, 43] to approximate the reachable set for solutions starting nearby an equilibrium and thereby assess stability numerically.

The conical transition graph is designed to simplify the analysis of asymptotic stability of isolated equilibria by creating a simplified representation of ways that solutions to a hybrid system can evolve continuously (called *flows*) or evolve discretely (called *jumps*). Collectively, we refer to flows and jumps as *transitions*. In particular, the CTG is a directed graph with set-valued weights assigned to each arrow. Each vertex in the CTG represents either the origin $0_n \in \mathbb{R}^n$ or a point in the unit sphere $\mathbb{S}^{n-1} \subset \mathbb{R}^n$, where each point $v \in \mathbb{S}^{n-1}$ acts as a representation of all the points in the ray $\{rv | r > 0\}$ spanned by v. In this way, we consider the projection of $\mathbb{R}^n$ onto $\mathbb{S}^{n-1} \cup \{0_n\}$, as shown in Figure 6.1. Roughly speaking, each arrow in the CTG represents the ways that solutions to a hybrid system, as projected onto $\mathbb{S}^{n-1} \cup \{0_n\}$, can transition (flow or jump) between points in $\mathbb{S}^{n-1} \cup \{0_n\}$. The weight of each arrow contains all possible relative changes in magnitude that a solution can exhibit when it undergoes the transition. Asymptotic stability can be determined

from the products of walks through the CTG. Products converging to zero indicate convergence of solutions to the origin.

This chapter extends the author's previous work, [23], in two ways. First, this chapter defines and analyzes conical hybrid systems with modes—allowing switching between several regimes. To aid in analysis, we introduce in this chapter the concept of a CTG-simulation of a solution to a hybrid system. By showing a correspondence between solutions and CTG-simulations, we show that the CTG of a hybrid system can be used to determine asymptotic stability. Beyond the results in this chapter, CTG-simulations may be a useful theoretic tool in future work for using CTG's in reachability analysis.

Second, we describe how to reduce the size of a conical transition graph by an "abstraction" that groups together sets of vertices. By applying this method to conical transition graphs with large—possibly infinite—numbers of vertices, we can reduce intractable computational problems into problems that are solvable.

The remainder of this chapter is organized as follows. Preliminary concepts and notation are introduced in Section 6.2. In Section 6.2.1 we introduce conical hybrid systems with modes, and in Section 6.2.2 we describe the important radial homogeneity property of conical hybrid systems. We briefly describe two applications of conical hybrid systems in Section 6.3. Conical transition graphs are introduced in Section 6.4. Our results, in Section 6.5, demonstrate how to use a conical transition graph to determine pre-asymptotic stability in conical hybrid systems. Section 6.5.1 describes CTG-simulations, which is a useful tool in the subsequent theoretical developments. Our stability and pre-asymptotic stability results are found in Section 6.5.2. Section 6.6 describes our approach to reducing the size of CTG's by creating an "abstract" CTG that groups together vertices.

6.2 Preliminaries

The unit sphere in $\mathbb{R}^n$ is denoted by $\mathbb{S}^{n-1} := \left\{x \in \mathbb{R}^n : |x| = 1\right\}$, and the unit sphere plus the origin is written as

$$\mathbb{S}_0^{n-1} := \mathbb{S}^{n-1} \cup \{0_n\}. \tag{6.1}$$

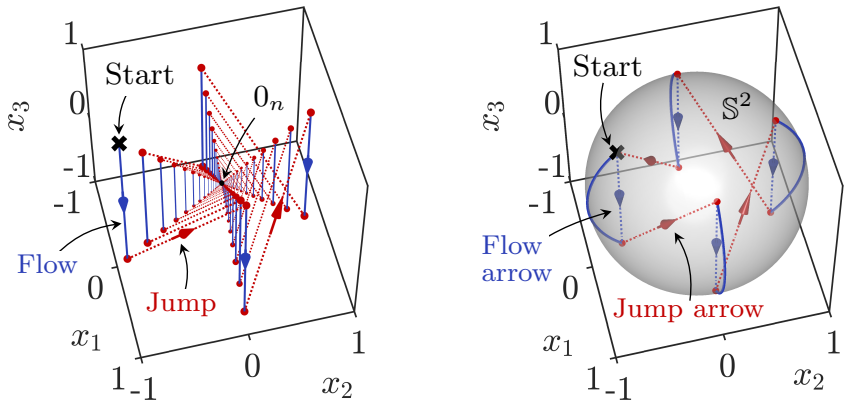

Figure 6.1. The evolution of solutions to a hybrid system on $\mathbb{R}^3$ (left) are reduced in the CTG (right) to discrete transitions on $\mathbb{S}^2$, which we label as *flow arrows* and *jump arrows*. In the right image, solid blue curves indicate continuous-time flows projected onto $\mathbb{S}^2$.

The *normalized radial vector* function $\operatorname{nrv} : \mathbb{R}^n \to \mathbb{S}_0^{n-1}$ is defined for each $v \in \mathbb{R}^n$ as

$$\operatorname{nrv}(v) := \begin{cases} v/|v| & \text{if } v \neq 0_n \\ 0_n & \text{if } v = 0_n. \end{cases} \tag{6.2}$$

The following properties of the nrv function are used in this work.

$$\forall x \in \mathbb{R}^n : \quad x = |x| \operatorname{nrv}(x). \tag{6.3}$$

$$\forall x \in \mathbb{R}^n \text{ and } r > 0 : \quad \operatorname{nrv}(rx) = \operatorname{nrv}(x). \tag{6.4}$$

$$\forall x \in \mathbb{R}^n \text{ and } A \in \mathbb{R}^{n \times n} : \quad \operatorname{nrv}(Ax) = \operatorname{nrv}(\operatorname{nrv}(Ax)) = \operatorname{nrv}(A \operatorname{nrv}(x)). \tag{6.5}$$

Let $S \subset \mathbb{R}^n$ be nonempty and let $x \in \overline{S}$. The *contingent cone* $T_S(x)$ is the set of all vectors $v \in \mathbb{R}^n$ such that there exist a sequence of positive real numbers $h_i \to 0^+$ and a sequence of vectors $v_i \to v$ such that $x + h_i v_i \in S$ for all $i \in \mathbb{N}$ (see [1]). For any $S \subset \mathbb{R}^n$ and $x \in \overline{S}$, the contingent cone of S at x is a *cone*, meaning that for all $x \in T_S(x)$ and all $\alpha > 0$, we have that $\alpha x \in T_S(x)$.

For any $x \in \mathbb{R}^n$, we write the open ray from the origin through x as

$$\operatorname{ray}(x) := \{\alpha x \in \mathbb{R}^n \mid \alpha > 0\}$$

and the corresponding closed ray as

$$\overline{\operatorname{ray}}(x) := \{\alpha x \in \mathbb{R}^n \mid \alpha \geq 0\}.$$

Given a cone $K \subset \mathbb{R}$ and any $x \in \mathbb{R}^n$,

$$x \in K \iff \operatorname{ray} x \subset K.$$

We write the *conical hull* of $x_1, x_2, \ldots, x_p \in \mathbb{R}^n$ as

$$\operatorname{cone}(x_1, x_2, \ldots, x_p) = \{\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_p x_p \mid \alpha_i \geq 0\}.$$

Given a set $\mathcal{S} \subset \mathbb{R}^n$ and linear map $A \in \mathbb{R}^{n \times n}$, then the transformation of $\mathcal{S}$ by A is defined as

$$A\mathcal{S} := \{Ax \in \mathbb{R}^n \mid x \in \mathcal{S}\}.$$

6.2.1 Conical Hybrid Systems

Definition 6.1 (Conical Hybrid System with Modes)**.** Let $\mathcal{Q} := \{1, 2, \ldots, N_{\mathrm{Q}}\}$ be a finite set of modes, let $\mathcal{E} \subset \mathcal{Q} \times \mathcal{Q}$ be directed edges (transitions) between modes. Consider a hybrid system $\mathcal{H}$ with state $x := (q, z) \in \mathcal{Q} \times \mathbb{R}^n$ in the form

$$\left\{\begin{array}{ll} \dot{x} = f(q,z) := \begin{bmatrix} 0 \\ f_q(z) \end{bmatrix} & x \in C := \left\{ \begin{bmatrix} q \\ z \end{bmatrix} \in \mathcal{Q} \times \mathbb{R}^n \,\middle|\, z \in C_q \right\} \\ x^+ \in G(q,z) := \left\{ \begin{bmatrix} q' \\ A_e z \end{bmatrix} \,\middle|\, \begin{array}{l} \exists e := (q,q') \in \mathcal{E} \\ \text{s.t. } z \in D_e \end{array} \right\} & x \in D := \left\{ \begin{bmatrix} q \\ z \end{bmatrix} \in \mathcal{Q} \times \mathbb{R}^n \,\middle|\, \begin{array}{l} \exists q' \in \mathcal{Q} \\ \text{s.t. } z \in D_{(q,q')} \end{array} \right\}, \end{array}\right. \tag{6.6}$$

where for each mode $q \in \mathcal{Q}$ and edge $e := (q, q') \in \mathcal{E}$, the function $z \mapsto f_q(z)$ is linear or constant, $A_e \in \mathbb{R}^{n \times n}$, the set $C_q \subset \mathbb{R}^n$ is a closed cone that defines the region where z is allowed to flow while in mode q, and for each $q' \in \mathcal{Q}$, the set $D_e \subset \mathbb{R}^n$ is a closed cone that defines the region where z is allowed to jump from mode q to mode q'. If $(q, q') \notin \mathcal{E}$, then $D_e = \varnothing$.

Since q does not depend on t, we write it as a function of j only when it occurs as a component of hybrid arcs, that is, $j \mapsto q(j)$. When mode q has linear flows, we write $\dot{z} = A_q z$, where $A_q \in \mathbb{R}^{n \times n}$, whereas when mode q has constant flows we write $\dot{z} = f_q^*$. ◇

A diagram of a conical hybrid system with two modes is shown in Figure 6.2.

Example 6.1 (Conical Hybrid System with Modes)**.** As an example of a hybrid system with modes, we consider a conical hybrid system $\mathcal{H}$ in $\mathbb{R}^2$ with two modes,

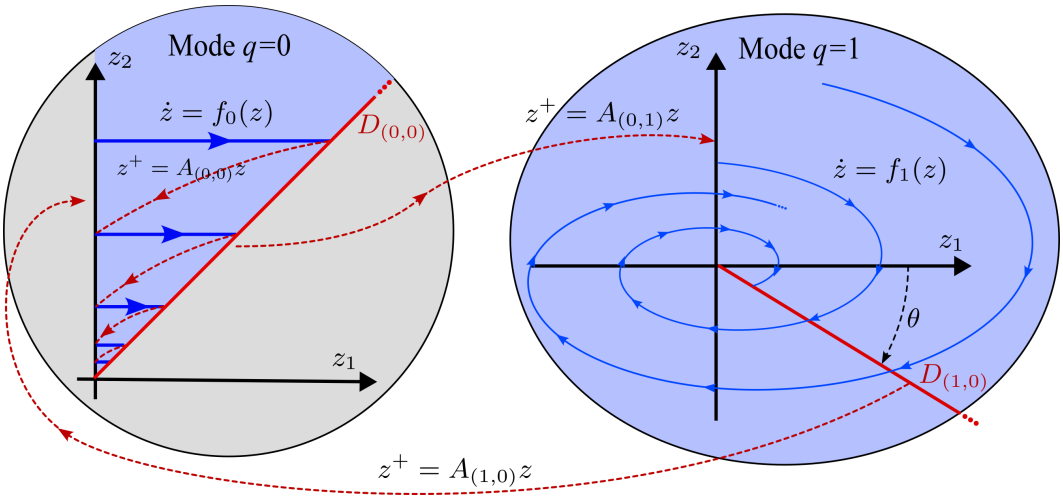

Figure 6.2. A conical hybrid system with two modes. Mode $q = 0$ (left) has constant flows and mode $q = 1$ (right) has linear flows.

$\mathcal{Q} := \{0, 1\}$, where mode $q = 0$ has constant flows and mode $q = 1$ has linear flow modes. For mode 0, let flows be defined by $\dot{z} = f_0^* := \left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right]$ and

$$C_0 := \left\{(z_1, z_2) \in \mathbb{R}^2 \;\middle|\; z_2 \geq z_1,\ z_1 \geq 0\right\},$$

and let

$$D_{(0,0)} := D_{(0,1)} := \{(z_1, z_2) \mid z_1 = z_2,\ z_1 \geq 0\}.$$

After a jump from $q = 0$ to $q = 0$, the value of z is given by $z^+ = A_{(0,0)}z$, and after a jump from $q = 0$ to $q = 1$, it is given by $z^+ = A_{(0,1)}$, where

$$A_{(0,0)} := \begin{bmatrix} 0 & 0 \\ \lambda_0 & 0 \end{bmatrix} \quad \text{and} \quad A_{(0,1)} := \begin{bmatrix} 0 & 0 \\ \lambda_1 & 0 \end{bmatrix},$$

with $\lambda_0 > 0$, $\lambda_1 > 0$. Thus, at jumps, z is mapped to the z_2-axis.

For mode $q = 1$, let $\dot{z} = A_1 z$ where

$$A_1 := \begin{bmatrix} -2 & 4 \\ -2 & -1 \end{bmatrix},$$

and $C_1 := \mathbb{R}^2$. The jump set is defined as the ray from the origin with an angle $\theta(-\pi/2, \pi/2)$ from the z_1-axis, i.e., $D_{(1,0)} := \overline{\text{ray}}\left[\begin{smallmatrix} \cos\theta \\ \sin\theta \end{smallmatrix}\right]$. The jump map from $q = 1$ to $q = 0$ is defined by

$$A_{(1,0)} = \begin{bmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{bmatrix},$$

which takes any vector $z \in D_{(1,0)}$ to $A_{(1,0)}z \in \{0\}\times\mathbb{R}_{\geq 0}$ (the z_2-axis). The transitions between modes are $\mathcal{E} = \{(0,0),(0,1),(1,0)\}$. Based on the choices of parameters λ_0, λ_1, and θ, the set $\mathcal{O} := \mathcal{Q} \times \{0_n\}$ will be asymptotic stable or unstable. The techniques introduced in this chapter reduces the problem of checking stability into analyzing a discrete graph. $\diamond$

6.2.2 Properties of Conical Hybrid Systems

An important property of conical hybrid systems, formalized in Proposition 6.1, below, is that their dynamics are radially homogenous—that is, a conical hybrid system behaves the same way at all distances from the origin, except for scaling effects.

Proposition 6.1. *Given a conical hybrid system with modes $\mathcal{H}$, let*

$$(t,j) \mapsto \varphi(t,j) := \big(q(t,j), z(t,j)\big)$$

be a solution to $\mathcal{H}$. Then, for each $r > 0$, the hybrid arc $(t,j) \mapsto \psi_r(t,j)$ defined by

$$\psi_r(t,j) := \begin{bmatrix} q(j) \\ rz(\alpha_r(t),j) \end{bmatrix} \quad \forall (\alpha_r(t),j) \in \mathrm{dom}(\varphi) \tag{6.7}$$

is also a solution to $\mathcal{H}$, where α_r is a class-$\mathcal{K}$ function defined, for all $(t,j) \in \mathrm{dom}(\varphi)$, by

$$\alpha_r(t) = \int_0^t \delta_r(\bar{\jmath}(\tau))\, d\tau, \tag{6.8}$$

and

$$\delta_r(j) := \begin{cases} 1/r & \text{if } q(j) \text{ is a mode with constant flow} \\ 1 & \text{if } q(j) \text{ is a mode with linear flow.} \end{cases} \tag{6.9}$$

The effect of δ_r in (6.8) is that in modes with linear flow, the time ψ_r spends traversing an interval of flow matches φ, but in modes with constant flow, the time is dilated by a factor r.

Proof. First, we show that α_r is class-$\mathcal{K}$. From the definition, $\alpha_r(0) = 0$ and α_r is continuous. Since δ_r is strictly positive, α_r is monotonically increasing, so α_r is class-$\mathcal{K}$.

Let $J := \sup_j \mathrm{dom}(\varphi)$, and let t_1, t_2, …, t_J be the jump times of φ. For ease of notation, let $t_0 := 0$ and, if J is finite, let $t_{J+1} := \sup_t \mathrm{dom}(\varphi)$. For each jump time

t_j in $\mathrm{dom}(\varphi)$, the hybrid times $(t_j, j-1)$ and (t_j, j) are in $\mathrm{dom}(\varphi)$, so $(\alpha_r^{-1}(t_j), j-1)$ and $(\alpha_r^{-1}(t_j), j)$ are in $\mathrm{dom}(\psi_r)$ (α_r is invertible because it is class-$\mathcal{K}$). Therefore, $t'_j := \alpha_r^{-1}(t_j)$ is a jump time in $\mathrm{dom}(\psi_r)$ for each $j \in \{1, 2, \ldots, J\}$.

Since α_r (and α_r^{-1}) is strictly increasing, $[t_j, t_{j+1}]$ is an interval of flow in $\mathrm{dom}(\varphi)$ if and only if $[t'_j, t'_{j+1}]$ is an interval of flow in $\mathrm{dom}(\psi_r)$. For each $(t, j) \in \mathrm{dom}(\psi_r)$, let $(t, j) \mapsto z_r(t, j) := rz(\alpha_r(t), j)$, so that $\psi_r(t, j) = (q(j), z_r(t, j))$. Since $z(t_j, j-1) \in D_{(q(j-1),q(j))}$ and $D_{(q(j-1),q(j))}$ is a cone, we have that

$$z_r(t'_j,\, j-1) = rz\Big(\alpha_r\big(\alpha_r^{-1}(t_j)\big),\, j-1\Big) = rz(t_j,\, j-1) \in D_{(q(j-1),q(j))}.$$

Therefore, $\psi_r(t'_j, j-1) \in D$, so ψ_r satisfies (1.5a). Similarly, since $C_{q(j)}$ is a cone and $z(t, j) \in C_q$ for all $t \in (t_j, t_{j+1})$, we have that $z_r(t, j) \in C_q$ and thus $\psi_r(t, j) \in C$ for all $t \in (t'_j, t'_{j+1})$. Therefore, ψ_r satisfies (1.6a).

Now that we have established the jump times and intervals of flows of ψ_r, we want to show that ψ_r satisfies the jump and flow conditions in (1.5) and (1.6). Take any $j \in \{1, 2, \ldots, J\}$. By (1.5b),

$$\varphi(t_j, j) = \begin{bmatrix} q(j) \\ z(t_j, j) \end{bmatrix} \in G(\varphi(t_j, j-1)),$$

so, from the definition of G in (6.6), $z(t_j, j) = A_{(q(j-1),q(j))} z(t_j, j-1)$. Thus, at $t'_j := \alpha_r^{-1}(t_j)$,

$$\psi_r(t'_j, j) = \begin{bmatrix} q(j) \\ rz\big(\alpha_r(\alpha_r^{-1}(t_j)), j\big) \end{bmatrix} = \begin{bmatrix} q(j) \\ A_{(q(j-1),q(j))}\big(rz(t_j, j-1)\big) \end{bmatrix}.$$

Since $D_{(q(j-1),q(j))}$ is a cone and $z(t_j, j-1)$ is in $D_{(q(j-1),q(j))}$, then $rz(t_j, j-1)$ is also in $D_{(q(j-1),q(j))}$. Therefore, $\psi_r(t_j, j)$ is in the set $G\big(\psi_r(t'_j, j-1)\big)$ as required by (1.5b).

If $t_{j+1} > t_j$, then $[t_j, t_{j+1}]$ is an interval of flow for φ, so for all $t \in (t_j, t_{j+1})$,

$$\dot{z}(t, j) = f_q(z(t, j)).$$

(Since f_q is linear or constant, we have that if $\dot{z} = f_q(z)$ for almost all $t \in (t_j, t_{j+1})$ then it, in fact, satisfies the ODE for all $t \in (t_j, t_{j+1})$). From Definition 6.1, the mode $q(j)$ has either linear or constant flows. Suppose, first, that $q(j)$ has linear flows. Then, for all $t \in (t_j, t_{j+1})$,

$$\frac{d}{dt}(z(t, j)) = A_{q(j)} z(t, j).$$

Applying the chain rule to $t \mapsto z_r(t,j) = rz(\alpha_r(t),j)$, we find

$$\dot{z}_r(t,j) = \frac{d}{dt}(rz(\alpha_r(t),j)) = A_{q(j)} rz(\alpha_r(t),j) \frac{d\alpha_r}{dt}(t) = A_{q(j)} z_r(t,j),$$

since $d\alpha_r/dt\,(t) = \delta_r(q(j)) = 1$ by applying the fundamental theorem of calculus to (6.9). Thus, ψ_r satisfies (1.6b) in the case of linear flows.

Suppose instead $q(j)$ has constant flows. Then, for all $t \in (t_j, t_{j+1})$,

$$\frac{d}{dt}(z(t_j, t_{j+1})) = f_{q(j)}.$$

Applying the chain rule to $t \mapsto z(\alpha_r(t),j)$, we find

$$\dot{z}_r(t,j) = \frac{d}{dt}(rz(\alpha_r(t),j)) = r f_{q(j)} \frac{d\alpha_r}{dt}(t) = \frac{r f_{q(j)}}{r} = f_{q(j)},$$

since $d\alpha_r/dt\,(t) = \delta_r(q(j)) = 1/r$. Thus, ψ_r satisfies (1.6b) in the case of linear flows. Therefore, ψ_r is a solution to $\mathcal{H}$. □

6.3 Applications of Conical Hybrid Systems

In this section, we introduce one application of conical hybrid systems.

6.3.1 Sampled Linear Systems

Example 6.2 (Linear System with Sampled Control)**.** Conical hybrid systems can be used to model and analyze linear control systems with sampled control updates. Consider the linear control system

$$\dot{z} = Az + Bu,$$

with state $z \in \mathbb{R}^n$ and input $u \in \mathbb{R}^m$. Suppose u is updated with period T according to $u := Kz$, where $K \in \mathbb{R}^{m \times n}$. One way to model such as system as a hybrid system is to use a timer variable $\tau \in [0,T]$ where $\dot{\tau} = 1$ and triggering events to update the input when $\tau = T$, and resetting $\tau^+ = 0$. Such an approach results in a non-conical hybrid system, because the set of τ-values where jumps are triggered is non-conical. As an alternative, we propose using a timer variable $\tau := (\tau_1, \tau_2) \in \mathbb{R}^2$ where τ evolves according to

$$\dot{\tau} = M\tau, \quad \text{with } M := \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix},$$

where $\omega := \pi/T$. When τ starts with $\tau \in (0,\infty) \times \{0\}$, it takes time T for τ to reach $(-\infty, 0) \times \{0\}$. Thus, to achieve periodic sampling, we will update $u^+ = Kx$ and $\tau^+ = -\frac{1}{2}\tau$ whenever $\tau \in (-\infty, 0) \times \{0\}$. (The $\frac{1}{2}$ causes τ to converge to 0_n, which is convenient for showing the origin is asymptotically stable.) A representative trajectory for τ is shown in Figure 6.3.

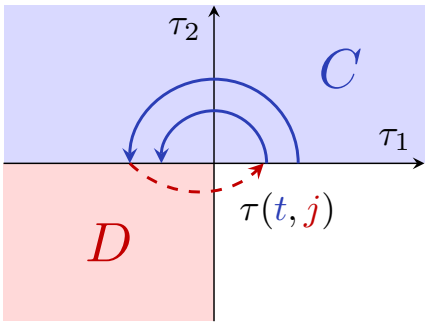

Figure 6.3. An example trajectory for the timer variable τ in Example 6.2.

The closed-loop system has state

$$x := (z, u, \tau) \in \mathcal{X} := \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^2,$$

and can be written as a conical hybrid system (without modes):

$$\begin{cases} \begin{bmatrix} \dot{z} \\ \dot{u} \\ \dot{\tau} \end{bmatrix} = \begin{bmatrix} Az + Bu \\ 0 \\ M\tau \end{bmatrix} & x \in C := \{(z, u, \tau_1, \tau_2) \in \mathcal{X} \mid \tau_2 \geq 0,\ \tau \neq (0,0)\} \\ \begin{bmatrix} z^+ \\ u^+ \\ \tau^+ \end{bmatrix} = \begin{bmatrix} z \\ Ku \\ -\frac{1}{2}\tau \end{bmatrix} & x \in D := \{(z, u, \tau_1, \tau_2) \in \mathcal{X} \mid \tau_1 < 0,\ \tau_2 = 0\}. \end{cases}$$

Various adjustments to this example could allow for modeling systems that have nondeterministic delays between samples and switching between modes. ◇

6.3.2 Conical Approximations

One application of conical hybrid systems are as approximations of non-conical hybrid systems. Such approximations are called *conical approximations*. The following assumption is necessary for the conical approximation of a hybrid system $\mathcal{H}$ to be well-defined at a point $x_* \in \mathbb{R}^n$.

Assumption 6.1. For a given hybrid system $\mathcal{H} := (C, f, D, g)$ and $x_* \in \mathbb{R}^n$, suppose that the following conditions hold:

1. If $x_* \in \overline{D}$, then $g(x_*) = x_*$ and g is continuously differentiable at x_*.
2. If $x_* \in \overline{C}$, then f is continuous at x_*.
3. If $x_* \in \overline{C}$ and $f(x_*) = 0_n$, then f is continuously differentiable at x_*. ◇

Definition 6.2 ([4])**.** Given a hybrid system $\mathcal{H} = (C, f, D, g)$ and a point $x_* \in \mathbb{R}^n$ that satisfy Assumption 6.1, the *conical approximation* of $\mathcal{H}$ at x_* is

$$\check{\mathcal{H}} : \begin{cases} \dot{x} \quad = \check{f}(x) := \begin{cases} f(x_*), & \text{if } f(x_*) \neq 0 \\ A_{\mathrm{C}}(x - x_*), & \text{if } f(x_*) = 0, \end{cases} & x \in \check{C} := T_C(x_*), \\ x^+ = \check{g}(x) := A_{\mathrm{D}}(x - x_*), & & x \in \check{D} := T_D(x_*), \end{cases} \tag{6.10}$$

where A_{C} and A_{D} denote the Jacobian matrices of g and f at x_*, respectively:

$$A_{\mathrm{C}} := \frac{\partial f}{\partial x}(x_*) \quad \text{and} \quad A_{\mathrm{D}} := \frac{\partial g}{\partial x}(x_*). \qquad \diamond$$

The following result establishes local pre-asymptotic stability in a hybrid system via pre-asymptotic stability in its conical approximation.

Theorem 6.1 ([24], Thm. 3.3)**.** *Suppose a hybrid system $\mathcal{H}$ and a point $x_* \in \mathbb{R}^n$ satisfy Assumption 6.1. Let $\check{\mathcal{H}}$ be the conical approximation of $\mathcal{H}$ at x_*. If 0_n is pAS for $\check{\mathcal{H}}$, then x_* is locally pAS for $\mathcal{H}$.*

6.4 Conical Transition Graph

This work relies on definitions from graph theory, provided in this section. See [44] for details.

Directed Graph A *directed graph* $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ consists of a set of *vertices* $\mathcal{V}$ and a set of *arrows* $\mathcal{A}$. Each arrow in $\mathcal{G}$ starts at some vertex $v_1 \in \mathcal{V}$ and ends at some vertex $v_2 \in \mathcal{V}$. We write an arrow from v_1 to v_2 as $v_1 \to v_2$. In a directed graph, an arrow can have the same start and end point ($v_1 = v_2$), in which case it is called a *loop*.

We also allow for multiple arrows that have the same start and end points. To distinguish between such arrows, we assign each arrow a *label*. An arrow with the label "L" is written as $\mathfrak{a}^{\text{L}} = v_1 \xrightarrow{\text{L}} v_2$. In this work, we use only two labels: "F" and "J," which stand for "flow" and "jump." Thus, for $v_1, v_2 \in \mathcal{V}$, there can be at most two distinct arrows $v_1 \xrightarrow{\text{F}} v_2$ and $v_1 \xrightarrow{\text{J}} v_2$. If the label is irrelevant for a particular point of discussion, then it can be omitted.

A *weighted* directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{W})$ is a directed graph $(\mathcal{V}, \mathcal{A})$ that also includes a weight function $\mathcal{W}$ that defines a weight for each arrow in $\mathcal{A}$. In a typical weighted graph, the weight function assigns a real number to each arrow, but in this work we use *set-valued* weights. Thus, the *weight function* is a set-valued map $\mathcal{W} : \mathcal{A} \rightrightarrows \mathbb{R}$ that maps each arrow $\mathfrak{a}$ in $\mathcal{A}$ to a set of real numbers $\mathcal{W}(\mathfrak{a}) \subset \mathbb{R}$.

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{W})$, a *walk* w through $\mathcal{G}$ is a finite or infinite sequence of arrows in $\mathcal{A}$. A walk of length $K \in \{1, 2, \dots\} \cup \{\infty\}$ is written

$$w = (\mathfrak{a}_0,\ \mathfrak{a}_1,\ \dots,\ \mathfrak{a}_{K-1}) = v_0 \to v_1 \to v_2 \to \cdots \to v_K,$$

such that $\mathfrak{a}_k = v_k \to v_{k+1}$ for each $k = 0, 1, \dots, K-1$.

We define the weight of a walk w as the cumulative *Minkowski set product* of the arrows in w. For any sets $A, B \subset \mathbb{R}$, the Minkowski set product of A and B is defined in [45] as $AB := \{ab \mid a \in A, b \in B\}$. For a finite-length walk $w = (\mathfrak{a}_0, \mathfrak{a}_2, \dots, \mathfrak{a}_{N-1})$, the *set-valued weight* of w is

$$\mathcal{W}(w) := \left\{ \prod_{k=0}^{N-1} r_k \;\middle|\; \begin{array}{l} r_k \in \mathcal{W}(\mathfrak{a}_k) \\ \quad \forall k = 0, 1, \dots, N-1 \end{array} \right\}. \tag{6.11}$$

If we let $K = \infty$, then $\mathcal{W}(w)$ may not be well-defined because the infinite product $\prod_{k=0}^{\infty} r_k$ in (6.11) may not converge. For this chapter, however, it is sufficient to define $\mathcal{W}(w)$ if and only if $\prod_{k=0}^{\infty} r_k$ converges to 0 for every choice of $\{r_k\}$. For an infinite-length walk $w := (\mathfrak{a}_0, \mathfrak{a}_1, \mathfrak{a}_2, \dots)$, we have that $\mathcal{W}(w) = \{0\}$ if and only if

$$\lim_{m \to \infty} \prod_{k=0}^{m} r_k = 0 \tag{6.12}$$

for every sequence $\{r_k\}_{k=0}^{\infty}$ with $r_k \in \mathcal{W}(\mathfrak{a}_k)$ for all $k \in \mathbb{N}$.

For an arrow $\mathfrak{a} \in \mathcal{A}$, we have that $\mathcal{W}(\mathfrak{a})$ is a set of real numbers, so we can write the *supremum weight* of $\mathfrak{a}$ as $\sup \mathcal{W}(\mathfrak{a})$. Similarly, for a walk w, we define $\sup \mathcal{W}(w)$ is the supremum weight of w.

Remark 6.1. Given a walk $w := (\mathfrak{a}_0, \mathfrak{a}_1, \ldots, \mathfrak{a}_N)$ through a graph with set-valued weights, the supremum weight $\sup \mathcal{W}(w)$ is not always equal to the product of the supremum weights of the arrows. That is, in some cases

$$\sup \mathcal{W}(w) \neq \big(\sup \mathcal{W}(\mathfrak{a}_0)\big)\big(\sup \mathcal{W}(\mathfrak{a}_1)\big) \cdots \big(\sup \mathcal{W}(\mathfrak{a}_N)\big).$$

For example, if $\mathcal{W}(\mathfrak{a}_0) = \{0\}$ and $\mathcal{W}(\mathfrak{a}_1) = (1, \infty)$, then $\sup \mathcal{W}(w) = 0$ but the product $\big(\sup \mathcal{W}(\mathfrak{a}_0)\big)\big(\sup \mathcal{W}(\mathfrak{a}_1)\big) = 0 \cdot \infty$ is undefined. Thus, it is important that the supremum is evaluated *after* computing the product.

The CTG is designed to be a simplified representation of a conical hybrid system $\mathcal{H}$ to facilitate the analysis of pre-asymptotic stability. To this end, we exploit properties of conical hybrid systems, along with assumptions on the continuous dynamics of the hybrid system, so that the CTG can be used to establish that the origin of $\mathcal{H}$ is pAS. In particular, we exploit two simplifications.

In a conical hybrid system, Proposition 6.1 asserts that the distance a solution starts from the origin of does not affect the way it can evolve (aside from scaling effects). Thus, if we consider any ray from the origin and allow every point in the ray to evolve according to the dynamics of $\mathcal{H}$, then that ray is (in a sense) preserved. Using this observation, the first simplification in the CTG comes from using the nrv function to map $\mathbb{R}^n$ to $\mathbb{S}_0^{n-1}$ so that each single point $p \in \mathbb{S}_0^{n-1}$ represents every point in $\operatorname{ray}(p)$.

Mapping $\mathbb{R}^n$ to $\mathbb{S}_0^{n-1}$ reduces the dimension by one and—more importantly—allows for recurrent walks through the CTG despite convergence of solutions (see Figure 6.1). For example, suppose that for some $v \in \mathbb{S}^{n-1}$, a solution φ to $\mathcal{H}$ repeatedly enters $\operatorname{ray}(v)$. That is, $\varphi(t_k, j_k) \in \operatorname{ray}(v)$ for a sequence of hybrid times $\{(t_k, j_k)\}$ in $\operatorname{dom}(\varphi)$. Then,

$$v = \operatorname{nrv}(\varphi(t_1, j_1)) = \operatorname{nrv}(\varphi(t_2, j_2)) = \cdots .$$

Furthermore, the set of possible rays that φ can transition into from $\varphi(t_k, j_k) \in \operatorname{ray}(v)$ via a single jump or flow is the same at every hybrid time (t_k, j_k) in the sequence. Exploiting this information allows us to uncover patterns in the behavior of $\mathcal{H}$.

By collapsing $\mathbb{R}^n$ to $\mathbb{S}_0^{n-1}$, however, we lose information about the magnitude (norm) of solutions. Instead, the weight of each arrow in the CTG typically contains every possible *relative* change of magnitude that a solution $(t, j) \mapsto \varphi(t, j)$ can

exhibit as $(t, j) \mapsto \mathrm{nrv}(\varphi(t, j))$ moves from the arrow's start vertex to its end vertex (both in $\mathbb{S}_0^{n-1}$) via a single jump or a single interval of flow.

The second simplification arising from the CTG is that it allows us to partition the analysis of pre-asymptotic stability by considering separately solutions that are eventually continuous and solutions that are not eventually continuous. A hybrid arc is called *eventually continuous* if it has an interval of flow after the last jump time in its hybrid time domain. The aspects of eventually continuous solutions that are relevant to pre-asymptotic stability in $\mathcal{H} = (C, f, D, g)$ can be determined by analyzing the continuous-time system (C, f). In particular, our results assume that 0_n is pAS for (C, f)—which is necessary for 0_n to be pAS for $\mathcal{H}$ and can be verified using methods from continuous-time system analysis. Thus, the CTG is a tool for analyzing the behavior of solutions that are not eventually continuous.

Assuming that 0_n is pAS (and thus stable) for (C, f) has the added benefit that if we can show that a given solution converges to 0_n at jump times, then we can establish asymptotic convergence without analyzing the trajectories of solutions *during* intervals of flow. This is shown in the following lemma.

Lemma 6.1. *Let $\mathcal{H} = (C, f, D, g)$ be a conical hybrid system with modes. Suppose that $\mathcal{O} := \mathcal{Q} \times \{0_n\}$ is stable for (C, f) and let φ be any solution to $\mathcal{H}$ with $\sup_j \mathrm{dom}(\varphi) = \infty$. Then,*

$$\lim_{j\to\infty} |\varphi(t_j, j)|_{\mathcal{O}} = 0 \implies \lim_{t+j\to\infty} |\varphi(t, j)|_{\mathcal{O}} = 0,$$

where each t_j is the jth jump time in $\mathrm{dom}(\varphi)$.

Proof. Let φ be any solution to $\mathcal{H}$ with $\sup_j \mathrm{dom}(\varphi) = \infty$. Let t_1, t_2, ..., be the jump times of φ and suppose that

$$\lim_{j\to\infty} |\varphi(t_j, j)|_{\mathcal{A}} = 0.$$

Take any $\varepsilon > 0$. We want to show that there exists $(t', j') \in \mathrm{dom}(\varphi)$ such that $|\varphi(t, j)|_{\mathcal{A}} < \varepsilon$ for all $(t, j) \in \mathrm{dom}(\varphi)$ such that $t + j \geq t' + j'$.

For each j such that $[t_j, t_{j+1}]$ is an interval of flow in $\mathrm{dom}(\varphi)$, the function $t \mapsto \varphi(t, j)$ is a solution to (C, f) for all $t \in [t_j, t_{j+1}]$. By the stability of $\mathcal{O}$ for (C, f), there exists $\delta \in (0, \varepsilon)$ such that

$$|\varphi(t_j, j)|_{\mathcal{A}} \leq \delta \implies |\varphi(t, j)|_{\mathcal{A}} \leq \varepsilon \quad \forall t \in [t_j, t_{j+1}]. \tag{6.13}$$

Since $j \mapsto \varphi(t_j, j)$ converges to $\mathcal{O}$, there exists $j' \in \mathbb{N}$ such that $|\varphi(t_j, j)|_{\mathcal{A}} \leq \delta$ for all $j \geq j'$. Let $t' := t_{j'}$. Thus, from (6.13), we have that $|\varphi(t, j)|_{\mathcal{A}} \leq \varepsilon$ for all $(t, j) \in \operatorname{dom}(\varphi)$ such that $t + j \geq t' + j'$. Since $\varepsilon > 0$ was arbitrary, we can take $\varepsilon \to 0$, thereby establishing that

$$\lim_{t+j\to\infty} |\varphi(t, j)|_{\mathcal{A}} = 0. \qquad \square$$

As a consequence of Lemma 6.1, when determining whether persistently jumping solutions converge to $\mathcal{O}$ (e.g., to establish pre-asymptotic stability), we can ignore the interior of intervals of flow and only focus on showing that the solution at jump times converges. By doing so, we treat flows as discrete transitions that take solutions from their values immediately after a jump to their values immediately before the next jump. This effectively ignores the ordinary time required to traverse the flow because it is irrelevant for determining pre-asymptotic stability. Based on this fact, we generalize a flow that takes a solution φ from $x^{(0)} \in \mathbb{R}^n$ to $x^{(\mathrm{f})} \in \mathbb{R}^n$ in mode $q \in \mathcal{Q}$ as a flow arrow $(q, \operatorname{nrv}(x^{(0)})) \xrightarrow{\mathrm{F}} (q, \operatorname{nrv}(x^{(\mathrm{f})}))$ in the CTG.

We design the CTG as a directed graph with set-valued weights with vertices that live in $\mathcal{Q} \times \mathbb{S}_0^{n-1}$. Each tuple $v := (q, s)$ in $\mathcal{Q} \times \mathbb{S}_0^{n-1}$ is a vertex in the CTG if it is possible for a solution to $\mathcal{H}$ to jump from or to v (i.e., if $v \in D \cup g(D)$). An arrow points between vertices $v_1 := (q_1, s_1)$ and $v_2 := (q_2, s_2)$ in the CTG if a solution to $\mathcal{H}$ can move directly from s_1 in mode q_1 to $\operatorname{ray}(v_2)$ in mode q_2 by a single jump or a single interval of flow. Each arrow is labeled by the type of transition it represents (either flow or jump). The weight of the arrow $v_1 \to v_2$ typically stores the relative change in the magnitude of a solution that starts at v_1 and ends in $\operatorname{ray}(v_2)$ (except if $s_1 = 0_n$, in which case the weight stores the absolute change—but the occurrence of such cases is limited). By multiplying together the weights of all the arrows in each walk through the CTG, we can analyze the relative change in distance of solutions from the origin (see Proposition 6.5, below).

Definition 6.3 (Conical Transition Graph)**.** Let $\mathcal{H} = (C, f, D, g)$ be a conical hybrid system on $\mathbb{R}^n$ with modes $\mathcal{Q}$. Let $\mathcal{L} := \{\text{“J”}, \text{“F”}\}$ be a set of labels (J stands for *jump* and F stands for *flow*). The CTG of $\mathcal{H}$ is a weighted, directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{W})$ where $\mathcal{V} \subset \mathcal{Q} \times \mathbb{S}_0^{n-1}$ is a set of *vertices*, $\mathcal{A} \subset \mathcal{V}^2 \times \mathcal{L}$ is a set of *arrows* between vertices, and $\mathcal{W} : \mathcal{A} \rightrightarrows \mathbb{R}_{\geq 0}$ is a set-valued *weight function* that assigns a set of

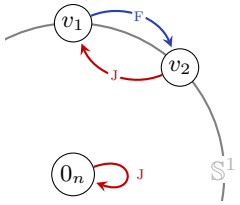

Figure 6.4. Conical transition graph for $\mathcal{H}$ in Example 6.3.

nonnegative weights to each arrow. The set of vertices is defined as

$$\mathcal{V} := \big(D \cup g(D)\big) \cap \big(\mathcal{Q} \times \mathbb{S}_0^{n-1}\big). \tag{6.14}$$

For each $v^{\ominus} := (q^{\ominus},\, s^{\ominus}) \in \mathcal{V} \cap D$, and each $(q^{\oplus}, z^{\oplus}) \in g(v^{\ominus})$, and for $s^{\oplus} := \mathrm{nrv}(z^{\oplus})$; a *jump arrow* $\mathfrak{a}^{\mathrm{J}} = v^{\ominus} \xrightarrow{\mathrm{J}} v^{\oplus}$ points from $v^{\ominus}$ to

$$v^{\oplus} := \big(q^{\oplus},\, s^{\oplus}\big) = \big(q^{\oplus},\, \mathrm{nrv}(A_e s^{\ominus})\big), \tag{6.15}$$

where $e := (q^{\ominus}, q^{\oplus})$. The weight of $\mathfrak{a}^{\mathrm{J}} = v^{\ominus} \xrightarrow{\mathrm{J}} v^{\oplus}$ is the (singleton) set

$$\mathcal{W}(\mathfrak{a}^{\mathrm{J}}) := \big\{|z^{\oplus}|\big\} = \big\{|A_e s^{\ominus}|\big\}. \tag{6.16}$$

There is a *flow arrow* $\mathfrak{a}^{\mathrm{F}} = v^{(0)} \xrightarrow{\mathrm{F}} v^{(\mathrm{f})}$ from $v^{(0)} := (q, s^{(0)}) \in \mathcal{V} \cap g(D)$ to $v^{(\mathrm{f})} := (q, s^{(\mathrm{f})}) \in \mathcal{V} \cap D$ if for some $\tau > 0$, there exists a function $\xi : [0, \tau] \to \mathbb{R}^n$ such that

$$\xi(0) = s^{(0)} \tag{6.17a}$$

$$\dot{\xi}(t) = f_q(\xi(t)) \quad \forall t \in (0, \tau) \tag{6.17b}$$

$$\xi(t) \in C_q \qquad \forall t \in (0, \tau) \tag{6.17c}$$

$$\mathrm{nrv}(\xi(\tau)) = s^{(\mathrm{f})}. \tag{6.17d}$$

The weight of $\mathfrak{a}^{\mathrm{F}}$ is

$$\mathcal{W}(\mathfrak{a}^{\mathrm{F}}) := \big\{|\xi(\tau)| \;\big|\; \xi : [0, \tau] \to \mathbb{R}^n \text{ satisfies (6.17) for some } \tau > 0\big\}. \tag{6.18}$$

That is, for each solution $\xi : [0, \tau] \to \mathbb{R}^n$ of (6.17)—which has $|\xi(0)| = 1$ (if $|s^{(0)}| = 1$) or $|\xi(0)| = 0$ (if $|s^{(0)}| = 0$)—the magnitude of ξ at time τ is an element of the weight set: $|\xi(\tau)| \in \mathcal{W}(\mathfrak{a}^{\mathrm{F}})$. ◇

Note that each *vertex* in a CTG is a tuple containing a mode $q \in \mathcal{Q}$ and a vector $v \in \mathbb{R}^n$ with $v \in \mathbb{S}_0^{n-1}$.

If an arrow $\mathfrak{a} := v_1 \to v_2$ points from $v_1 := (q_1, s_1) \in \mathcal{V}$ to $v_2 := (q_2, s_2) \in \mathcal{V}$ with $s_1 \neq 0_n$, then the weight of $\mathfrak{a}$ is the set of all of the possible *relative* changes in the magnitude of a solution that transitions from $\operatorname{ray}(s_1)$ in mode q_1 to $\operatorname{ray}(s_2)$ in mode q_2 via a single jump or interval of flow (the mode can change only for jump arrows. For a flow arrow, $q_1 = q_2$). On the other hand, if $s_1 = 0_n$, then the weight of $\mathfrak{a}$ is the set of all of the possible *absolute* changes in magnitude for a transition from 0_n to $\operatorname{ray}(s_2)$ via a single jump or interval of flow (the relative change of distance is undefined because the initial distance 0 would be in the denominator).

In the following example, we consider a conical hybrid system with a single mode, so we omit the logic variable. In particular, we will consider only mode $q = 0$ from Example 6.1. To simplify the exposition, we will omit the mode variable "q" during this example.

Example 6.3. Consider the following conical hybrid system on $\mathbb{R}^2_{\geq 0}$ (the non-negative quadrant of $\mathbb{R}^2$) with a single mode:

$$\mathcal{H}: \begin{cases} f(x) := \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \forall x \in C := \left\{ x \in \mathbb{R}^2_{\geq 0} \;\middle|\; x_2 \geq x_1 \right\}, \\ g(x) := \begin{bmatrix} 0 \\ \gamma x_1 \end{bmatrix} & \forall x \in D := \overline{\operatorname{ray}} \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right] = \left\{ x \in \mathbb{R}^2_{\geq 0} \;\middle|\; x_2 = x_1 \right\}, \end{cases} \tag{6.19}$$

with $\gamma > 0$. We will construct the CTG for $\mathcal{H}$. Let $v_1 := \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]$ and $v_2 := \frac{1}{\sqrt{2}} \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]$, so $g(D) = \overline{\operatorname{ray}}\, v_1$ and $D = \overline{\operatorname{ray}}\, v_2$. Thus, the set of vertices is

$$\mathcal{V} = \big(\{0_n\} \cup \operatorname{ray} v_1 \cup \operatorname{ray} v_2\big) \cap \mathbb{S}^{n-1}_0 = \{0_n, v_1, v_2\}$$

and the set of arrows is

$$\mathcal{A} = \{\underbrace{0_n \xrightarrow{\text{J}} 0_n,\ v_2 \xrightarrow{\text{J}} v_1}_{\text{Jump arrows}},\ \underbrace{v_1 \xrightarrow{\text{F}} v_2}_{\text{Flow arrow}}\}.$$

The CTG of $\mathcal{H}$ is depicted in Figure 6.4. ◇

Example 6.4 (Example 6.1, cont.)**.** Now, we will consider the full conical hybrid system $\mathcal{H}$ with modes from Example 6.1. By examining Figure 6.2 and the data of the system, we find that the vertices in the CTG are

$$\big(0, 0_n\big), v_0 := \big(0, \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]\big), v_1 := \big(0, \operatorname{nrv}\left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]\big), \big(1, 0_n\big), v_2 := \big(1, \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]\big), v_3 := \big(1, \left[\begin{smallmatrix} \cos\theta \\ \sin\theta \end{smallmatrix}\right]\big),$$

and the arrows are

$$\begin{array}{lll} (0,0_n) \xrightarrow{\mathrm{J}} (0,0_n), & (1,0_n) \xrightarrow{\mathrm{J}} (1,0_n), & (1,0_n) \xrightarrow{\mathrm{F}} (1,0_n), \\ (0,0_n) \xrightarrow{\mathrm{J}} (1,0_n), & (1,0_n) \xrightarrow{\mathrm{J}} (0,0_n), & v_0 \xrightarrow{\mathrm{F}} v_1, \\ v_1 \xrightarrow{\mathrm{J}} v_0, & v_1 \xrightarrow{\mathrm{J}} v_2, & v_2 \xrightarrow{\mathrm{F}} v_3, \\ v_3 \xrightarrow{\mathrm{J}} v_0. & & \end{array} \tag{6.20}$$

There is not a flow arrow from v_3 to v_2 because flow arrows must start in $G(D)$, nor is $v_2 \xrightarrow{\mathrm{F}} v_2$ because flow arrows must end in D. ◇

The need for the weights to be set-valued comes from the fact that there may be multiple solutions to (6.17) with different final magnitudes, $|\xi(T)|$, as in (6.18). The following example presents a conical hybrid system with a flow arrow that has a non-singleton weight.

Example 6.5. Consider the following conical hybrid system:

$$\mathcal{H} : \begin{cases} \dot{x} &= f(x) := -1 \quad x \in C := \mathbb{R}_{\geq 0}, \\ x^+ &= g(x) := x/2 \quad x \in D := \mathbb{R}_{\geq 0}. \end{cases}$$

Every maximal solution to $\mathcal{H}$ evolves by a non-deterministic combination of flows and jumps until it reaches 0_n, at which point it must jump from 0_n to 0_n forevermore. Thus, 0_n is pre-asymptotically stable for $\mathcal{H}$.

The vertex set of the CTG is $\mathcal{V} = \{0, 1\}$ and the arrow set is

$$\mathcal{A} = \left\{0 \xrightarrow{\mathrm{J}} 0,\ 1 \xrightarrow{\mathrm{J}} 1,\ 1 \xrightarrow{\mathrm{F}} 0,\ 1 \xrightarrow{\mathrm{F}} 1\right\}.$$

Consider, in particular, the arrow $1 \xrightarrow{\mathrm{F}} 1$. For all $T \in (0, 1)$, the function

$$\begin{aligned} \xi : [0, T] &\to \mathbb{R}_{\geq 0} \\ t &\mapsto \xi(t) := 1 - t \end{aligned}$$

satisfies (6.17) with $v^{(0)} := 1$, $v^{(\mathrm{f})} := 1$, and

$$|\xi(T)| = 1 - T \in (0, 1).$$

Thus, $1 \xrightarrow{\mathrm{F}} 1$ is a flow arrow in the CTG with set-valued weight $\mathcal{W}(1 \xrightarrow{\mathrm{F}} 1) = (0, 1)$. ◇

Whereas non-singleton weights for conical hybrid systems with constant flows are typically continuous intervals, such as $(0,1)$, for conical hybrid systems with linear flows, non-singleton weights are infinite sets of discrete points, as shown in the next example.

Example 6.6. Consider the conical hybrid $\mathcal{H} = (C, f, D, g)$ on $\mathbb{R}^2$ with dynamics given by

$$\begin{cases} f(x) := Ax & \forall x \in C := \mathbb{R}^2 \\ g(x) := \begin{bmatrix} -x_1 \\ 0 \end{bmatrix} & \forall x \in D := \overline{\text{ray}}\left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right], \end{cases}$$

where $A := \left[\begin{smallmatrix} \gamma & -1 \\ 1 & \gamma \end{smallmatrix}\right]$ and $\gamma \in \mathbb{R}$. The vertex set for the conical transition graph is $\mathcal{V} = \{0_n, \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right]\}$. It can be shown that there are two jump arrows $0_n \xrightarrow{\text{J}} 0_n$ and $\left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right] \xrightarrow{\text{J}} \left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right]$, and one flow arrow $\left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right] \xrightarrow{\text{F}} \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right]$ (recall that the start of a flow arrow must be in $g(D)$ and the end must be in D). The weights for the jump arrows are

$$\mathcal{W}(0_n \xrightarrow{\text{J}} 0_n) = \{0\} \quad \text{and} \quad \mathcal{W}(\left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right] \xrightarrow{\text{J}} \left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right]) = \{1\}.$$

Solutions to (6.17) for the flow arrow $\mathfrak{a}^{\text{F}} := \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right] \xrightarrow{\text{F}} \left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right]$ are given for each $T \in \{\pi, 3\pi, 5\pi, \dots\}$ by

$$t \mapsto \xi(t) := \exp(\gamma t) \begin{bmatrix} \cos t \\ -\sin t \end{bmatrix} \quad \forall t \in [0, T].$$

At $t = T$, the magnitude of ξ is $|\xi(T)| = \exp(\gamma T)$. Thus, the weight of $\mathfrak{a}$ is

$$\mathcal{W}(\mathfrak{a}^{\text{F}}) = \{\exp(\gamma T) \mid T = \pi, 3\pi, 5\pi, \dots\}. \qquad \diamond$$

In addition to having a non-singleton weight, the flow arrow $1 \xrightarrow{\text{F}} 1$ in Example 6.5 illustrates an exceptional case that we must consider. In Example 6.5, the origin is pAS for $\mathcal{H}$, so we want every infinite-length walk through the CTG to have weight $\{0\}$ (see Theorem 6.2, below). But, the weight of $w := 1 \xrightarrow{\text{F}} 1 \xrightarrow{\text{F}} 1 \xrightarrow{\text{F}} \cdots$ is actually $\mathcal{W}(w) = [0, 1)$. To see $\mathcal{W}(w)$ contains $(0, 1)$, take any $s > 0$ and let $r_k := \exp(-s/2^{k+1})$, which is in $\mathcal{W}(1 \xrightarrow{\text{F}} 1) = (0, 1)$ for each $k \in \mathbb{N}$. Then, by selecting $\{r_k\}_{k=0}^{\infty}$ in (6.11), we compute

$$\prod_{k=0}^{\infty} r_k = \exp\big(-s/2 - s/4 - s/8 - \cdots\big) = e^{-s} \in (0, 1).$$

Alternatively, selecting $r_k := 1/2 \in \mathcal{W}(1 \xrightarrow{\mathrm{F}} 1)$ results in $\prod_{k=0}^{\infty} 1/2 = 0$. Hence, $\mathcal{W}(w) = [0, 1)$. The crux of the problem is that by repeatedly traversing the loop $1 \xrightarrow{\mathrm{F}} 1$, the walk w represents a solution that flows part of the way to the origin, then flows a little more, and a little more, *ad infinitum*, without ever jumping. As indicated by the weight $\mathcal{W}(w)$, we can construct such as sequence of flows that will converge to 0, but also sequences that converge to any value in $[0, 1)$. Fortunately, any finite sequence of consecutive flow arrows can be replaced by a single flow arrow, whereas any infinite sequence of flow arrows represents a solution that never jumps, so we analyze it using continuous-time methods instead of the CTG. Therefore, we exclude walks with consecutive flow arrows from consideration.

Definition 6.4 (Well-formed Walk)**.** We say that a walk w through a conical transition graph $\mathcal{G}$ is *well-formed* if no pair of consecutive arrows in w are both flow arrows. That is, $w = (\mathfrak{a}_0, \mathfrak{a}_1, \ldots, \mathfrak{a}_{N-1})$ is a well-formed walk through $\mathcal{G}$ if for every $i \in \{1, 2, \ldots, N-1\}$, either $\mathfrak{a}_{i-1}$ or $\mathfrak{a}_i$ is a jump arrow. $\diamond$

Remark 6.2. A well-formed walk may include consecutive jump arrows.

6.5 Establishing Pre-asymptotic Stability via the CTG

This section presents a result that allows for pre-asymptotic stability of $\mathcal{O} := \mathcal{Q} \times \{0_n\}$ (the combined origins of all of the modes) to be established by analyzing the CTG. For $\mathcal{O}$ to be pre-asymptotically stable, $\mathcal{O}$ must be forward invariant. Forward invariance of $\mathcal{O}$ can be easily checked for a conical hybrid system, as asserted by the following result.

Proposition 6.2. *The set $\mathcal{O} := \mathcal{Q} \times \{0_n\}$ is not forward invariant with respect to conical hybrid system $\mathcal{H}$ if and only if it has a mode $q_{\mathrm{C}} \in \mathcal{Q}$ with constant flows and $f_{q_{\mathrm{C}}} \in C_{q_{\mathrm{C}}} \setminus \{0_n\}$. Furthermore, if $\mathcal{O}$ is not forward invariant, then there exists a complete solution φ to $\mathcal{H}$ such that*

$$\lim_{t+j\to\infty} |\varphi(t, j)|_{\mathcal{O}} = \infty.$$

Proof. Suppose $\mathcal{O}$ is not forward invariant. From the definition of g in (6.6), we find $g(\mathcal{O}) \subset \mathcal{O}$, so solutions to $\mathcal{H}$ cannot leave the origin at jumps. Thus, for some $q_{\mathrm{C}} \in \mathcal{Q}$, solutions to $\mathcal{H}$ can flow away from the origin. Flows in q_{C} are either linear or

constant. In both cases, the flow map is Lipschitz continuous, so solutions are unique. If flows are linear, then $f_{q_C}(0_n) = 0_n$, so all solutions to that start in $\{q_C\} \times \{0_n\}$ remain in $\{q_C\} \times \{0_n\}$. Hence, flows cannot be linear. Similarly, if flows are constant and $f_{q_C} = 0_n$, then solutions cannot leave the origin, so we must have constant flows with $f_{q_C} \neq 0_n$.

It remains to be shown that $f_{q_C} \in C$. If $f_{q_C} \notin C$, then any solution to $\dot{z} = f_{q_C}$ from $z(0) = 0_n$ immediately leaves C_{q_C}, so solutions to $\mathcal{H}$ cannot flow from the origin, contradicting our assumption that the origin is not forward invariant. Therefore, $f_{q_C} \in C_{q_C} \setminus \{0_n\}$.

To prove the converse direction, suppose mode q_C has constant flows and $f_{q_C} \in C_{q_C} \setminus \{0_n\}$. Then $\varphi : \mathbb{R}_{\geq 0} \times \{0\} \to \mathbb{R}^n$ defined by

$$\varphi(t, 0) := (q_C, t f_{q_C}) \quad \forall t \geq 0$$

is a complete solution to $\mathcal{H}$. (Since $f_{q_C} \in C_{q_C}$ and C_{q_C} is a cone, $t f_{q_C}$ is also in C_{q_C} for all $t \geq 0$.) Finally, since $f_{q_C} \geq 0$, we have that $|\varphi(t, j)|_{\mathcal{O}} \to \infty$. □

For a simple illustration of Proposition 6.2, consider $\mathcal{H}$ on $\mathbb{R}^n$ with a single mode q that has constant flows $\dot{z} = f_q^*$ and a flow set consisting of a single ray, $C := \overline{\text{ray}}\, f_q^*$, where $f_q^* \in \mathbb{S}^{n-1}$. We have $f_q^* \in C$, so, by Proposition 6.2, the set $\mathcal{O}$ is not forward invariant for $\mathcal{H}$. The hybrid arc $\varphi : \mathbb{R}_{\geq 0} \times \{0\} \to \{q\} \times \mathbb{R}^n$ defined by $\varphi(t, j) := \left(q, t f_q^*\right)$ for all $(t, j) \in \text{dom}(\varphi)$ is a complete solution to $\mathcal{H}$ that leaves 0_n, and $\lim_{t+j \to \infty} |\varphi(t, j)|_{\mathcal{O}} = \infty$.

6.5.1 CTG Simulations

This section establishes a correspondence between solutions to a conical hybrid system and walks through the CTG. For each solution, there is a unique walk called the CTG-simulation of that solution (Definition 6.5). That a CTG-simulation is, in fact, a walk through the CTG is asserted in Proposition 6.3. Conversely, Proposition 6.4 asserts that for every well-formed nonempty walk through the CTG of a hybrid system that starts and ends with a jump arrow, there exists a solution that the walk simulates. This section is concluded with Proposition 6.5, which shows that the relative change in the magnitude of a solution is an element in the set-valued weight of the solutions CTG-simulation.

Definition 6.5 (CTG Simulation)**.** Let $\mathcal{H}$ be a conical hybrid system with modes $\mathcal{Q}$ and conical transition graph $\mathcal{G}$. Consider any solution $(t,j) \mapsto \varphi(t,j) := \big(q(j), z(t,j)\big)$ to $\mathcal{H}$ that jumps at least once. Let $J := \sup_j \operatorname{dom}(\varphi) \in \{1,2,\ldots,\infty\}$. Let $t_0 := 0$ and let t_j denote the jth jump time of φ for each $j \in \{1,2,\ldots,J\}$. Let $K_0 := 0$ and for each finite $j \in \{1,\ldots,J\}$, let K_j be the cumulative number of jumps and intervals of flow in φ between $(t_1,0) \in \operatorname{dom}(\varphi)$ and $(t_j,j) \in \operatorname{dom}(\varphi)$. Let $h_0 := (t_1,0)$, and for each $k \in \{1,\ldots,K_J\}$, let h_k be the first hybrid time among

$$(t_1,1),\ (t_2,1),\ (t_2,2),\ \ldots,\ (t_J,J-1),\ (t_J,J) \tag{6.21}$$

that does not occur among h_0, h_1, …, h_{k-1}. We denote the t-component of h_k as $\pi_{\mathrm{T}}(h_k)$ and the j-component as $\pi_{\mathrm{J}}(h_k)$, i.e., $h_k = (\pi_{\mathrm{T}}(h_k), \pi_{\mathrm{J}}(h_k))$. Note that for each $k \in \{0,1,\ldots,K_J-1\}$, either $\pi_{\mathrm{J}}(h_k) = \pi_{\mathrm{J}}(h_{k+1})$ and $\pi_{\mathrm{T}}(h_k) < \pi_{\mathrm{T}}(h_{k+1})$, or $\pi_{\mathrm{J}}(h_k) < \pi_{\mathrm{J}}(h_{k+1})$ and $\pi_{\mathrm{T}}(h_k) = \pi_{\mathrm{T}}(h_{k+1})$.

We say that

$$w := (v_0 \xrightarrow{\ell_0} v_1 \xrightarrow{\ell_1} \ldots \xrightarrow{\ell_{(K_J-2)}} v_{(K_J-1)} \xrightarrow{\ell_{(K_J-1)}} v_{K_J})$$

is the *CTG simulation* or the *$\mathcal{G}$-simulation* of φ, where $\{v_k\}_{k=0}^{K_J}$ is a sequence in $\mathcal{Q} \times \mathbb{S}_0^{n-1}$ defined as

$$v_k := \big(q(h_k), \operatorname{nrv}(z(h_k))\big) \quad \forall k \in \{0,1,\ldots,K_J\} \tag{6.22}$$

and $\{\ell_k\}_{k=0}^{K_J-1}$ is a sequence of labels in $\mathcal{L}$ defined by

$$\ell_k := \begin{cases} \mathrm{J} & \text{if } \pi_{\mathrm{J}}(h_{k+1}) > \pi_{\mathrm{J}}(h_k) \\ \mathrm{F} & \text{if } \pi_{\mathrm{T}}(h_{k+1}) > \pi_{\mathrm{T}}(h_k) \end{cases} \quad \forall k \in \{0,1,\ldots,K_J-1\}. \tag{6.23}$$

$\diamond$

Remark 6.3. A CTG simulation of a solution φ is a representation of φ with "snapshots" of the solution projected onto $\mathbb{S}_0^{n-1}$ by the nrv function before and after each jump. Such a simulation does not say anything about how φ flows before the first jump or after the last jump.

Lemma 6.2. *Suppose $\mathcal{H} := (C,f,D,G)$ is a conical hybrid system with modes $\mathcal{Q}$ and transitions $\mathcal{E}$. For any $(q^\ominus, z^\ominus) \in D$ and $(q^\oplus, z^\oplus) \in G(q^\ominus, z^\ominus)$, let $s^\ominus := \operatorname{nrv}(z^\ominus)$ and $s^\oplus := \operatorname{nrv}(z^\oplus)$. Then,*

$$v^\ominus := \big(q^\ominus, s^\ominus\big) \in \mathcal{V} \cap D, \quad v^\oplus := \big(q^\oplus, s^\oplus\big) \in \mathcal{V} \cap G(D),$$

and $\mathfrak{a}^{\mathrm{J}} := v^\ominus \xrightarrow{\mathrm{J}} v^\oplus$ is a jump arrow in the CTG of $\mathcal{H}$.

Furthermore, if $(t,j) \mapsto \varphi(t,j) = \big(q(j), z(t,j)\big)$ *is a solution to* $\mathcal{H}$*, then for each jump time* t_j *in* $\operatorname{dom}(\varphi)$*,*

$$\big(q(j-1), \operatorname{nrv}(z(t_j, j-1))\big) \xrightarrow{\text{J}} \big(q(j), \operatorname{nrv}(z(t_j, j))\big) \tag{6.24}$$

is a jump arrow in $\mathcal{G}$*.*

Proof. Take any $(q^\ominus, z^\ominus) \in D$ and $(q^\oplus, z^\oplus) \in G(q^\ominus, z^\ominus)$. It follows immediately from the definition of the jump set that $e := (q^\ominus, q^\oplus) \in \mathcal{E}$ and $z^\ominus \in D_e$. Since D_e is a cone, $s^\ominus := \operatorname{nrv}(z^\ominus)$ is also in D_e, so $v^\ominus := (q^\ominus, s^\ominus) \in \mathcal{V} \cap D$.

Next, we will show that $v^\oplus := (q^\oplus, s^\oplus)$ is a vertex in $\mathcal{V} \cap G(D)$ (specifically, $v^\oplus \in \mathcal{V} \cap G(D)$), where $s^\oplus := \operatorname{nrv}(z^\oplus)$. Let

$$z^* := \begin{cases} s^\ominus / |A_e s^\ominus| & \text{if } A_e s^\ominus \neq 0_n \\ s^\ominus & \text{if } A_e s^\ominus = 0_n. \end{cases}$$

Since $s^\ominus \in D_e$ and D_e is a cone, we have that $z^* \in D_e$, so $(q^\ominus, z^*) \in D$. Then,

$$A_e z^* = s^\oplus.$$

To see why, first suppose that $A_e s^\ominus \neq 0_n$. Then,

$$A_e z^* = A_e(s^\ominus / |A_e s^\ominus|) = \operatorname{nrv}(A_e s^\ominus) = \operatorname{nrv}(A_e z^\ominus) = s^\oplus,$$

where the penultimate equality is a result of (6.5). On the other hand, if $A_e s^\ominus = 0_n$, then $A_e z^* = 0_n = s^\oplus$. Therefore, $v^\oplus \in G(q^\ominus, z^*)$, so $v^\oplus$ is in $G(D)$ and $\mathcal{V}$.

To finish the proof, we must show that $v^\ominus \xrightarrow{\text{J}} v^\oplus$ is a jump arrow in the CTG of $\mathcal{H}$. Using the definitions of $s^\oplus$ and $z^\oplus$, we have that $s^\oplus = \operatorname{nrv}(z^\oplus) = \operatorname{nrv}(A_e z^\ominus)$. By linearity, $A_e z^\ominus = |z^\ominus| A_e s^\ominus$, so $\operatorname{nrv}(A_e z^\ominus) = \operatorname{nrv}(A_e s^\ominus)$. Therefore, per (6.15), $v^\ominus \to v^\oplus$ is a jump arrow.

Finally, (6.24) is a jump arrow in $\mathcal{G}$ since $\varphi(t_j, j-1) \in D$ at each jump time t_j. □

Lemma 6.3. *Consider a conical hybrid system* $\mathcal{H} := (C, f, D, G)$ *with modes* $\mathcal{Q}$ *and transitions* $\mathcal{E}$*. Let* $(t,j) \mapsto \varphi(t,j) = \big(q(j), z(t,j)\big)$ *be any solution to* $\mathcal{H}$ *with jump times* t_j *and* $J := \sup_j \operatorname{dom}(\varphi)$*. For each interval flow* $[t_j, t_{j+1}]$ *in* $\operatorname{dom}(\varphi)$*, if* $j \in \{1, 2, \ldots, J-1\}$*, then*

$$\big(q(j), \operatorname{nrv}(z(t_j, j))\big) \xrightarrow{\text{F}} \big(q(j), \operatorname{nrv}(z(t_{j+1}, j))\big) \tag{6.25}$$

is a flow arrow in $\mathcal{G}$*.*

Proof. Let $(t,j) \mapsto \varphi(t,j) = \big(q(j), z(t,j)\big)$ be a solution to $\mathcal{H}$. Without loss of generality, suppose $J > 1$ (otherwise the conclusion is vacuously true). Take any $j \in \{1, 2, \ldots, J-1\}$. Let $z^{(0)} := z(t_j, j)$, $s^{(0)} := \operatorname{nrv}(z^{(0)})$, $z^{(\mathrm{f})} := z(t_{j+1}, j)$, and $s^{(\mathrm{f})} := \operatorname{nrv}(z^{(\mathrm{f})})$.

Then, $\varphi(t_j, j) \in G(\varphi(t_j, j-1))$, so $v^{(0)} := (q(j), s^{(0)})$ is a vertex in $\mathcal{V}$. Similarly, $\varphi(t_{j+1}, j) \in D$, so $v^{(\mathrm{f})} := (q(j), s^{(\mathrm{f})})$ is in $\mathcal{V}$.

To show that $v^{(0)} \xrightarrow{\mathrm{F}} v^{(\mathrm{f})}$ is a flow arrow, for $\tau := t_{j+1} - t_j$, let $\xi : [0, \tau] \to \mathbb{R}^n$ be defined by

$$t \mapsto \xi(t) := \begin{cases} z(t + t_j, j)/|z^{(0)}| & \text{if } |z^{(0)}| \neq 0 \\ z(t + t_j, j) & \text{if } |z^{(0)}| = 0_n. \end{cases}$$

Since $[t_j, t_{j+1}]$ is an interval of flow, τ is positive. We will check each condition in the flow arrow conditions (6.17). Equation (6.17a) is satisfied because $\xi(0) = \operatorname{nrv}(z^{(0)}) = s^{(0)}$. From the flow condition (1.6b) of hybrid solutions, we have that $\dot{z}(t,j) = f_q(z(t,j))$ for almost all $t \in [t_j, t_{j+1}]$. Since f_q is either constant or linear, $t \mapsto z(j,t)$ is the unique solution to $\dot{x} = f_q(x)$ and $\dot{z}(j,t) = f_q(z(j,t))$ for all $t \in (t_j, t_{j+1})$ (rather than merely *almost all*). Therefore, (6.17b) is satisfied:

$$\frac{d\xi}{dt}(t) = \frac{dz}{dt}(t + t_j) = f_q(z(t + t_j)) = f_q(\xi(t)) \quad \forall t \in (0, \tau).$$

By (1.6a), $\varphi(t,j) \in C$ for all $t \in (t_j, t_{j+1})$, so $z(t,j) \in C_q$ for all $t \in (t_j, t_{j+1})$ and

$$\xi(t) \in C_q \quad \forall t \in (0, T),$$

satisfying (6.17c).

Finally, (6.17d) is satisfied:

$$\begin{aligned} \operatorname{nrv}(\xi(T)) &= \begin{cases} \operatorname{nrv}(z^{(\mathrm{f})}/|z^{(0)}|) & \text{if } |z^{(0)}| \neq 0 \\ \operatorname{nrv}(z^{(\mathrm{f})}) & \text{if } |z^{(0)}| = 0 \end{cases} \\ &= \operatorname{nrv}(z^{(\mathrm{f})}) = s^{(\mathrm{f})}. \end{aligned}$$

Therefore, $v^{(0)} \xrightarrow{\mathrm{F}} v^{(\mathrm{f})}$ is a flow arrow in $\mathcal{G}$. □

Proposition 6.3. *Consider a conical hybrid system $\mathcal{H}$ with conical transition graph $\mathcal{G}$. For any solution φ to $\mathcal{H}$, the $\mathcal{G}$-simulation of φ is a well-formed walk through $\mathcal{G}$.*

Proof. Let w be the $\mathcal{G}$-simulation of φ, and let $\{K_j\}_{j=0}^{J}$, $\{v_k\}_{k=0}^{K_J}$, and $\{\ell_k\}_{k=0}^{K_J-1}$ be defined as in Definition 6.5. We write the components of φ as $\varphi(t,j) = \big(q(j), z(t,j)\big)$.

To show that w is a walk through $\mathcal{G}$, we must show that each v_k is a vertex in $\mathcal{V}$, and for each $k \in \{0, 1, \ldots, K_J\}$ that $v_k \to v_{k+1}$ is an arrow in $\mathcal{G}$. The values of K_j always increment by $+1$ or $+2$, i.e., $K_{(j+1)} \in \{K_j + 1, K_j + 2\}$. Thus, for each $j \in \{0, 1, \ldots, J\}$, we need to show that $v_{K_j} \in \mathcal{V}$ and (if $j < J$) that $v_{K_j+1} \in \mathcal{V}$. Furthermore, we need to show $v_{K_j} \xrightarrow{\ell_{K_j}} v_{K_j+1}$ is an arrow in $\mathcal{A}$. If $K_{(j+1)} = K_j + 2$, we also need to show $v_{K_j+1} \xrightarrow{\ell_{K_j+1}} v_{K_j+2}$ is an arrow in $\mathcal{A}$. We will consider separately the cases of $K_{(j+1)} = K_j + 1$ and $K_{(j+1)} = K_j + 2$.

Take any $j \in \{0, 1, \ldots, J\}$.

Case 1 ($K_{(j+1)} = K_j + 1$). Suppose $K_{(j+1)} = K_j + 1$, which requires that either $j = 0$ or $t_j = t_{j+1}$. For the case where $j = 0$, there is a jump arrow in $\mathcal{G}$ from $v_0 = \big(q(0), \mathrm{nrv}(z(t_1, 0))\big)$ to $v_1 = \big(q(1), \mathrm{nrv}(t_1, 1)\big)$ per Lemma 6.2, since t_1 is a jump time in $\mathrm{dom}(\varphi)$. Similarly, if $t_j = t_{j+1}$, then

$$\big(q(j), \mathrm{nrv}(z(t_j, j))\big) = \big(q(j), \mathrm{nrv}(z(t_{j+1}, j))\big) \in D,$$

so, again by Lemma 6.2, there is a jump arrow in $\mathcal{G}$ from

$$v_{K_j} = \big(q(j), \mathrm{nrv}(z(t_j, j))\big) \quad \text{to} \quad v_{K_{(j+1)}} = \big(q(j+1), \mathrm{nrv}(t_{j+1}, j+1)\big).$$

Case 2 ($K_{(j+1)} = K_j + 2$). Suppose $K_{(j+1)} = K_j + 2$. From the definition of K_j, it follows that $j \in \{1, 2, \ldots, J-1\}$ and $[t_j, t_{j+1}]$ is an interval of flow in $\mathrm{dom}(\varphi)$. By Lemma 6.3, there is a flow arrow in $\mathcal{G}$ from $v_{K_j} = \big(q(j), \mathrm{nrv}(z(t_j, j))\big)$ to $v_{K_j+1} = \big(q(j), z(t_{j+1}, j)\big)$. Additionally, because t_{j+1} is a jump time, there is a jump arrow in $\mathcal{G}$ from

$$v_{(K_j+1)} \quad \text{to} \quad v_{(K_j+2)} = v_{K_{(j+1)}} = \big(q(j+1), \mathrm{nrv}(t_{j+1}, j+1)\big),$$

per Lemma 6.2.

Therefore, we have shown that each v_k is a vertex in $\mathcal{V}$ and each step in w is an arrow in $\mathcal{A}$, so w is a walk through $\mathcal{G}$. Furthermore, each flow arrow in w is followed by a jump arrow, as shown in Case 2, so w is well-formed. □

Proposition 6.4. *Consider a conical hybrid system $\mathcal{H}$ with modes $\mathcal{Q}$ and conical transition graph $\mathcal{G}$. For some $K \in \{1, 2, \ldots, \infty\}$, suppose that*

$$w := (v_0 \xrightarrow{\ell_0} v_1 \xrightarrow{\ell_1} \cdots \xrightarrow{\ell_{(K-1)}} v_K)$$

is a well-formed walk through $\mathcal{G}$ with $\ell_0 = \textsc{j}$ and if $K < \infty$, then $\ell_{(K-1)} = \textsc{j}$. Then, there exists a solution φ to $\mathcal{H}$ such that w is the $\mathcal{G}$-simulation of φ.

Proof. Let $\tilde{J}$ be the total number of jump arrows in w. For each finite $\tilde{\jmath} \in \{0, 1, \ldots, \tilde{J}\}$, let $K_{\tilde{\jmath}}$ be the index of the vertex in w immediately after $\tilde{\jmath}$-many jump arrows. That is, $K_j \in \mathbb{N}$ is the smallest number such that there are $\tilde{\jmath}$ jump labels in $\{\ell_0, \ell_1, \ldots, \ell_{K_{\tilde{\jmath}}}\}$.

For each finite $k \in \{0, 1, 2, \ldots, K\}$, let $(q_k, s_k) := v_k$. We will construct a sequence $\{\varphi_{\tilde{\jmath}}\}_{\tilde{\jmath}=1}^{\tilde{J}}$ of hybrid arcs in the form

$$(t, j) \mapsto \varphi_{\tilde{\jmath}}(t, j) = \big(p_{\tilde{\jmath}}(j),\, z_{\tilde{\jmath}}(t, j)\big), \tag{6.26}$$

where $\operatorname{dom}(\varphi_{\tilde{\jmath}})$ and $z_{\tilde{\jmath}}$ are defined below, and $j \mapsto p_{\tilde{\jmath}}(j) := q_{K_j}$ for each $j \in \{0, 1, \ldots, \tilde{\jmath}\}$. By induction, we will show that for each $\tilde{\jmath} \in \{1, 2, \ldots, \tilde{J}\}$,

(S1) if $\tilde{\jmath} > 1$, then $\varphi_{\tilde{\jmath}}$ is an extension of $\varphi_{\tilde{\jmath}-1}$,

(S2) $\varphi_{\tilde{\jmath}}$ is a solution to $\mathcal{H}$ that jumps $\tilde{\jmath}$ times (i.e., $\sup_j \operatorname{dom}(\varphi_{\tilde{\jmath}}) = \tilde{\jmath}$),

(S3) $\operatorname{nrv}\big(z_{\tilde{\jmath}}(T_{\tilde{\jmath}}, \tilde{\jmath})\big) = s_{K_{\tilde{\jmath}}}$, where $T_{\tilde{\jmath}} := \sup_t \operatorname{dom}(\varphi_{\tilde{\jmath}})$ is finite (and $\tilde{\jmath} = \sup_j \operatorname{dom}(\varphi_{\tilde{\jmath}})$),

(S4) $w_{\tilde{\jmath}} := (v_0 \xrightarrow{\ell_0} v_1 \xrightarrow{\ell_1} \cdots \xrightarrow{\ell_{(K_{\tilde{\jmath}}-1)}} v_{K_{\tilde{\jmath}}})$ is the $\mathcal{G}$-simulation of $\varphi_{\tilde{\jmath}}$.

The following definition is used to construct $\varphi_{\tilde{\jmath}}$. For each $k \in \{0, 1, \ldots, K-1\}$ such that $\ell_k = \textsc{f}$, take $\tau_k > 0$ and $\xi_k : [0, \tau_k] \to \mathbb{R}^n$ that satisfy the flow arrow conditions in (6.17).

For the base case ($\tilde{\jmath} = 1$), let $\operatorname{dom}(\varphi_1) := \{(0,0), (0,1)\}$, $z_1(0,0) := s_0$, and $z_1(0,1) := A_{e_0} s_0$. Hence, $\varphi_1(0,0) = (q_0, s_0) = v_0$ and $\varphi_1(0,1) = \big(q_1, A_{e_0} s_0\big)$. Condition (S1) is vacuously satisfied because $\tilde{\jmath} = 1$. Since $v_0 \xrightarrow{\textsc{j}} v_1$ is a jump arrow, $\varphi_1(0,0)$ is in D. Thus, φ_1 is a solution to $\mathcal{H}$ with one jump—satisfying (S2)—because $\operatorname{dom}(\varphi_1)$ has no intervals of flow and satisfies (1.5) at $t_1 = 0$, the only jump time in $\operatorname{dom}(\varphi_1)$. Additionally, since $(q_0, s_0) \xrightarrow{\textsc{j}} (q_1, s_1)$ is a jump arrow, (6.15) requires that $s_1 = \operatorname{nrv}(A_{e_0} s_0)$. Thus, $\operatorname{nrv}(z_1(T_1, J_1)) = \operatorname{nrv}(z_1(0,1)) = s_1 = s_{K_1}$, satisfying (S3). The walk $w_1 = v_0 \xrightarrow{\textsc{j}} v_1$ is the $\mathcal{G}$-simulation of φ_1 with $h_0 = (0,0)$ and $h_1 = (0,1)$, as defined in Definition 6.5, thus (S4) is satisfied, finishing the proof that the base case satisfies (S1)–(S4).

For the inductive case, take any $\tilde{\jmath} \in \{1, 2, \ldots, \tilde{J} - 1\}$ and suppose that $\varphi_{\tilde{\jmath}}$ is a hybrid arc that satisfies (S1)–(S4). We define $\varphi_{\tilde{\jmath}+1}$ as an extension of $\varphi_{\tilde{\jmath}}$, i.e., $\operatorname{dom}(\varphi_{\tilde{\jmath}}) \subset \operatorname{dom}(\varphi_{\tilde{\jmath}+1})$ and $\varphi_{\tilde{\jmath}+1}(t, j) := \varphi_{\tilde{\jmath}}(t, j)$ for all $(t, j) \in \operatorname{dom}(\varphi_{\tilde{\jmath}})$, so (S1) holds by construction. We define $\varphi_{\tilde{\jmath}+1}$ beyond the domain of $\varphi_{\tilde{\jmath}}$ via three cases. In

each case, we will define $k^{\ominus}$ and $k^{\oplus}$ and, for Cases 2 and 3, we also define $k^{(0)}$ and $k^{(\mathsf{f})}$. For the given definitions of $k^{\ominus}$, $k^{\oplus}$, $k^{(0)}$, and $k^{(\mathsf{f})}$, let

$$v^{\circledast} := v_{k^{\circledast}},\ s^{\circledast} := s_{k^{\circledast}},\ \text{and } q^{\circledast} := q_{k^{\circledast}} \text{ for each } \circledast \in \{\ominus, \oplus, (0), (\mathsf{f})\},$$

and $e := (q_{k^{\ominus}}, q_{k^{\oplus}})$.

Case 1 (jump arrow). Suppose $\ell_{K_{\tilde{\jmath}}}$ is a jump label. There are, $\tilde{\jmath}$-many jump arrows from v_0 to $v_{K_{\tilde{\jmath}}}$ (by the definition of $K_{\tilde{\jmath}}$) and it takes one additional step $v_{K_{\tilde{\jmath}}} \xrightarrow{\text{J}} v_{K_{\tilde{\jmath}}+1}$ for the walk $w_{(\tilde{\jmath}+1)}$ to contain $\tilde{\jmath}+1$ jump arrows, because $\ell_{K_{\tilde{\jmath}}} = \text{J}$, so $K_{(\tilde{\jmath}+1)} = K_{\tilde{\jmath}} + 1$. Let $k^{\ominus} := K_{\tilde{\jmath}}$ and $k^{\oplus} := K_{(\tilde{\jmath}+1)}$, We also define $r_0 := |z_{\tilde{\jmath}}(T_{\tilde{\jmath}}, \tilde{\jmath})|$ and $r_{\text{J}} := |A_e s^{(0)}| \in \mathcal{W}(v^{\ominus} \xrightarrow{\text{J}} v^{\oplus})$. Let

$$\text{dom}(\varphi_{\tilde{\jmath}+1}) := \text{dom}(\varphi_{\tilde{\jmath}}) \cup (\{T_{\tilde{\jmath}}\} \times \{\tilde{\jmath}, \tilde{\jmath}+1\}),$$

and

$$z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}}, \tilde{\jmath}+1) := r_0 A_e s^{\ominus}.$$

Since $v^{\ominus} \xrightarrow{\text{J}} v^{\oplus}$ is a jump arrow in $\mathcal{A}$, we have that $(q^{\ominus}, s^{\ominus}) \in D$. By property (6.3) of the nrv function and (S3) from the inductive hypothesis,

$$z_{\tilde{\jmath}}(T_{\tilde{\jmath}}, \tilde{\jmath}) = |z_{\tilde{\jmath}}(T_{\tilde{\jmath}}, \tilde{\jmath})| \,\text{nrv}(z_{\tilde{\jmath}}(T_{\tilde{\jmath}}, \tilde{\jmath})) = r_0 s^{\ominus}.$$

Since D_e is a cone containing $s^{\ominus}$, we have that $z_{\tilde{\jmath}}(T_{\tilde{\jmath}}, \tilde{\jmath}) \in D_e$ and thus $\varphi_{\tilde{\jmath}}(T_{\tilde{\jmath}}, \tilde{\jmath})$ is in D. Additionally,

$$\varphi_{\tilde{\jmath}+1}(T_{\tilde{\jmath}}, \tilde{\jmath}+1) = \big(q^{\oplus}, A_e(r_0 s^{\ominus})\big) \in G(\varphi_{\tilde{\jmath}+1}(T_{\tilde{\jmath}}, \tilde{\jmath})).$$

Therefore, $\varphi_{\tilde{\jmath}+1}$ is a solution that jumps $\tilde{\jmath}+1$ times, thereby satisfying (S2). At the end of $\varphi_{\tilde{\jmath}+1}$, we have $z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}+1) = r_0 A_e s^{\ominus}$. By the definition of a jump arrow in (6.15), $s^{\oplus} = \text{nrv}(A_e s^{\ominus})$. Furthermore, $s^{\oplus} = \text{nrv}(r_0 A_e s^{\ominus}) = \text{nrv}(z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}+1))$ because $r_0 \geq 0$ with $r_0 = 0$ if and only if $s^{\ominus} = 0_n$, in which case $A_e s^{\ominus} = 0_n$, also. Thus, (S3) is satisfied.

By (S4) in the inductive hypothesis, $w_{\tilde{\jmath}}$ is the CTG-simulation of $\varphi_{\tilde{\jmath}}$, so (6.22) and (6.23) are satisfied up to $K_{\tilde{\jmath}}$ and $K_{\tilde{\jmath}} - 1$, respectively. For $\tilde{\jmath}+1$, the hybrid times $h_0, h_1, \ldots, h_{K_{\tilde{\jmath}}}$ defined in Definition 6.5 are the same as for $\tilde{\jmath}$ and $h_{K_{\tilde{\jmath}+1}} = (T_{\tilde{\jmath}+1}, \tilde{\jmath}+1)$. Using (S3), we find that

$$v^{\oplus} = \big(q^{\oplus}, s^{\oplus}\big) = \Big(p_{\tilde{\jmath}+1}(\tilde{\jmath}+1), \text{nrv}\big(z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}+1)\big)\Big),$$

satisfying (6.22) for $k = K_{\tilde{\jmath}+1}$. Finally, $\pi_{\text{J}}\big(h_{K_{\tilde{\jmath}+1}}\big) > \pi_{\text{J}}\big(h_{K_{\tilde{\jmath}}}\big)$, so (6.23) is satisfied for $k = K_{\tilde{\jmath}+1} - 1$. Therefore, $w_{\tilde{\jmath}+1}$ is the CTG-simulation of $\varphi_{\tilde{\jmath}+1}$, as required by (S4).

Case 2 (flow arrow in mode with linear flows). Suppose $\ell_{K_{\tilde{\jmath}}} = \text{F}$ and $q_{K_{\tilde{\jmath}}}$ is a mode with linear flows. Since $\ell_{K_{\tilde{\jmath}}} = \text{F}$ and w is well-formed, $\ell_{K_{\tilde{\jmath}+1}}$ is a jump label, so $K_{(\tilde{\jmath}+1)} = K_{\tilde{\jmath}} + 2$. Let $k^{(0)} := K_{\tilde{\jmath}}$, $k^{(\text{f})} := K_{\tilde{\jmath}} + 1$, $k^{\ominus} := K_{\tilde{\jmath}} + 1$, and $k^{\oplus} := K_{\tilde{\jmath}}+2$. From the definition of flow arrows, take $\tau > 0$ and $\xi : [0, \tau] \to \mathbb{R}^n$ that satisfy (6.17) for $v^{(0)} \xrightarrow{\text{F}} v^{(\text{f})}$. We also define $q := q^{(0)} = q^{(\text{f})}$, $r_0 := |z_{\tilde{\jmath}}(T_{\tilde{\jmath}}, \tilde{\jmath})|$, $r_{\text{F}} := |\xi(\tau)| \in \mathcal{W}(v^{(0)} \xrightarrow{\text{F}} v^{(\text{f})})$, and $r_{\text{J}} := |A_e s^{\ominus}| \in \mathcal{W}(v^{(0)} \xrightarrow{\text{J}} v^{(\text{f})})$. The extension of $\text{dom}(\varphi_{\tilde{\jmath}})$ is defined as

$$\text{dom}(\varphi_{\tilde{\jmath}+1}) := \text{dom}(\varphi_{\tilde{\jmath}}) \cup ([T_{\tilde{\jmath}}, T_{\tilde{\jmath}} + \tau] \times \{\tilde{\jmath}, \tilde{\jmath} + 1\}).$$

Thus, $T_{\tilde{\jmath}+1} = T_{\tilde{\jmath}} + \tau$. The values of $z_{\tilde{\jmath}+1}$ are defined for $(t, j) \in \text{dom}(\varphi_{\tilde{\jmath}+1}) \setminus \text{dom}(\varphi_{\tilde{\jmath}})$ as

$$z_{\tilde{\jmath}+1}(t, \tilde{\jmath}) := r_0 \xi(t - T_{\tilde{\jmath}}) \quad \forall t \in [T_{\tilde{\jmath}}, T_{\tilde{\jmath}+1}] \tag{6.27a}$$

$$z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath} + 1) := r_0 r_{\text{F}} A_e s^{\ominus}. \tag{6.27b}$$

Since $\xi(t) \in C_q$ for all $t \in (0, \tau)$ and C_q is a cone, $z_{\tilde{\jmath}+1}(t, \tilde{\jmath})$ is also in C_q for all $t \in (T_{\tilde{\jmath}}, T_{\tilde{\jmath}+1})$, satisfying (1.6a). Furthermore, the hybrid arc $\varphi_{\tilde{\jmath}+1}$ satisfies the flow condition (1.6b) for all $t \in (T_{\tilde{\jmath}}, T_{\tilde{\jmath}+1})$:

$$\begin{aligned}\frac{dz_{\tilde{\jmath}+1}}{dt}(t, j) &= \frac{d}{dt}\big(r_0 \xi\big(t - T_{\tilde{\jmath}}\big)\big) = r_0 f_q(\xi(t - T_{\tilde{\jmath}})) = r_0 A_q \xi(t - T_{\tilde{\jmath}}) \\ &= A_q r_0 \xi(t - T_{\tilde{\jmath}}) = f_q(z_{\tilde{\jmath}+1}(t, j)).\end{aligned}$$

By the definition of flow arrows, namely (6.17d), $\text{nrv}(\xi(\tau)) = s^{(\text{f})}$, so at the end of the interval of flow $(T_{\tilde{\jmath}}, T_{\tilde{\jmath}+1})$, we have $z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}) = r_0 \xi(\tau) \in D_e$. Thus, $\varphi_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}) \in D$, satisfying (1.5a). Furthermore, (1.5b) is satisfied because $\varphi_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}) \in G(\varphi_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}))$ because

$$\begin{aligned}z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath} + 1) &= r_0 r_{\text{F}} A_e s^{\ominus} = A_e(r_0 r_f s^{\ominus}) = A_e(r_0 |\xi(\tau)| \,\text{nrv}(\xi(\tau))) \\ &= A_e(r_0 \xi(\tau)) = A_e z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}).\end{aligned} \tag{6.28}$$

Therefore, $\varphi_{\tilde{\jmath}+1}$ is a solution to $\mathcal{H}$ that jumps one more time than $\varphi_{\tilde{\jmath}}$, satisfying (S2).

Let $z_0 := z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}}, J_{\tilde{\jmath}})$ and $z_f := z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, J_{\tilde{\jmath}+1})$. We want to show $\mathrm{nrv}(z_f) = s^{\oplus} = s_{K_{\tilde{\jmath}}+1}$. By definition, in (6.28), $z_f = r_0 r_{\mathrm{F}} A_e s^{\ominus}$. If $r_0 r_{\mathrm{F}} > 0$, then we have that $\mathrm{nrv}(z_f) = s^{\oplus}$, per (6.4), since $s^{\oplus} = \mathrm{nrv}(A_e s^{\ominus})$.

On the other hand, if $r_0 = 0$, then $s^{(\mathrm{f})} = s^{\ominus} = z_0 = 0_n$ and, because mode q has linear flows, solutions cannot flow away from the origin, so $r_{\mathrm{F}} = 0$, $z_f = 0_n$, and $s^{(\mathrm{f})} = s^{\ominus} = 0_n$, so $s^{\oplus} = \mathrm{nrv}(A_e s^{\ominus}) = \mathrm{nrv}(0_n) = 0_n$. Thus, $\mathrm{nrv}(z_f) = s^{\oplus}$.

Finally, if $r_{\mathrm{F}} = 0$, then $s^{(\mathrm{f})} = s^{\ominus} = 0_n$, because $|\xi(\tau)| = 0$. Since $s^{\oplus} = \mathrm{nrv}(A_e s^{\ominus}) = 0_n$, we have that

$$\mathrm{nrv}(z_f) = s^{\oplus} = 0_n.$$

Next, we want to show (S4), i.e., that $w_{\tilde{\jmath}+1}$ is the $\mathcal{G}$-simulation of $\varphi_{\tilde{\jmath}+1}$, which requires showing (6.22) holds for $k \in \{K_{\tilde{\jmath}} + 1, K_{\tilde{\jmath}} + 2\}$, and (6.23) holds for $k \in \{K_{\tilde{\jmath}}, K_{\tilde{\jmath}} + 1\}$. By assumption, $w_{\tilde{\jmath}}$ is the $\mathcal{G}$-simulation of $z_{\tilde{\jmath}}$. For $z_{\tilde{\jmath}+1}$, the sequence $h_0, h_1, \ldots, h_{K_{(\tilde{\jmath}+1)}}$ defined in Definition 6.5 has two more elements than the corresponding sequence for $z_{\tilde{\jmath}}$, namely $h_{(K_{\tilde{\jmath}}+1)} = (T_{\tilde{\jmath}+1}, \tilde{\jmath})$ and $h_{(K_{\tilde{\jmath}}+2)} = h_{K_{(\tilde{\jmath}+1)}} = (T_{\tilde{\jmath}+1}, \tilde{\jmath} + 1)$.

First, we will show that $w_{\tilde{\jmath}+1}$ satisfies (6.22) for $k = K_{\tilde{\jmath}} + 1$. We have that

$$p_{\tilde{\jmath}+1}(h_{K_{\tilde{\jmath}}+1}) = p_{\tilde{\jmath}+1}(\tilde{\jmath}) = q_{K_{\tilde{\jmath}}}$$

because $p_{\tilde{\jmath}+1}(j) = q_{K_j}$ by definition for each $j \in \{0, 1, \ldots, \tilde{\jmath} + 1\}$. But,

$$p_{\tilde{\jmath}+1}(h_{K_{\tilde{\jmath}}+1}) = q_{K_{\tilde{\jmath}}+1},$$

because $q = q^{(0)} = q_{K_{\tilde{\jmath}}} = q_{K_{\tilde{\jmath}}+1} = q^{(\mathrm{f})}$, as required by (6.22) for $k = K_{\tilde{\jmath}} + 1$. For the z-component,

$$z_{\tilde{\jmath}+1}(h_{K_{\tilde{\jmath}}+1}) = z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}) = r_0 \xi(\tau).$$

Suppose, first, that $r_0 > 0$. Then, by (6.4), Suppose, instead, that $r_0 = 0$. In this case, ξ is identically zero because $t \mapsto \xi(t) := 0_n$ is the unique solution to $\dot{x} = A_q x$ from $x_0 = 0_n$. Thus, $\mathrm{nrv}(r_0 \xi(\tau)) = \mathrm{nrv}(\xi(\tau)) = 0_n$. By (6.17d), $\mathrm{nrv}(\xi(\tau)) = s^{(\mathrm{f})}$, so

$$z_{\tilde{\jmath}+1}(h_{K_{\tilde{\jmath}}+1}) = s_{K_{\tilde{\jmath}}+1},$$

therefore (6.22) is satisfied for $k = K_{\tilde{\jmath}} + 1$.

Next, we will show that $w_{\tilde{\jmath}+1}$ satisfies (6.22) for $k = K_{\tilde{\jmath}} + 2 = K_{(\tilde{\jmath}+1)}$. We have that

$$p_{\tilde{\jmath}+1}(h_{K_{(\tilde{\jmath}+1)}}) = p_{\tilde{\jmath}+1}(\tilde{\jmath}+1) = q_{K_{(\tilde{\jmath}+1)}},$$

as required by (6.22) for $k = K_{\tilde{\jmath}} + 1$. For the z-component,

$$z_{\tilde{\jmath}+1}(h_{K_{(\tilde{\jmath}+1)}}) = z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}+1).$$

We have already shown that (S3) holds, so $\mathrm{nrv}(z_{\tilde{\jmath}+1}(h_{K_{(\tilde{\jmath}+1)}})) = s_{K_{(\tilde{\jmath}+1)}}$. Therefore, (6.22) holds for $k = K_{(\tilde{\jmath}+1)}$.

Finally, (6.23) is satisfied for $k = K_{\tilde{\jmath}}$ because $\ell_{K_{\tilde{\jmath}}} = \textsc{f}$ and

$$\pi_{\textsc{t}}\big(h_{K_{\tilde{\jmath}}+1}\big) > \pi_{\textsc{t}}\big(h_{K_{\tilde{\jmath}}}\big),$$

and is satisfied for $k = K_{\tilde{\jmath}} + 1$ because $\ell_{K_{\tilde{\jmath}}+1} = \textsc{j}$ and

$$\pi_{\textsc{j}}\big(h_{K_{\tilde{\jmath}+1}}\big) > \pi_{\textsc{j}}\big(h_{K_{\tilde{\jmath}}+1}\big).$$

Therefore, $w_{\tilde{\jmath}+1}$ is the $\mathcal{G}$-simulation of $\varphi_{\tilde{\jmath}+1}$, satisfying (S4).

Case 3 (flow arrow in mode with constant flows). Suppose $\ell_{K_{\tilde{\jmath}}} = \textsc{f}$ and $q_{K_{\tilde{\jmath}}}$ is a mode with constant flows. Since w is well-formed, $\ell_{K_{\tilde{\jmath}}+1} = \textsc{j}$ and $K_{(\tilde{\jmath}+1)} = K_{\tilde{\jmath}} + 2$. Let $k^{(0)} := K_{\tilde{\jmath}}$, $k^{(\mathrm{f})} := K_{\tilde{\jmath}} + 1$, $k^{\ominus} := K_{\tilde{\jmath}} + 1$, and $k^{\oplus} := K_{\tilde{\jmath}} + 2$. From the definition of flow arrows, take $\tau > 0$ and $\xi : [0, \tau] \to \mathbb{R}^n$ that satisfy (6.17) for $v^{(0)} \xrightarrow{\textsc{f}} v^{(\mathrm{f})}$. We also define $q := q^{(0)} = q^{(\mathrm{f})}$, $r_0 := |z_{\tilde{\jmath}}(T_{\tilde{\jmath}}, \tilde{\jmath})|$, $r_{\textsc{f}} := |\xi(\tau)| \in \mathcal{W}(v^{(0)} \xrightarrow{\textsc{f}} v^{(\mathrm{f})})$, and $r_{\textsc{j}} := |A_e s^{\ominus}| \in \mathcal{W}(v^{(0)} \xrightarrow{\textsc{j}} v^{(\mathrm{f})})$. Let

$$\mathrm{dom}(\varphi_{\tilde{\jmath}+1}) := \mathrm{dom}(\varphi_{\tilde{\jmath}}) \cup \big([T_{\tilde{\jmath}}, T_{\tilde{\jmath}+1}] \times \{\tilde{\jmath}, \tilde{\jmath}+1\}\big),$$

where

$$T_{\tilde{\jmath}+1} := T_{\tilde{\jmath}} + \begin{cases} r_0\tau & \text{if } r_0 > 0 \\ \tau & \text{if } r_0 = 0. \end{cases}$$

The value of $z_{\tilde{\jmath}+1}(t, j)$ is defined for $(t, j) \in \mathrm{dom}(\varphi_{\tilde{\jmath}+1}) \setminus \mathrm{dom}(\varphi_{\tilde{\jmath}})$ as

$$z_{\tilde{\jmath}+1}(t, \tilde{\jmath}) := \begin{cases} r_0\xi\big((t - T_{\tilde{\jmath}})/r_0\big) & \text{if } r_0 > 0 \\ \xi\big(t - T_{\tilde{\jmath}}\big) & \text{if } r_0 = 0 \end{cases} \quad \forall t \in [T_{\tilde{\jmath}}, T_{\tilde{\jmath}+1}] \tag{6.29a}$$

$$z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}+1) := \begin{cases} r_0 r_{\textsc{f}} A_e s^{\ominus} & \text{if } r_0 > 0 \\ r_{\textsc{f}} A_e s^{\ominus} & \text{if } r_0 = 0. \end{cases} \tag{6.29b}$$

To show (S2), we will consider separately the cases $r_0 > 0$ and $r_0 = 0$. Suppose, first, that $r_0 > 0$. Then, $\varphi_{\tilde{\jmath}+1}$ satisfies the flow condition (1.6b) for all $t \in (T_{\tilde{\jmath}}, T_{\tilde{\jmath}+1})$:

$$\begin{aligned}\dot{z}_{\tilde{\jmath}+1}(t,\tilde{\jmath}) &= \frac{d}{dt}\big(r_0\xi\big((t-T_{\tilde{\jmath}})/r_0\big)\big) \\ &= r_0 f_q\big(\xi\big((t-T_{\tilde{\jmath}})/r_0\big)\big)\frac{d}{dt}\big((t-T_{\tilde{\jmath}})/r_0\big) \\ &= f_q(z_{\tilde{\jmath}+1}(t,\tilde{\jmath})).\end{aligned}$$

If, instead, $r_0 = 0$, then,

$$\dot{z}_{\tilde{\jmath}+1}(t,\tilde{\jmath}) = \frac{d}{dt}\big(\xi\big(t-T_{\tilde{\jmath}}\big)\big) = f_q\big(z_{\tilde{\jmath}+1}(t,\tilde{\jmath})\big),$$

again satisfying (1.6b). In both cases, $z_{\tilde{\jmath}+1}(t,\tilde{\jmath}) \in C_q$ for all $t \in (T_{\tilde{\jmath}}, T_{\tilde{\jmath}+1})$, satisfying (1.6a).

By (6.17d) in the definition of flow arrows, $\mathrm{nrv}(\xi(\tau)) = s^{(\mathsf{f})} \in D_e$, so

$$z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1},\tilde{\jmath}) \in D_e.$$

Thus, $\varphi_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1},\tilde{\jmath}) \in D$, satisfying (1.5a). Furthermore, (1.5b) is satisfied at the only jump time, $T_{\tilde{\jmath}+1}$, in $\mathrm{dom}(\varphi_{\tilde{\jmath}+1}) \setminus \mathrm{dom}(\varphi_{\tilde{\jmath}})$ because $z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1},\tilde{\jmath}+1) = A_e z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1},\tilde{\jmath})$. Therefore, $\varphi_{\tilde{\jmath}+1}$ is a solution to $\mathcal{H}$ that jumps one more time than $\varphi_{\tilde{\jmath}}$, satisfying (S2).

Let $z_0 := z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}}, J_{\tilde{\jmath}})$ and $z_f := z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, J_{\tilde{\jmath}+1})$. To prove (S3) holds, we must show $\mathrm{nrv}(z_f) = s^{\oplus} = s_{K_{\tilde{\jmath}}+1}$. From (6.29b), $\mathrm{nrv}(z_f) = \mathrm{nrv}(A_e s^{\ominus})$, which equals $s^{\oplus}$, satisfying (S3).

Next, we want to show (S4), i.e., that $w_{\tilde{\jmath}+1}$ is the $\mathcal{G}$-simulation of $\varphi_{\tilde{\jmath}+1}$, which requires showing (6.22) holds for $k \in \{K_{\tilde{\jmath}}+1, K_{\tilde{\jmath}}+2\}$, and (6.23) holds for $k \in \{K_{\tilde{\jmath}}, K_{\tilde{\jmath}}+1\}$. By assumption, $w_{\tilde{\jmath}}$ is the $\mathcal{G}$-simulation of $z_{\tilde{\jmath}}$. For $z_{\tilde{\jmath}+1}$, the sequence $h_0, h_1, \ldots, h_{K_{(\tilde{\jmath}+1)}}$ defined in Definition 6.5 has two more elements than the corresponding sequence for $z_{\tilde{\jmath}}$, namely $h_{(K_{\tilde{\jmath}}+1)} = (T_{\tilde{\jmath}+1}, \tilde{\jmath})$ and $h_{(K_{\tilde{\jmath}}+2)} = h_{K_{(\tilde{\jmath}+1)}} = (T_{\tilde{\jmath}+1}, \tilde{\jmath}+1)$. First, we will show that $w_{\tilde{\jmath}+1}$ satisfies (6.22) for $k = K_{\tilde{\jmath}}+1$. We have that

$$p_{\tilde{\jmath}+1}(h_{K_{\tilde{\jmath}}+1}) = p_{\tilde{\jmath}+1}(\tilde{\jmath}) = q_{K_{\tilde{\jmath}}}$$

because $p_{\tilde{\jmath}+1}(j) = q_{K_j}$ by definition for each $j \in \{0, 1, \ldots, \tilde{\jmath}+1\}$. But,

$$p_{\tilde{\jmath}+1}(h_{K_{\tilde{\jmath}}+1}) = q_{K_{\tilde{\jmath}}+1},$$

because $q = q^{(0)} = q_{K_{\tilde{\jmath}}} = q_{K_{\tilde{\jmath}}+1} = q^{(\mathrm{f})}$, thereby satisfying (6.22) for $k = K_{\tilde{\jmath}} + 1$. For the z-component,

$$z_{\tilde{\jmath}+1}(h_{K_{\tilde{\jmath}}+1}) = z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}) = \begin{cases} r_0\xi(\tau) & \text{if } r_0 > 0 \\ \xi(\tau) & \text{if } r_0 = 0 \end{cases}$$

Thus, $\mathrm{nrv}(z_{\tilde{\jmath}+1}) = \mathrm{nrv}(\xi(\tau))$. By (6.17d), $\mathrm{nrv}(\xi(\tau)) = s^{(\mathrm{f})}$, so

$$z_{\tilde{\jmath}+1}(h_{K_{\tilde{\jmath}}+1}) = s_{K_{\tilde{\jmath}}+1},$$

therefore (6.22) is satisfied for $k = K_{\tilde{\jmath}} + 1$.

Next, we will show that $w_{\tilde{\jmath}+1}$ satisfies (6.22) for $k = K_{\tilde{\jmath}} + 2 = K_{(\tilde{\jmath}+1)}$. We have that

$$p_{\tilde{\jmath}+1}(h_{K_{(\tilde{\jmath}+1)}}) = p_{\tilde{\jmath}+1}(\tilde{\jmath}+1) = q_{K_{(\tilde{\jmath}+1)}},$$

as required by (6.22) for $k = K_{\tilde{\jmath}} + 1$. For the z-component,

$$z_{\tilde{\jmath}+1}(h_{K_{(\tilde{\jmath}+1)}}) = z_{\tilde{\jmath}+1}(T_{\tilde{\jmath}+1}, \tilde{\jmath}+1).$$

We have already shown that (S3) holds, so $\mathrm{nrv}(z_{\tilde{\jmath}+1}(h_{K_{(\tilde{\jmath}+1)}})) = s_{K_{(\tilde{\jmath}+1)}}$. Therefore, (6.22) holds for $k = K_{(\tilde{\jmath}+1)}$.

Finally, (6.23) is satisfied for $k = K_{\tilde{\jmath}}$ because $\ell_{K_{\tilde{\jmath}}} = \mathrm{F}$ and

$$\pi_{\mathrm{T}}\big(h_{K_{\tilde{\jmath}}+1}\big) > \pi_{\mathrm{T}}\big(h_{K_{\tilde{\jmath}}}\big),$$

and is satisfied for $k = K_{\tilde{\jmath}} + 1$ because $\ell_{K_{\tilde{\jmath}}+1} = \mathrm{J}$ and

$$\pi_{\mathrm{J}}\big(h_{K_{\tilde{\jmath}+1}}\big) > \pi_{\mathrm{J}}\big(h_{K_{\tilde{\jmath}}+1}\big).$$

Therefore, $w_{\tilde{\jmath}+1}$ is the $\mathcal{G}$-simulation of $\varphi_{\tilde{\jmath}+1}$, satisfying (S4).

Thus, for the inductive case, $\varphi_{\tilde{\jmath}+1}$ is a hybrid arc that satisfies (S1)–(S4). Therefore, by induction, $w_J = w$ is the $\mathcal{G}$-simulation of $\varphi_J = \varphi$. □

The following result asserts that the weight of a solution φ's CTG-simulation contains the relative change in the distance of φ from $\mathcal{Q} \times \{0_n\}$. In other words, the weights of CTG-simulations tell us how solutions move toward or away from $\mathcal{Q} \times \{0_n\}$.

Proposition 6.5. *Consider a conical hybrid system $\mathcal{H}$ with conical transition graph $\mathcal{G}$ and a solution*

$$(t, j) \mapsto \varphi(t, j) = \big(q(j), z(t, j)\big).$$

Suppose φ jumps at least once and let

$$w := (v_0 \xrightarrow{\ell_0} v_1 \xrightarrow{\ell_1} \cdots \xrightarrow{\ell_{(K_J-1)}} v_{K_J})$$

be the $\mathcal{G}$-simulation of φ, with $J := \sup_j \mathrm{dom}(\varphi)$ and with $K_0, K_1, \ldots, K_J$ defined as in Definition 6.5. Furthermore, let h_0, h_1, ..., h_{K_J} be the hybrid times associated with the $\mathcal{G}$-simulation of φ, as defined in Definition 6.5, and for each finite $k \in \{0, 1, \ldots, K_J\}$, let

$$w_k := (v_0 \xrightarrow{\ell_0} v_1 \xrightarrow{\ell_1} \cdots \xrightarrow{\ell_{(k-1)}} v_k)$$

be the truncation of w to the first k arrows.

Suppose that there does not exist a flow arrow $(q, 0_n) \xrightarrow{\mathrm{F}} (q, s^{(f)})$ in w such that $s^{(f)} \neq 0_n$. Then, for each finite $k \in \{1, 2, \ldots, K_J\}$,

$$|z(h_k)| = r_k |z(h_0)| \text{ for some } r_k \in \mathcal{W}(w_k). \tag{6.30}$$

Proof. For each $k \in \{0, 1, \ldots, K_J\}$, let $(q_k, s_k) := v_k$, and if $\ell_k = \textsc{j}$, then let $e_k := (q_k, q_{k+1})$. Let $r_0 := |z(h_0)|$.

We proceed by induction over k. For the base case, consider $k = 1$. The first arrow in a CTG-simulation is always a jump arrow. Let $r_1 := |A_{e_0} s_0| \in \mathcal{W}(v_0 \xrightarrow{\mathrm{J}} v_1)$. Since $z(t_1, 1) = A_{e_0} z(h_0)$ and $z(h_0) = r_0 s_0$, we have

$$|z(h_1)| = |A_{e_0} r_0 s_0| = r_0 r_1 = r_1 |z(h_0)|.$$

Therefore, (6.30) holds for $k = 1$, proving the base case.

Now, suppose that (6.30) holds for $k \in \{1, \ldots, K_J - 1\}$. That is, there exists $r_k \in \mathcal{W}(w_k)$ such that $|z(h_k)| = r_k |z(h_0)|$.

Suppose, first, that $\ell_k = \textsc{j}$. Let $r'_{k+1} := |A_{e_k} s_k| \in \mathcal{W}(v_k \xrightarrow{\mathrm{J}} v_{k+1})$. Thus, $r_{k+1} := r_k r'_{k+1} \in \mathcal{W}(w_{k+1})$. Furthermore, $z(h_{k+1}) = A_{e_k} z(h_k)$, so

$$|z(h_{k+1})| = |A_{e_k} z(h_k)| = |A_{e_k} r_0 r_k s_k| = r_0 r_k |A_{e_k} s_k| = r_0 r_k r'_{k+1} = r_0 r_{k+1}.$$

Thus, $|z(h_{k+1})| = r_{k+1} |z(h_0)|$ for $r_{k+1} \in \mathcal{W}(w_{k+1})$.

Alternatively, suppose that $\ell_k = \text{F}$. Let $q := q_k = q_{k+1}$. If $r_k = 0$, then $s_k = 0_n$, so $s_{k+1} = 0_n$ also, by assumption, in which case $f_q(0_n) = 0_n$. Since f_q is Lipschitz continuous, solutions to $\dot{z} = f_q(z)$, $z(0) = 0_n$ are unique, namely $t \mapsto \xi(t) := 0_n$. Thus, it must be the case that $z(h_{k+1}) = 0_n$. From (6.18), we have that $0 \in \mathcal{W}(v_k \xrightarrow{\text{F}} v_{k+1})$, so $r_{k+1} := 0 \in \mathcal{W}(w_{k+1})$. Thus, (6.30) holds:

$$|z(h_{k+1})| = 0 = r_{k+1} r_0 = r_{k+1} |z(h_0)|.$$

Suppose, instead, that $r_k > 0$, which also implies that $r_0 > 0$ (if $r_0 = 0$, then $r_1 = r_2 = \cdots = r_k = 0$). We will define $\tau > 0$ and $\xi : [0, \tau] \to \mathbb{R}^n$ to satisfy the flow arrow conditions in (6.17) for $v_k \xrightarrow{\text{F}} v_{k+1}$. Let $j := \pi_{\text{J}}(h_k) = \pi_{\text{J}}(h_{k+1})$, $t_k := \pi_{\text{T}}(h_k)$, and $t_{k+1} = \pi_{\text{T}}(h_{k+1})$. We define

$$\tau := \begin{cases} t_{k+1} - t_k & \text{if } q \text{ has linear flows} \\ \big(t_{k+1} - t_k\big)/r_0 r_k & \text{if } q \text{ has constant flows,} \end{cases}$$

and for all $t \in [0, \tau]$, let

$$t \mapsto \xi(t) := \begin{cases} z\big(t_k + t,\ j\big)/r_0 r_k & \text{if } q \text{ has linear flows} \\ z\big(t_k + r_0 r_k t,\ j\big)/r_0 r_k & \text{if } q \text{ has constant flows.} \end{cases}$$

By the inductive hypothesis, $r_0 r_k = |z(h_k)|$, so we find that (6.17a) is satisfied:

$$\xi(0) = \frac{1}{r_0 r_k} z(t_k, j) = \frac{z(h_k)}{|z(h_k)|} = \text{nrv}(z(h_k)) = s_k.$$

To check that $\dot{\xi}(t) = f_q(\xi(t))$, that is, (6.17b), we consider constant flows and linear flows separately. If q has linear flows, then for all $t \in (0, \tau)$,

$$\begin{aligned} \dot{\xi}(t) &= \frac{d}{dt}\big(z(t_k + t, j)/r_0 r_k\big) = \frac{1}{r_0 r_k} \dot{z}(t_k + t, j) \\ &= \frac{1}{r_0 r_k} f_q(z(t_k + t, j)) = \frac{1}{r_0 r_k} A_q z(t_k + t, j) \\ &= A_q z(t_k + t, j)/r_0 r_k = A_q \xi(t) \\ &= f_q(\xi(t)). \end{aligned}$$

Alternatively, if q has constant flows, then

$$\begin{aligned} \dot{\xi}(t) &= \frac{d}{dt}\left(\frac{1}{r_0 r_k} z(t_k + r_0 r_k t)\right) \\ &= \dot{z}(t_k + r_0 r_k t) \\ &= f_q\big(z(t_k + r_0 r_k t)\big). \end{aligned}$$

Since f_q is constant, i.e., $f_q(z) = f_q^*$ for all $z \in \mathbb{R}^n$,

$$f_q\big(z(t_k + r_0 r_k t)\big) = f_q^* = f_q(\xi(t)),$$

so $\dot{\xi}(t) = f_q(\xi(t))$. In both cases, (6.17b) is satisfied.

We have that $\xi(t) \in C_q$ for all $t \in [0, \tau]$, C_q is a cone, so $z(t, j) \in C_q$ for all $t \in [t_k, t_{k+1}]$, satisfying (6.17c).

Checking the terminal flow arrow condition (6.17d), we find

$$\begin{aligned}
\xi(\tau) &= \begin{cases} \xi\big(t_{k+1} - t_k\big) & \text{if } q \text{ has linear flows} \\ \xi\big((t_{k+1} - t_k)/r_0 r_k\big) & \text{if } q \text{ has constant flows} \end{cases} \\
&= \begin{cases} z\big(t_k + (t_{k+1} - t_k), j\big)/r_0 r_k & \text{if } q \text{ has linear flows} \\ z\big(t_k + r_0 r_k\,(t_{k+1} - t_k)/r_0 r_k\,, j\big)/r_0 r_k & \text{if } q \text{ has constant flows} \end{cases} \\
&= z\big(t_{k+1}, j\big)/r_0 r_k \\
&= z\big(h_{k+1}\big)/r_0 r_k\,. \qquad (6.31)
\end{aligned}$$

Therefore, (6.17d) holds:

$$\operatorname{nrv}(\xi(\tau)) = \operatorname{nrv}\big(z(h_{k+1})/r_k r_0\big) = \operatorname{nrv}(z(h_{k+1})) = s_{k+1}.$$

Finally, let $r'_{k+1} := |\xi(\tau)| \in \mathcal{W}(v_k \xrightarrow{\mathrm{F}} v_{k+1})$ and $r_{k+1} := r_k r'_{k+1} \mathcal{W}(w_{k+1})$. Rewriting (6.31), we find

$$|z(h_{k+1})| = r_0 r_k |\xi(\tau)| = r_0 r_k r'_{k+1} = r_0 r_{k+1} = r_{k+1} |z(h_0)|.$$

Therefore, (6.30) holds for all $k \in \{1, 2, \ldots, K_J\}$, by induction. □

6.5.2 Stability and Asymptotic Stability

By applying Propositions 6.3–6.5, we can use the CTG of $\mathcal{H}$ to determine pre-asymptotic stability of $\mathcal{O}$. First, in Proposition 6.6, we use the CTG to establish stability, which we use to establish pre-asymptotic stability in Theorem 6.2.

Proposition 6.6. *Let $\mathcal{H} = (C, f, D, g)$ by a conical hybrid system with modes $\mathcal{Q}$ and conical transition graph $\mathcal{G}$. Suppose that $\mathcal{O} := \mathcal{Q} \times \{0_n\}$ is stable for (C, f) and that there exists $M \geq 1$ such that $\sup \mathcal{W}(w) \leq M$ for each walk w through $\mathcal{G}$. Then, $\mathcal{O}$ is stable for $\mathcal{H}$.*

Proof. Take any $\varepsilon > 0$. Since $\mathcal{O}$ is stable for (C, f), there exists $\delta \in (0, \varepsilon)$ such that for every solution $t \mapsto \xi(t)$ to (C, f) with $|\xi(0)| \leq \delta$,

$$|\xi(t)| \leq \varepsilon \quad \forall t \in \mathrm{dom}(\xi).$$

Let $\varepsilon' := \delta/M$. Then, again by the stability of 0_n, there exists $\delta' > 0$ such that, for every solution $t \mapsto \xi(t)$ to (C, f) with $|\xi(0)| \leq \delta'$,

$$|\xi(t)| \leq \varepsilon' \quad \forall t \in \mathrm{dom}(\xi).$$

Let $(t, j) \mapsto \varphi(t, j) := (q(j), z(t, j))$ be any solution to $\mathcal{H}$ with $|z(0, 0)| \leq \delta'$. Thus, $|z(t, 0)| \leq \varepsilon'$ for all $t \in [0, t_1]$, where t_1 is the first jump time in $\mathrm{dom}(\varphi)$. In particular, we will use the fact that

$$|z(t_1, 0)| \leq \varepsilon'.$$

Since $\mathcal{O}$ is stable for (C, f), solutions to $\mathcal{H}$ cannot leave $\mathcal{O}$ by flowing. Furthermore, from the definition of conical hybrid systems, $g(\mathcal{O}) \subset \mathcal{O}$, so solutions to $\mathcal{H}$ cannot jump away from $\mathcal{O}$. Therefore, $\mathcal{O}$ is forward invariant for $\mathcal{H}$.

Let w be the $\mathcal{G}$-simulation of φ with K_1, K_2, …, K_J and h_0, h_1, …, h_{K_J} defined as in Definition 6.5, and let w_k be the truncation of w to the first $k > 0$ steps. By Proposition 6.5, for each jump time t_j in $\mathrm{dom}(\varphi)$, there exists $r_j \in \mathcal{W}(w_{K_j})$ such that

$$|z(t_j, j)| = r_{K_j} |z(t_1, 0)|.$$

Since the weight of every walk is bounded by M and $|z(t_1, 0)| \leq \varepsilon'$,

$$|z(t_j, j)| = r_{K_j} |z(t_1, 0)| \leq M\varepsilon' = \delta.$$

Thus, every interval of flow $[t_j, t_{j+1}]$ starts with $|z(t_j, j)| \leq \delta$, so $|z(t, j)| \leq \varepsilon$ for all $t \in [t_j, t_{j+1}]$. Therefore, $\mathcal{O}$ is stable for $\mathcal{H}$. □

The next theorem is of central importance to this work as it allows one to establish pre-asymptotic stability using the CTG.

Theorem 6.2. *Let $\mathcal{H} = (C, f, D, g)$ be a conical hybrid system with modes $\mathcal{Q}$ and conical transition graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{W})$. Suppose the following:*

(P1) *For each $q \in \mathcal{Q}$, the origin 0_n is pre-asymptotically stable for (C_q, f_q).*

(P2) *There exists $M > 0$ such that every walk w through $\mathcal{G}$ satisfies* $\sup \mathcal{W}(w) \leq M$.

(P3) *Every well-formed infinite-length walk w through $\mathcal{G}$ satisfies* $\mathcal{W}(w) = \{0\}$.

Then, the set $\mathcal{O} := \mathcal{Q} \times \{0_n\}$ is pAS with respect to $\mathcal{H}$.

Proof. Items (P1) and (P2) satisfy the assumptions of Proposition 6.6, which asserts that the origin of $\mathcal{H}$ is stable. By stability and the radial homogeneity property of $\mathcal{H}$ established in Proposition 6.1, every solution is bounded. Thus, we only need to show that every complete solution to $\mathcal{H}$ converges to $\mathcal{O}$. As a consequence of stability, $\mathcal{O}$ is forward invariant and there does not exist any flow arrows in $\mathcal{G}$ in the form $(q, 0_n) \xrightarrow{\text{F}} (q, s^{(\text{f})})$, where $s^{(\text{f})} \neq 0_n$.

Let $(t, j) \mapsto \varphi(t, j) = (q(j), z(t, j))$ be any solution to $\mathcal{H}$, let $J := \sup_j \operatorname{dom}(\varphi)$ and $T := \sup_t \operatorname{dom}(\varphi)$, let $t_0 := 0$, and for each $j \in \{1, 2, \ldots, J\}$, let t_j denote the jth jump time of φ. Showing that φ converges to $\mathcal{O}$ is equivalent to showing z converges to 0_n.

If $J < \infty$, then there are no jumps after t_J, so the function $t \mapsto \varphi(t, J)$ is a solution to (C, f) for all $t \in [t_J, T)$. If φ is complete, then $\lim_{t\to\infty} z(t, J) = 0_n$ due to (P1).

Suppose, instead, that $J = \infty$. Let w be the $\mathcal{G}$-simulation of φ with K_1, K_2, ..., K_J; $h_0, h_1, \ldots, h_{K_J}$; and w_k defined as in Definition 6.5. Then, by Proposition 6.5, for each finite $k \in \{1, 2, \ldots, K_J\}$, there exists $r_k \in \mathcal{W}(w_k)$ such that

$$|z(h_k)| = r_k |z(t_1, 0)| \leq \sup \mathcal{W}(w_k).$$

Per Proposition 6.3, the CTG-simulation of φ is a well-formed walk through $\mathcal{G}$, so by (P3), we have that $\mathcal{W}(w) = \{0\}$. Thus,

$$\lim_{k\to\infty} \sup \mathcal{W}(w_k) = 0 \quad \text{and} \quad \lim_{k\to\infty} |z(t_j, j)| = 0.$$

It remains to be shown that the value of the solution during intervals of flow *between* jump times also converges. By assumption, $\mathcal{O}$ is stable for (C, f), so the assumptions of Lemma 6.1 are satisfied. Thus, by Lemma 6.1, we have that

$$\lim_{t+j\to\infty} z(t, j) = 0_n.$$

Therefore, since every complete solution to $\mathcal{H}$ converges to $\mathcal{O}$, we conclude that $\mathcal{O}$ is pAS for $\mathcal{H}$. □

Remark 6.4. If, additionally, $\check{\mathcal{H}}$ is the conical approximation of a hybrid system $\mathcal{H}$ about $x_* \in \mathbb{R}^n$, then Theorem 6.1 asserts that x_* is (locally) pre-asymptotically stable for $\mathcal{H}$.

The assumptions in Theorem 6.2 can be simplified when $\mathcal{G}$ is finite. When $\mathcal{V}$ is finite, condition (P3) is satisfied if and only if $\sup \mathcal{W}(w) < 1$ for every elementary cycle w in $\mathcal{G}$. A walk through a graph is called an *elementary cycle* if it starts and ends at the same vertex and does not visit any other vertex more than once. To check (P3), it is necessary to enumerate over all of the elementary cycles. One efficient algorithm for this purpose is Johnson's enumeration algorithm [46]. For a CTG with $|\mathcal{V}|$ vertices, $|\mathcal{A}|$ arrows, and c elementary circuits (not counting cyclic permutations), the worst-case time complexity of Johnson's algorithm is $O\big((|\mathcal{V}| + |\mathcal{A}|)(c+1)\big)$. Furthermore, if the weight of each arrow is bounded and $\mathcal{V}$ is finite, then (P3) implies (P2).

6.6 Abstractions to Reduce the Graph Size

A problem that arises when applying CTG–based analysis is that the set of vertices $\mathcal{V}$ is often infinite. In this section, we introduce results that allow for reducing an infinite CTG into a finite graph while preserving relevant properties of the graph. Such a reduction is called an *abstraction*. Previous work has used abstractions to reduce the infinite state space of timed processes [47] and timed hybrid automatons [48] into a finite number of states, allowing for algorithmic analysis.

Our general approach is to cover $\mathbb{S}_0^{n-1}$ with a finite number of sets, which we use as replacements for individual points as vertices in graphs. Given a set S, a *cover* of S is a collection of sets $\{P^i\}_{i\in\mathcal{I}}$ indexed over $\mathcal{I} \subset \mathbb{N}$ such that $P^i \subset S$ for each $i \in \mathcal{I}$, and

$$S = \bigcup_{i\in\mathcal{I}} P^i.$$

Given a conical hybrid system $\mathcal{H} := (C, f, D, G)$ with modes $\mathcal{Q}$, we consider a cover of $\mathbb{S}_0^{n-1}$ for each mode. That is, for each $q \in \mathcal{Q}$, let $\mathcal{P}_q := \{P_q^i\}_{i\in\mathcal{I}_q}$ be a cover of $\mathbb{S}_0^{n-1}$ with index set $\mathcal{I}_q$. We impose that $\mathcal{P}_q$ is a finite collection of sets to allow for computational tractability, we write index sets in the form $\mathcal{I}_q := \{0, 1, \ldots, m\}$

where $m \in \mathbb{N}$. For each $e := (q^{\ominus}, q^{\oplus}) \in \mathcal{E}$, let

$$\mathcal{I}_e^{\ominus} := \left\{ i \in \mathcal{I}_e \;\middle|\; P^i_{q^{\ominus}} \cap D_e \neq \varnothing \right\} \quad \text{and} \quad \mathcal{I}_e^{\oplus} := \left\{ i \in \mathcal{I}_e \;\middle|\; P^i_{q^{\oplus}} \cap (A_e D_e) \neq \varnothing \right\}. \tag{6.32}$$

Thus, $z \in D_e$ and $e := (q^{\ominus}, q^{\oplus}) \in \mathcal{E}$,

$$\exists i^{\ominus} \in \mathcal{I}_e^{\ominus} : z \in P^{i^{\ominus}}_{q^{\ominus}} \quad \text{and} \quad \exists i^{\oplus} \in \mathcal{I}_e^{\oplus} : A_e z \in P^{i^{\oplus}}_{q^{\oplus}}. \tag{6.33}$$

In other words, for each $e := (q^{\ominus}, q^{\oplus}) \in \mathcal{E}$,

$$D_e \subset \bigcup_{i^{\ominus} \in \mathcal{I}_e^{\ominus}} P^{i^{\ominus}}_{q^{\ominus}}, \quad \text{and} \quad A_e D_e \subset \bigcup_{i^{\oplus} \in \mathcal{I}_e^{\oplus}} P^{i^{\oplus}}_{q^{\oplus}}.$$

We then define abstract conical transition graphs as follows.

Definition 6.6 (Abstract Conical Transition Graph)**.** Consider a conical hybrid system $\mathcal{H}$ on $\mathbb{R}^n$ with modes $\mathcal{Q}$ and conical transition graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{W})$. For each mode $q \in \mathcal{Q}$, let $\mathcal{P}_q = \{P^i_q\}_{i \in \mathcal{I}_q}$ be a cover of $\mathbb{S}_0^{n-1}$ and for each $e \in \mathcal{E}$, let $\mathcal{I}_e^{\ominus}$ and $\mathcal{I}_e^{\oplus}$ be defined as in (6.32). The *abstract conical transition graph* (ACTG) defined by the partitions $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{|\mathcal{Q}|}$ is a directed graph $\widetilde{\mathcal{G}} = (\widetilde{\mathcal{V}}, \widetilde{\mathcal{A}}, \widetilde{\mathcal{W}})$ with set-valued weights. The vertex set $\widetilde{\mathcal{V}} := \mathcal{V}^{\ominus} \cup \mathcal{V}^{\oplus}$ is defined by

$$\mathcal{V}^{\ominus} := \bigcup_{(q^{\ominus}, q^{\oplus}) \in \mathcal{E}} \{q^{\ominus}\} \times \mathcal{I}^{\ominus}_{(q^{\ominus}, q^{\oplus})} \qquad \mathcal{V}^{\oplus} := \bigcup_{(q^{\ominus}, q^{\oplus}) \in \mathcal{E}} \{q^{\oplus}\} \times \mathcal{I}^{\oplus}_{(q^{\ominus}, q^{\oplus})}. \tag{6.34}$$

For each $v^{\ominus} := (q^{\ominus}, i^{\ominus}) \in \mathcal{V}^{\ominus}$ and each $v^{\oplus} := (q^{\oplus}, i^{\oplus}) \in \mathcal{V}^{\oplus}$, let

$$e := (q^{\ominus}, q^{\oplus}), \quad P^{\ominus} := P^{i^{\ominus}}_{q^{\ominus}}, \quad \text{and} \quad P^{\oplus} := P^{i^{\oplus}}_{q^{\oplus}}.$$

There is a *jump arrow* $\mathfrak{a}^{\mathrm{J}} := v^{\ominus} \xrightarrow{\mathrm{J}} v^{\oplus}$ in $\widetilde{\mathcal{A}}$ if $(A_e P^{\ominus}) \cap P^{\oplus}$ is nonempty. The set-valued weight of $\mathfrak{a}^{\mathrm{J}}$ is

$$\widetilde{\mathcal{W}}(\mathfrak{a}^{\mathrm{J}}) := \left\{ |A_e s^{\ominus}| \;\middle|\; s^{\ominus} \in P^{\ominus} \right\}. \tag{6.35}$$

For each $v^{(0)} := (q, i^{(0)}) \in \mathcal{V}^{\oplus}$ and each $v^{(\mathrm{f})} := (q, i^{(\mathrm{f})}) \in \mathcal{V}^{\ominus}$, let $P^{(0)} := P^{i^{(0)}}_q$ and $P^{(\mathrm{f})} := P^{i^{(\mathrm{f})}}_q$. There is a flow arrow $\mathfrak{a}^{\mathrm{F}} := (v^{(0)} \xrightarrow{\mathrm{F}} v^{(\mathrm{f})})$ in $\widetilde{\mathcal{A}}$ if for some $\tau > 0$, there exists $\xi : [0, \tau] \to \mathbb{R}^n$ such that

$$\xi(0) \in P^{(0)} \tag{6.36a}$$

$$\dot{\xi}(t) = f_q(\xi(t)) \quad \forall t \in (0, \tau) \tag{6.36b}$$

$$\xi(t) \in C_q \qquad \forall t \in (0, \tau) \tag{6.36c}$$

$$\operatorname{nrv}(\xi(\tau)) \in P^{(\mathrm{f})}. \tag{6.36d}$$

The weight of each flow arrow $\mathfrak{a}^{\mathrm{F}} = (q, i^{(0)}) \xrightarrow{\mathrm{F}} (q, i^{(\mathrm{f})})$ is

$$\mathcal{W}(\mathfrak{a}^{\mathrm{F}}) := \big\{ |\xi(\tau)| \;\big|\; \xi : [0,\tau] \to \mathbb{R}^n \text{ satisfies (6.36) for some } \tau > 0 \big\}. \tag{6.37}$$

$\diamond$

The following result establishes pre-asymptotic stability from ACTG's, analogously to Theorem 6.2 for CTG's.

Theorem 6.3. *Let $\mathcal{H} = (C, f, D, g)$ be a conical hybrid system with modes $\mathcal{Q}$ and conical transition graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{W})$. For each mode $q \in \mathcal{Q}$, let $\mathcal{P}_q = \{P_q^i\}_{i \in \mathcal{I}_q}$ be a cover of $\mathbb{S}_0^{n-1}$ with $\mathcal{I}_q$ finite, and let $\widetilde{\mathcal{G}} = (\widetilde{\mathcal{V}}, \widetilde{\mathcal{A}}, \widetilde{\mathcal{W}})$ be the abstract conical transition graph defined by $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{|\mathcal{Q}|}$. Suppose the following:*

(R1) *For each $q \in \mathcal{Q}$, the origin 0_n is pre-asymptotically stable for (C_q, f_q).*

(R2) *For each arrow $\mathfrak{a} \in \widetilde{\mathcal{A}}$, the weight $\widetilde{\mathcal{W}}(\mathfrak{a})$ is bounded.*

(R3) *For each well-formed elementary cycle w through $\widetilde{\mathcal{G}}$,*

$$\sup \widetilde{\mathcal{W}}(w) < 1.$$

Then, the set $\mathcal{O} := \mathcal{Q} \times \{0_n\}$ is pAS with respect to $\mathcal{H}$.

Proof. The proof proceeds by proving two facts.

Fact 1 There exists $M > 0$ such that for every walk $\tilde{w}$ through $\widetilde{\mathcal{G}}$, we have that

$$\sup \widetilde{\mathcal{W}}(\tilde{w}) \leq M, \tag{6.38}$$

and if $\tilde{w}$ is infinite, then $\widetilde{\mathcal{W}}(\tilde{w}) = \{0\}$.

Fact 2 For every walk $w := (v_0 \to v_1 \to \cdots \to v_K)$ through $\mathcal{G}$ (for some $K \in \{1, 2, \ldots, \infty\}$), there exists a walk $\tilde{w} := (\tilde{v}_0 \to \tilde{v}_1 \to \cdots \to \tilde{v}_K)$ through $\widetilde{\mathcal{G}}$ such that $\mathcal{W}(w) \subset \widetilde{\mathcal{W}}(\tilde{w})$.

These two facts, along with (R1), imply assumptions (P1)–(P3) of Theorem 6.2, so we can apply Theorem 6.2 to conclude $\mathcal{O}$ is pAS.

To prove Fact 1, let $\tilde{w}$ be any walk through $\widetilde{\mathcal{G}}$. Since $|\widetilde{\mathcal{V}}|$ is finite, every walk through $\widetilde{\mathcal{G}}$ returns to a vertex it has already visited every $|\widetilde{\mathcal{V}}| + 1$ or fewer steps (or possibly never, if the length of $\tilde{w}$ is less than $|\widetilde{\mathcal{V}}|$). As a result, $\tilde{w}$ must have the following structure:

1. The walk starts with an acyclical portion consisting of between zero and $|\widetilde{\mathcal{V}}|$-many steps that do not repeat any vertices.

2. The acyclical portion of the walk is followed by any number of cycles (infinitely many, if $\tilde{w}$ is infinite).

3. If the walk is finite, it ends with another acyclical portion of between zero and $|\widetilde{\mathcal{V}}|$-many steps.

Let $\tilde{w}_0$ be the acyclical portion of $\tilde{w}$ before the first cycle, let $\tilde{w}_f$ be the acyclical portion of $\tilde{w}$ after the last cycle, and let $\tilde{w}_c$ be the cyclical part in the middle. Let

$$\mu := \sup\Big\{\sup \widetilde{\mathcal{W}}(\mathfrak{a}') \;\Big|\; \mathfrak{a}' \in \widetilde{\mathcal{A}}\Big\}.$$

By (R2), every step in arrow in $\mathfrak{a}'$ has a bounded weight, so $\mu < \infty$. Thus,

$$\sup \mathcal{W}(\tilde{w}_0) \leq \Big(\sup \widetilde{\mathcal{W}}(\tilde{v}_0 \to \tilde{v}_1)\Big)\Big(\sup \widetilde{\mathcal{W}}(\tilde{v}_1 \to \tilde{v}_2)\Big)\cdots \leq \mu^{|\widetilde{\mathcal{V}}|}.$$

Similarly, $\sup \mathcal{W}(\tilde{w}_f) \leq \mu^{|\widetilde{\mathcal{V}}|}$. For the cyclical portion $\tilde{w}_c$, we have, per (R3), that

$$\sup \widetilde{\mathcal{W}}(\tilde{w}_c) < 1$$

because each cycle multiplies the weight by a value less than 1. Thus, the weight of $\tilde{w}$ must satisfy

$$\sup \mathcal{W}(\tilde{w}) < M := \mu^{2|\widetilde{\mathcal{V}}|},$$

proving Fact 1.

To show Fact 2, let $w := (v_0 \xrightarrow{\ell_0} v_1 \xrightarrow{\ell_1} \ldots \xrightarrow{\ell_{K-1}} v_K)$ be any walk through $\mathcal{G}$ with $K \in \{1, 2, \ldots, \infty\}$. Take any $k \in \{0, 1, \ldots, K-1\}$. Suppose $\ell_k = \textsc{j}$ and let

$$v^{\ominus} := (q^{\ominus}, s^{\ominus}) := v_k, \quad v^{\oplus} := (q^{\oplus}, s^{\oplus}) := v_{k+1}, \quad \text{and} \quad e := (q^{\ominus}, q^{\oplus}) \in \mathcal{E}.$$

Per (6.15),

$$s^{\ominus} \in D_e \cap \mathbb{S}_0^{n-1} \quad \text{and} \quad s^{\oplus} = \mathrm{nrv}(A_{(q^{\ominus}, q^{\oplus})} s^{\ominus}) \in A_e D_e \cap \mathbb{S}_0^{n-1}.$$

Because $\{P^{i^{\ominus}}_{q^{\ominus}}\}_{i^{\ominus} \in \mathcal{I}_e^{\ominus}}$ covers D_e, there is some $i^{\ominus} \in \mathcal{I}_e^{\ominus}$ such that $s^{\ominus} \in P^{\ominus} := P^{i^{\ominus}}_{q^{\ominus}}$. Similarly, for some $i^{\oplus} \in \mathcal{I}_e^{\oplus}$, we have $s^{\oplus} \in P^{\oplus} := P^{i^{\oplus}}_{q^{\oplus}}$. Since $s^{\oplus} \in A_e P^{\ominus} \cap P^{\oplus}$, we have that $(q^{\ominus}, i^{\ominus}) \xrightarrow{\textsc{j}} (q^{\oplus}, i^{\oplus})$ is an arrow in $\widetilde{\mathcal{A}}$. The weight of $v^{\ominus} \xrightarrow{\textsc{j}} v^{\oplus}$ is $\{|A_e s^{\ominus}|\}$, which is a subset of $\widetilde{\mathcal{W}}(v^{\ominus} \xrightarrow{\textsc{j}} v^{\oplus})$, per (6.35).

Alternatively, suppose $\ell_k = \text{F}$ and let

$$v^{(0)} := (q, s^{(0)}) := v_k \quad \text{and} \quad v^{(\text{f})} := (q, s^{(\text{f})}) := v_{k+1}.$$

Take any $r \in \mathcal{W}(v^{(0)} \xrightarrow{\text{F}} v^{(\text{f})})$. Per (6.18), there exist $\tau > 0$ and $\xi : [0, \tau] \to \mathbb{R}^n$ that satisfy (6.17) such that $r = |\xi(\tau)|$. We have that $v^{(0)} \in \mathcal{V} \cap G(D)$, so there exists $q^{\ominus} \in \mathcal{Q}$ such that $e := (q^{\ominus}, q) \in \mathcal{E}$, and $s^{\ominus} \in D_e$ such that

$$s^{(0)} = A_e s^{\ominus} \in A_e D_e.$$

Thus, there exists $i^{(0)} \in \mathcal{I}_e^{\oplus}$ such that $s^{(0)} \in P^{(0)} := P_q^{i^{(0)}}$. Similarly, $v^{(\text{f})} \in \mathcal{V} \cap D$, so there exists $q^{\oplus} \in \mathcal{Q}$ such that $e := (q, q^{\oplus}) \in \mathcal{E}$ and

$$s^{(\text{f})} \in D_e.$$

Thus, there exists $i^{(\text{f})} \in \mathcal{I}_e^{\ominus}$ such that $s^{(\text{f})} \in P^{(\text{f})} := P_q^{i^{(\text{f})}}$. We then have that $\tilde{v}^{(0)} := (q, i^{(0)}) \in \mathcal{V}^{\oplus}$ and $\tilde{v}^{(\text{f})} = (q, i^{(\text{f})}) \in \mathcal{V}^{\ominus}$, and

$$\xi(0) = s^{(0)} \in P^{(0)} \quad \text{and} \quad \text{nrv}(\xi(\tau)) s^{(\text{f})} \in P^{(\text{f})},$$

satisfying (6.36). Therefore, $\tilde{v}^{(0)} \xrightarrow{\text{F}} \tilde{v}^{(\text{f})}$ is an arrow in $\widetilde{\mathcal{A}}$ and

$$r = |\xi(\tau)| \in \widetilde{\mathcal{W}}(\tilde{v}^{(0)} \xrightarrow{\text{F}} \tilde{v}^{(\text{f})}).$$

In the manner described above, we construct a walk

$$\tilde{w} := (\tilde{v}_0 \xrightarrow{\ell_0} \tilde{v}_1 \xrightarrow{\ell_1} \dots \xrightarrow{\ell_K} \tilde{v}_K),$$

and since

$$\mathcal{W}(v_k \xrightarrow{\ell_k} v_{k+1}) \subset \widetilde{\mathcal{W}}(\tilde{v}_k \xrightarrow{\ell_k} \tilde{v}_{k+1}) \quad \forall k \in \{0, 1, \dots, K-1\}$$

we have that

$$\mathcal{W}(w) \subset \widetilde{\mathcal{W}}(\tilde{w}),$$

completing the proof of Fact 2.

It follows from Facts 1 and 2 that (P2) and (R3) hold, so by Theorem 6.2, the set $\mathcal{O}$ is pAS for $\mathcal{H}$. □

6.7 Numerical Example

In this section, we present an example where we construct an abstract CTG for a hybrid system with modes and apply Theorem 6.3 to determine asymptotic stability of the origin. In particular, we consider a hybrid system $\mathcal{H}$ as in (6.6) in $\mathbb{R}^2$ with two modes, $\mathcal{Q} := \{0, 1\}$. The system has linear flows maps in each mode $q \in \mathcal{Q}$ defined by $\dot{z} = A_q z$ where,

$$A_0 = \begin{bmatrix} 2 & 2 \\ -3 & 1 \end{bmatrix} \qquad\qquad A_1 = \begin{bmatrix} -1 & 1 \\ -4 & -2 \end{bmatrix}.$$

The eigenvalues of A_0 and A_1 are complex, resulting in flows that spiral around the origin, with the flows in mode $q = 0$ spiraling outward and the flows in $q = 1$ spiraling inward. The components of the flow set in each mode are

$$C_0 := \big\{(x_1, x_2) \in \mathbb{R}^2 \;\big|\; x_1 \leq 0\big\} \quad \text{and} \quad C_1 := \mathbb{R}^2,$$

where the choice of $C_0 \neq \mathbb{R}^2$ is important to ensure that the origin is stable for flows in mode 0, since solutions spiral outward but can only flow for a finite amount of time before reaching the boundary of C_0.

In each mode, the system can jump within the same mode or jump to the other mode, so the set of mode transition edges is

$$\mathcal{E} := \{(0,0), (0,1), (1,0), (1,1)\}.$$

The jump map for each transition $e \in \mathcal{E}$ is defined by a linear map $z^+ = A_e z$, where

$$\begin{aligned} A_{(0,0)} &= \begin{bmatrix} 1 & 1/2 \\ -2 & 2 \end{bmatrix}, \quad A_{(0,1)} = \gamma \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \\ A_{(1,0)} &= \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}, \qquad A_{(1,1)} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \end{aligned} \tag{6.39}$$

where $\gamma > 0$ is a parameter we discuss in Section 6.7.1. The jump sets to trigger a jump along each transition are

$$\begin{aligned} D_{(0,0)} &:= \operatorname{cone}\big(\left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} -4 \\ -1 \end{smallmatrix}\right]\big), & D_{(0,1)} &:= \operatorname{cone}\big(\left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right], \left[\begin{smallmatrix} -1 \\ 2 \end{smallmatrix}\right]\big), \\ D_{(1,0)} &:= \operatorname{cone}\big(\left[\begin{smallmatrix} -4 \\ -1 \end{smallmatrix}\right], \left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix}\right]\big), & D_{(1,1)} &:= \operatorname{cone}\big(\left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} 4 \\ 1 \end{smallmatrix}\right]\big). \end{aligned}$$

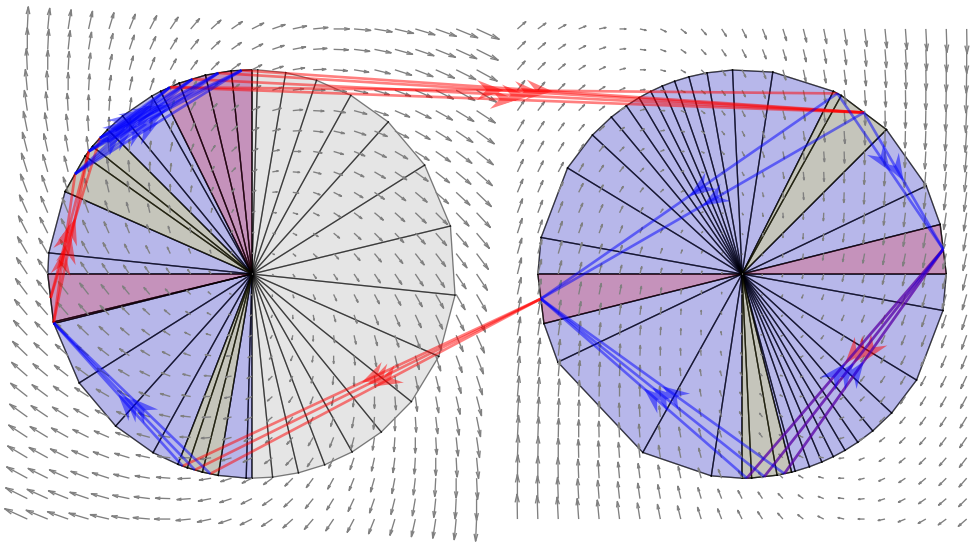

Figure 6.5. The system $\mathcal{H}$ from Section 6.7 overlaid by an ACTG. Mode $q = 0$ is shown on the left and $q = 1$ is on the right. For each $q \in \mathcal{Q}$ and $e \in \mathcal{E}$, the set C_q is blue, D_e is red, and $A_e D_e$ are yellow. Jump arrows are drawn as red lines and flow arrows are blue.

A plot of the sets in $\mathcal{H}$ is shown in Figure 6.5, overlaid with the arrows of the conical transition graph.

The conical partition $\mathcal{P}_0$ and $\mathcal{P}_1$ are constructed using a method similarly to the authors of [35], with additional partitions added as needed so that each boundary of C_q, D_e, and $A_e D_e$ align with the boundaries of cones in the partition. As a result, every cone in the partition is either entirely inside or entirely outside C_q, D_e, and $A_e D_e$, respectively.

The construction of flow arrows requires determining reachability from each cone in $G(D)$ to each cone in D via flows in C. In each cone of the conical partition, we find the adjacent cones (those that share a boundary), and determine the direction of flow through the boundary. In $\mathbb{R}^2$, all convex cones are polyhedral, which we exploit in our implementation. The find the full set of reachable points reachability analysis, we over approximate the reachable set within a cone K from a polyhedron set of

initial positions $P_0 \subset K$ for flows along $\dot{z} = A_q z$ using the fact that $A_q z \in A_q K$ for all $x \in K$. Thus, the reachable set from P_0 in K is given by $(P_0 + A_q K) \cap K$. By picking $P_0 \supset \mathbb{S}^n$, we can over approximate the change in magnitude of a solution as it flows through a cone, allowing us to construct the weights of flow arrows. The code for this example is available at github.com/pwintz/conical-transition-graph.

6.7.1 Results

In Figure 6.6, we present the maximum and minimum weights for cycles through the ACTG for various choices of $\gamma > 0$, used to define $A_{(0,1)}$ in (6.39). We see that for small values of γ, the maximum weight is less than 1, satisfying (R3) in Theorem 6.3. Furthermore, (R1) and (R2) can also be shown to hold. Therefore, by Theorem 6.3, the set $\mathcal{Q} \times \{0_n\}$ is pAS for $\mathcal{H}$. Increasing γ above $\gamma \approx 10^{-1}$, however, causes the maximum cycle weight to become greater than 1, so Theorem 6.3 no longer applies. Note, however, that this is insufficient to conclude that the system becomes unstable—the test is indeterminate and the actual value of γ where instability occurs is likely larger. Over approximations used in the construction of the ACTG cause the maximum walk weight to be inflated. Examining Figure 6.6, we see that the effect of modifying γ becomes saturated. As γ increases, the minimum cycle weight increases up to a point. After $\gamma = 10^0$, increasing γ has no effect on the minimum cycle weight. The cause of this is the presence of cycles in the graph that don't pass through the transition that depends on γ. Similarly, as γ decreases toward zero, the maximum cycle weight also saturates, as the cycle with the largest cycle becomes one with no dependence on γ.

6.8 Future Work

There are several avenues for future work on conical transition graphs. There are some promising directions for expanding the generality of the proposed approach. One could relax assumptions on the system to allow for more general types of dynamics, such as allowing for higher order homogenous systems. In particular, the approach is agnostic to how quickly the magnitude of the flow map grows along each ray, so long as all the flows along each ray points in the same direction. In fact, that requirement could also be relaxed to allow for systems with moderate nonlinearities

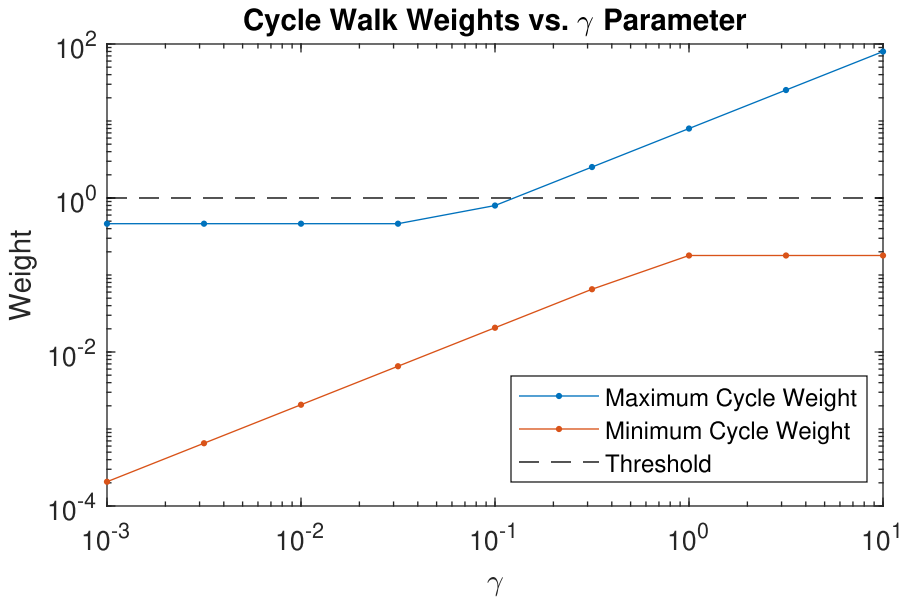

Figure 6.6. Maximum and minimum weights of cycles in the ACTG of the hybrid system $\mathcal{H}$ described in Section 6.7 for various values of γ.

in the direction of flow. We are also interested in extending the CTG approach to include a broader class of hybrid systems, in particular hybrid systems with set-valued flow and jump maps as in [24, Thm. 3.16]. By extending the approach to allow for set-valued dynamics, could over approximate nonlinear vector field as a set-valued map that contains all the flow directions along a particular ray from the origin, or within some cone.

Chapter 7

Simulator for Hardware Architecture and Real-time Control (SHARC)

One use case for the uniting feedback schemes presented in Chapters 2 and 3 and Section 4.2 is to safely exploit advanced controllers that cannot be certified due to unpredictable computational delays. By utilizing uniting feedback, one can provide a fast, certified controller as a backup in case the advanced controller fails due to computational delays. For safety critical applications, however, one may wish to verify that the uniting feedback scheme will work as expected when deployed on a physical system with computational delays. In many cases, deploying on the real system, however, is expensive and risky, so it is preferable to perform initial verification in simulations. In this chapter, we present the Simulator for Hardware Architecture and Real-time Control (SHARC). We designed SHARC to accurately co-simulate computational hardware and physical dynamics, incorporating computational delays into control updates [49].

7.1 Introduction

Cyber-physical systems (CPSs) are engineered systems that incorporate digital sensors, computers, and actuators interacting with physical processes. CPSs are ubiquitous in modern critical infrastructure such as transportation systems, energy delivery, and health care. Typically, a CPS has strong coupling between its computational hardware, physics, and control algorithms. CPU designers, however, usually optimize for generic instruction sets—not for a specific algorithm—whereas control designers typically do not design their algorithms for a particular hardware architec-

ture. Many properties of the physical system, including stability, safety, and liveness, are affected by the latency of computing the next control input. The computational delay depends on the underlying computational hardware on which the control algorithm is run. Thus, to develop CPSs that satisfy demanding design specifications, engineers must jointly consider the computing power and control methods. To address this challenge, this chapter introduces a tool called the *Simulator for Hardware Architecture and Real-time Control* (SHARC) that simulates the physical evolution of a CPS and the execution of control algorithms on different computing platforms to incorporate realistic controller delays in the closed-loop simulation. The ability to incorporate accurate computation delays into simulations allows system designers to gain a better understanding of how computational limitations affect the behavior of the system, thereby informing better designs.

To allow for quick and easy installation of SHARC, a Dockerized version of SHARC is provided. To aid in the development of SHARC and implementation of controllers and dynamics using SHARC, the project is configured to support running Docker images in a Dev-container. A suite of unit tests is included in the SHARC to verify SHARC's internal logic.

7.1.1 Problem Setting

Creating tools to analyze CPSs is challenging due to the interactions between computational hardware, physics, and control software. There has been a trend toward consolidating control software components onto shared multicore processors to reduce size, weight, and power while improving performance. The adoption of multicore hardware in practical CPSs, including automotive [50], avionics [51], and medical systems [52], brings many benefits but also introduces a new host of challenges for modeling and analysis. In a multicore processor, some resources, such as the caches, memory bus, and random access memory (RAM), are shared between cores, which affects the timing of the control software in complex ways, making the timing difficult to model and predict.

Beyond traditional CPUs, specialized computing hardware is becoming prevalent in CPSs, including highly-parallelized processors (GPUs), configurable hardware (e.g., field programmable gate arrays, or FPGAs), and domain-specific accelerators (such as chips designed for self-driving autonomous driving). These components are

critical for the unique demands of real-time machine learning, computer vision, and other applications. On the other hand, control software is also becoming increasingly complex, with diverse and rapidly changing resource requirements and performance goals [53]. Computation-aware control algorithms require detailed information about the performance of the computational hardware via strong performance monitoring capabilities in order to understand how to safely and effectively optimize the hardware and control algorithms.

Creating tools that can assess and simulate complicated interactions between hardware and software is critical for automotive, avionics, and other domains in which computationally-intensive processes play a significant role at runtime. In particular, it is important to ensure that advanced control algorithms can run on the available computing hardware with high confidence. For example, automotive Original Equipment Manufacturers need to evaluate whether updated software for advanced controllers or perception algorithms can be safely deployed on vehicle models of the previous year. To ensure that advanced control algorithms can run on unsophisticated computing hardware, regression analysis of candidate controllers should be concurrently tested *in situ* for deployed systems. Some algorithms for achieving robust autonomy, such as model predictive control (MPC), have limited deployment due to insufficient computing power. By using SHARC, however, to simulate, analyze, and optimize controllers and hardware platforms, engineers can design solutions that improve the use of onboard energy and computational sources, allowing cheaper and more efficient implementations of advanced control schemes.

7.1.2 Literature Review

Prior research has investigated quantifying the computational demands of various control algorithms and mitigating the effects of computational delays. The authors of [54], [55], [56], [57], [58] investigate the effects of the worst-case execution time on the performance of several control algorithms, including linear quadratic control (LQR), model predictive control (MPC), and state-dependent Riccati equation (SDRE) nonlinear control. In particular, linear and nonlinear MPC schemes that preserve stability and performance under computational delays are presented in [59], [60] with a brief introduction given in [59, Section 7.6]. However, none of these schemes provides a tool or methodology that explicitly accounts for the effects

of many kinds of computational hardware, including hardware that has not yet been fabricated. In contrast to these works, SHARC is able to analyze the effects of diverse microarchitectures—including hypothetical configurations—on the performance of the closed-loop system. This allows the user to conduct both microarchitecture design exploration and control design optimization.

CPU manufacturers and computer architecture researchers rely on microarchitectural simulation to explore the hardware design space, prototype new hardware ideas, and evaluate application performance on hypothetical hardware. Microarchitecture simulators model the internal hardware architecture of CPUs, including branch predictors, instruction fetch and decode units, functional units, instruction schedulers, and memory subsystems with multiple levels of caches. Some noteworthy microarchitecture simulators are *gem5* [61], *ChampSim* [62], *ZSim* [63], *MARS* [64], and *Sniper* [65]. While these simulators enable accurate application simulation and performance modeling, they are not designed to simulate controllers interacting in a closed-loop with a physical system. In particular, they cannot be directly integrated into a model where the controller interacts with a physics simulation since microarchitecture simulators do not incorporate methods for synchronizing the passage of time in dynamical simulations with the execution of the controller code inside the microarchitecture simulator.

Prior works have proposed tools for testing control algorithms interacting with physical systems via co-simulations and via hardware-in-the-loop (HIL). Co-simulation tools [66], [67] provide simulations of the CPU capable of modeling hardware events such as interrupts, but they do not provide a cycle-accurate timing simulation, which is necessary to ensure the safe and reliable operation of control algorithms under computational constraints. Hardware-in-the-loop simulation integrates real-time hardware into a simulated environment, enabling realistic validation of control algorithms. The system's physics are simulated on a real-time platform, interacting in a closed loop with actual hardware, such as an embedded controller. HIL is widely used in automotive and aerospace applications to evaluate controllers for autonomous vehicles, flight control systems, and industrial automation, ensuring robust performance before deployment [68], [69]. However, traditional HIL setups rely on fixed hardware, limiting flexibility in exploring different architectures or optimizing control algorithms under varying constraints. SHARC overcomes these

limitations by using a reconfigurable cycle-accurate microarchitectural simulator in a closed-loop with a simulation of the system's physics.

The remainder of the paper is structured as follows. Section 7.2 introduces the modeling framework for the physics and the computational hardware, as used by SHARC. Section 7.3 describes the implementation and basic usage of the simulator, with a serial execution mode described in Section 7.3.1 and a parallel mode described in Section 7.3.2. Two examples are presented in Section 7.4. In particular, section Section 7.4.1 contains an example of adaptive cruise control for longitudinal vehicle control via linear MPC. Section 7.5 describes a number of future research directions and gives conclusions.

7.2 Modeling

In this section, we introduce our modeling framework for the physics, controller, and computational hardware of a CPS, and their interconnection.

7.2.1 Physics and Controller

A physical system controlled by a controller is often called a *plant*. The physics of a plant are typically modeled as a differential equation, which we write as

$$\dot{x} = \tilde{f}(t, x, u, w), \tag{7.1a}$$

$$y = h(x, u, w). \tag{7.1b}$$

The plant has state $x \in \mathbb{R}^{n_x}$, control input $u \in \mathbb{R}^{n_u}$, output $y \in \mathbb{R}^{n_y}$, and a disturbance $w \in \mathbb{R}^{n_w}$. The disturbance (or *exogenous input*) is given as a function $t \mapsto w(t) \in \mathbb{R}^{n_w}$ for all $t \geq 0$. Although physical systems are nicely represented—mathematically—by differential equations, most methods for numerically simulating continuous-time systems use discretization. In SHARC, we use a discrete model on an evenly-spaced time grid with period $T > 0$, defined by $t_k := kT$ for each $k \in \mathbb{N}$. We write the discrete dynamics of the plant as

$$x_{k+1} = f(t_k, x_k, u_k, w(t_k)), \tag{7.2}$$

where f is a discretization of the physics with sample time T, and $x_k := x(t_k)$, $u_k := u(t_k)$, and $y_k := y(t_k)$ for each k. Figure 7.1 illustrates the discretization of the continuous-time physics in (7.1).

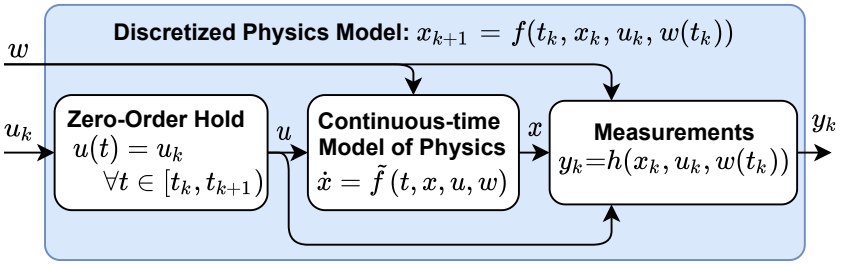

Figure 7.1. Diagram showing how a continuous-time physics model in the form (7.1) is discretized into form (7.2).

The control input u is generated by a control algorithm that evaluates a control function $(t, y) \mapsto g(t, y) \in \mathbb{R}^{n_u}$. We assume that control values can only change at sample times. When discretizing (7.1), as shown in Figure 7.1, SHARC interpolates the input between consecutive time steps t_k and t_{k+1} using zero-order hold. In particular, $u(t) = u_k$ for all $t \in [t_k, t_{k+1})$, where u_k is the value of the control input received from the controller at t_k. The output y_k represents periodic sensor measurements.

SHARC users can implement the physics in two ways. If they are starting with a continuous-time system, they can provide $\tilde{f}$ to have SHARC automatically generate and evaluate the discretization f via numerical integration. Alternatively, users can provide f directly if they wish to implement other types of physical models, such as hybrid systems or stochastic differential equations.

7.2.2 Interaction between Physics and Controller with Computation Delays

In an idealized system, the controller g immediately provides the next control value $g(t_k, y_k)$ at time t_k. In realistic systems, however, computing the control update takes time, so the new control value is not available until after some computation delay $\tau_k > 0$. To capture the delay, we execute the controller code to calculate $g(t_k, y_k)$ and store the "pending" control value in a memory variable $\tilde{u}_{k+1} := g(t_k, y_k)$ until the next sample time after $\tilde{t}_{k+1} := t_k + \tau_k$. To determine the computation delay τ_k, SHARC simulates the execution of the controller code on a given processor using a cycle-accurate microarchitectural simulator, as described in Section 7.2.3. The system continues to use $u = u_k$ until the next sample time after $\tilde{t}_k$, at which point

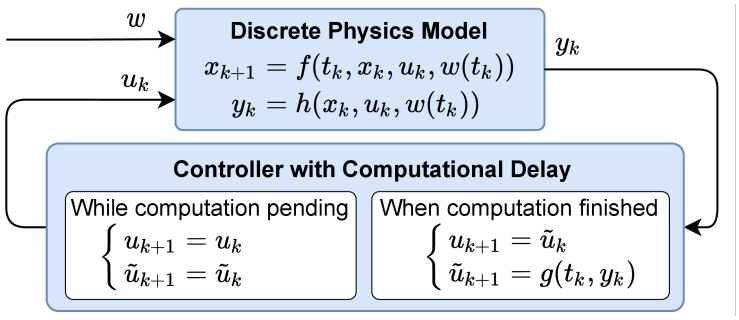

Figure 7.2. A feedback diagram of the closed-loop model of the physics with a controller that has computational delays. After a control computation is started but before its computation time has elapsed in the simulation, it is stored as $\tilde{u}_k$. The values of u and $\tilde{u}$ are held constant until the computation is finished.

we set $u = \tilde{u}_k$. Figure 7.2 shows the feedback diagram for the closed-loop system using the discrete system physics model in (7.3) interconnected with a controller that exhibits computational delays.

Thus, the model used by SHARC for the closed-loop dynamics of a CPS's controlled by a computationally delayed controller is

$$x_{k+1} = f\big(t_k, x_k, u_k, w(t_k)\big) \tag{7.3a}$$

$$y_k = h\big(x_k, u_k, w(t_k)\big) \tag{7.3b}$$

$$\text{If } t_{k+1} < \tilde{t}_k, \text{ (Computation in progress)} \begin{cases} u_{k+1} = u_k \\ \tilde{u}_{k+1} = \tilde{u}_k \\ \tilde{t}_{k+1} = \tilde{t}_k \end{cases} \tag{7.3c}$$

$$\text{If } t_{k+1} \geq \tilde{t}_k, \text{ (Computation finished)} \begin{cases} u_{k+1} = \tilde{u}_k \\ \tilde{u}_{k+1} = g(t_k, y_k) \\ \tilde{t}_{k+1} = t_k + \tau_k \text{ for computing } g(t_k, y_k), \end{cases} \tag{7.3d}$$

where the initial state $x_0 \in \mathbb{R}^{n_x}$ and initial control value $u_0 \in \mathbb{R}^{n_u}$ are given, the initial pending control value is $\tilde{u}_0 = g(t_0, y_0)$, and $\tilde{t}_0 := \tau_0$ is the time required to execute $g(t_0, y_0)$. In (7.3c), the memory variable $\tilde{u}$ and the end time $\tilde{t}$ of the computation are held constant while the computation is in progress. When the computation finishes, u is set to $\tilde{u}$. Then $\tilde{u}$ and $\tilde{t}$ are updated to record a new execution of the calculation of $g(t_k, y_k)$.

7.2.3 Computational Hardware Simulation

In this section we introduce how we model and simulate the computational hardware. To calculate the computational delay of a given controller, we use a microarchitectural simulator. Existing microarchitectural simulators can execute accurate simulations of arbitrary computer programs on a given hardware platform. We will discuss in Section 7.3 how SHARC jointly simulates the hardware execution of a control algorithm and the physics of a system. In SHARC, the *Scarab Microarchitectural Simulator* [70], [71] is used to perform hardware simulation. To simulate the execution time of a control algorithm, Scarab processes either compiled x86 binaries or traces of x86 assembly instructions. The microarchitectural simulator is agnostic to the programming language because it consumes compiled assembly code.

Although it is possible to measure application execution times on a physical processor, simulating the controller executable with Scarab provides the following advantages:

1. Scarab allows for arbitrary modification of hardware parameters, such as the cache size, clock frequency, and the depth and width of the CPU pipeline, to allow for analysis of hypothetical computing platforms without necessitating fabrication. Thus, by using Scarab with SHARC, we can prototype new hardware components and measure the resulting changes in a system's dynamical performance.

2. Scarab produces detailed statistics, making the internal hardware state observable and thereby allowing for better performance analysis. This is in contrast to physical CPUs, which only provide limited visibility through existing performance monitoring unit (PMU) counters.

All these aspects are crucial for enabling optimal hardware-software co-design of control systems.

The simulation of control algorithms in Scarab provides high precision and fidelity because Scarab models both the architectural and microarchitectural states of the CPU at the level of individual clock cycles. The architectural state includes all registers, program counters (PC), and the memory of the processor as specified by the instruction set architecture (ISA). The microarchitectural state comprises the tables

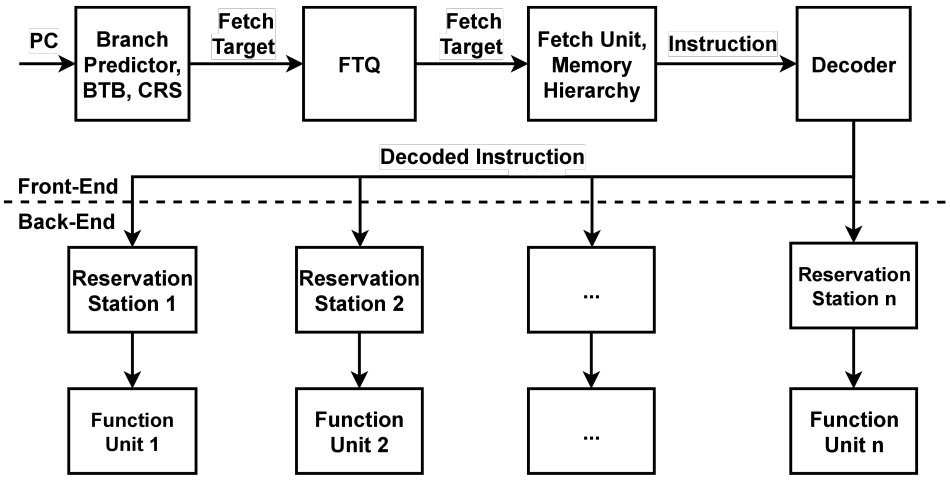

Figure 7.3. Scarab's Architecture. Modern CPUs are comprised of the *Frontend*, responsible for predicting future executed instructions (Branch Predictor), buffering their instruction address (FTQ), fetching them from the instruction cache (Fetch), and decoding their arithmetic operation (Decoder). Decoded instructions are then forwarded to the *Backend*, which contains the instruction schedulers (Reservation Stations) selecting ready instructions to be processed by the functional units (Units 1 through N).

and internal meta-data utilized by the branch predictor [72], prefetchers, and cache replacement mechanisms. In a microarchitectural simulator, each simulated assembly instruction moves through a pipeline of various stages during its lifetime, including the *fetch*, *decode*, *execution*, and *retirement* stages. At each stage, the instruction triggers events along its path. Modern CPUs implement instruction pipelines that are deep and wide, meaning that there can be hundreds of instructions in the pipeline at the same time, each one triggering events in every cycle. The full pipeline of instruction processing is accurately modeled by Scarab.

Furthermore, the modern CPU architecture, as emulated by the Scarab simulator, follows an out-of-order CPU design that can be divided into two parts, as shown in Figure 7.3. The *front-end* identifies the next instructions to be fetched from main memory, stores them into the fetch target queue (FTC) and instruction cache, and decodes them. The front end is also responsible for handling control-flow instructions, such as jumps and branches, utilizing a TAgged GEometric (TAGE) history length branch predictor [72], branch target buffer (BTB), and return address stack (RAS) predictor.

The *back-end* consumes the stream of instructions provided by the front-end and executes them through different functional units based on the instruction type (e.g., loads, stores, Arithmetic Logic Unit (ALU), vector instruction queues) acting as reservation stations [73]. The instruction scheduler picks instructions as soon as they are ready (all source operands are available) and forwards them to the appropriate functional units. The execution stage also detects mispredicted branch instructions to trigger pipeline flushes ensuring correct execution. To emulate, serve, load, and store instructions, the simulator models three cache levels and implements a detailed Dynamic Random Access Memory (DRAM) model utilizing Ramulator [74].

The Scarab simulator provides observability of over a thousand low-level events, including the number of executed CPU cycles, mispredicted control-flow instructions, data and instruction cache misses, and a tally of the number of cycles each functional unit is busy. Analyzing these statistics reveals which CPU components limit the performance of a particular program and thereby provides insights into how to improve the hardware architecture or software implementation. Scarab features two simulator frontends: an execution-driven and a trace-based approach. We utilize trace-based simulation to supply instructions to the CPU pipeline. The traces, captured using DynamoRIO [75], preserve a precise continuous sequence of dynamically executed instructions including memory addresses for load and store operations. DynamoRIO is a runtime code manipulation system that enables dynamic analysis, profiling, and optimization by allowing arbitrary modifications to application instructions on various architectures and operating systems.

7.3 SHARC Simulator

In SHARC, the microarchitectural simulator is executed in parallel with a simulation of the physics. Figure 7.4 illustrates how SHARC simulates the physics and the control algorithm in parallel. The simulation of the physics is executed through a user-provided implementation of a Python interface, which may call external physics simulators. For each simulation experiment, one or more subprocesses are started by SHARC to simulate a controller executable with the microarchitectural simulator. The particular dynamics of a system are defined by writing a subclass of a Python class named `Dynamics`, provided by SHARC. Pseudocode for `MyDynamics` subclass

of `Dynamics` is shown here:

```
class MyDynamics(Dynamics):
  def evolve_state(self, t0, x0, u, tf):
    # Evolve the state from t0 to tf given
    # initial state x0 and control input u.
    return xf # Final state of the system at tf.

  def get_output(self, x, u, w):
    return y # Generate output

  def get_exogenous_input(self, t):
    return w # Generate exogenous input
```

Similarly, a controller is defined in C++ by writing a subclass of a C++ class provided with SHARC named `Controller`. Pseudocode for a `MyController` subclass of `Controller` is as follows:

```
class MyController : public Controller {
    void calculateControl(int k, double t, const Vec &y){
        // Evaluate u = g(t, y) and set the
        // object's 'control' property to the result.
        control = u;
    }
};
```

The code in the `calculateControl` function is simulated by Scarab to determine the computational delay of computing $g(t, y)$. A noteworthy feature of SHARC's design is that the same controller code can be used by SHARC as would be deployed on an actual cyber-physical system.

7.3.1 Serial Mode

The SHARC simulator supports two modes. While the *serial* mode is optimized for maximum accuracy, the *parallelized* mode minimizes simulation time through parallel processing. We will describe the serial mode in the following and refer to Section 7.3.2 for a detailed description of the parallel mode. When running in serial mode, SHARC executes the controller in a single subprocess that runs for the entire duration of the simulation.[1] The controller subprocess simulates the controller with Scarab using an "execution-driven" mode, which allows for statistics, such as CPU cycle counts, to be accessed during the execution of the simulation, as opposed to having to wait until the simulation completes. At each time step,

[1] Here, "serial" vs. "parallel" mode refers only to whether one time step or many are computed concurrently. In the serial mode, parallelization is used to run controller and physics concurrently.

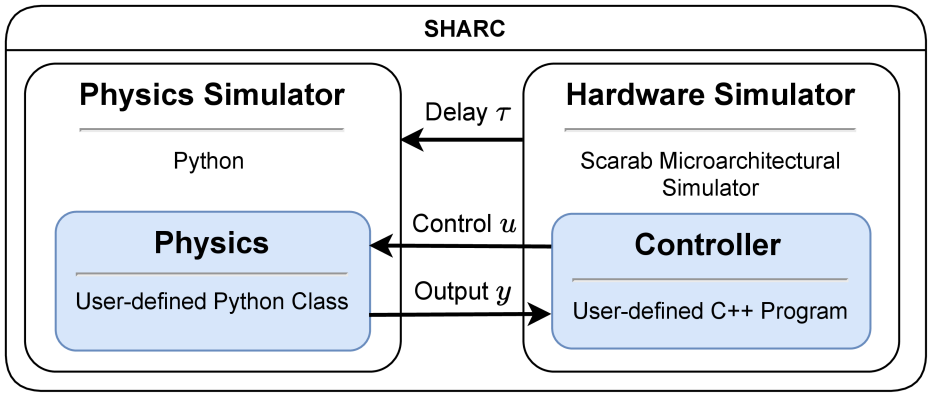

Figure 7.4. Diagram of the SHARC simulator structure. The physics simulator and hardware simulators run in separate processes, with inter-process communication done via named pipe files.

SHARC sends the current time step k, the current time t_k and output y_k to the controller. The exogenous input w is generated at each time $t \in [t_k, t_{k+1}]$ using the `get_exogenous_input` function. Once the values are received by the controller, Scarab begins recording statistics while the control value is computed, at which point the statistics are saved, and the control value u is sent back to the Python process running the physics simulator. The inter-process communication between the dynamics simulator and the controller is accomplished using named pipe files. After u is received by the physics simulator, SHARC reads the computation statistics from Scarab, which it uses to determine the computation time τ for computing u.

7.3.2 Parallel Mode

For computationally intensive control algorithms, running simulations in serial mode can take a long time. Due to simulating the microarchitectural components of a CPU, the time to simulate a controller algorithm with Scarab can be over 10,000 times longer than computing the same algorithm on physical hardware. To address this challenge, we developed a method to improve simulation times through parallel execution.

Recall that the discretized physics are assumed to use a periodic time grid $t_0 := 0$, $t_1 := T$, $t_2 := 2T$, ..., with both the sensor measurements and control inputs discretized at a constant sampling rate of T. This allows SHARC to parallelize simulations across time steps. Figure 7.5 provides an overview of the parallelized

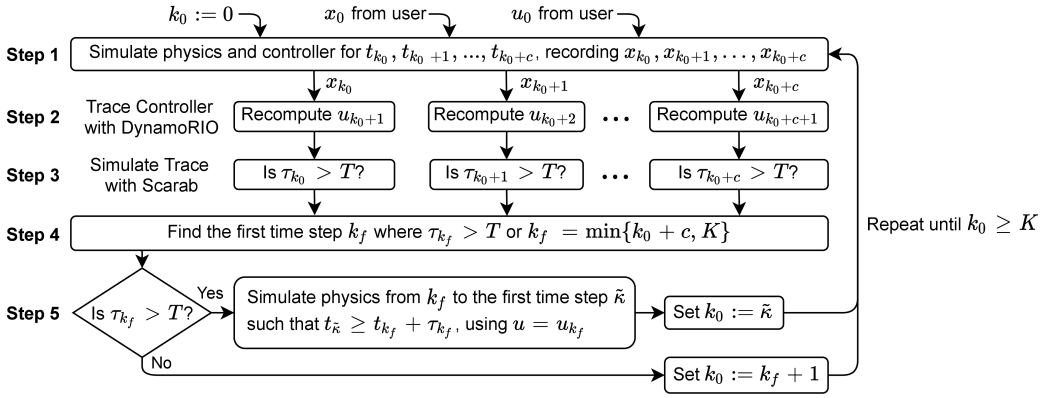

Figure 7.5. Diagram of SHARC's parallel mode using a simulation horizon of K time steps parallelized across c processors. Each column is executed in parallel. In this diagram, full-state feedback $y_k = x_k$ is used, for simplicity.

approach, where full-state feedback is used for simplicity ($y_k = x_k$). Each column of Steps 2 and 3 are executed in parallel.

The simulation takes the initial state value $x_0 \in \mathbb{R}^{n_x}$ and initial control value $u_0 \in \mathbb{R}^{n_u}$ (which is to be applied until the first control update is finished) and runs over a time horizon $K \in \mathbb{N}$. It is divided into batches each containing c-many time steps, where c is typically the number of processors available. During Step 1, the simulator assumes that each control update u_{k+1} can be computed within one sample time ($\tau_k < T$) and thus always updates the control at the next time step. The resulting sequence x_{k_0}, x_{k_0+1}, ..., x_{k_0+c} is an initial guess of the trajectory that the system would take if every control update is computed within one sample time ($\tau_k < T$). Since Step 1 assumes no delays in computing the control inputs, Scarab is not used to record computation times, so this step is executed very quickly.

In Steps 2 and 3, SHARC backtracks to check whether the computational delay of updating the control at each time step is actually less than T. In particular, SHARC creates c-many processes—one for each time step in the batch. Each process is assigned a unique $t_k \in \{t_{k_0}, t_{k_0+1}, \ldots, t_{k_0+c-1}\}$. Using the precomputed value x_k and u_k, the simulator recomputes the control update $u_{k+1} = g(t_k, y_k)$, but this time SHARC runs the controller executable with DynamoRIO to generate a trace of its execution (Step 2) which is then simulated using Scarab to generate the computation time τ_k (Step 3).

In Step 4, SHARC searches for the first time step k_f where the computation

time τ_{k_f} exceeds T. If such a time step is found, then any subsequent time steps ($k > k_f$) are invalid because the states were generated using control values that were applied (in the simulation) before the controller could compute them. If all of the control updates took less than the sample time ($\tau_k < T$), then k_f is defined as the last sample in the batch ($k_f := k_0 + c$) or the simulation ($k_f := K$), whichever is first.[2]

In Step 5, SHARC checks whether there was a missed computation ($\tau_{k_f} > T$), revises the simulation trajectory accordingly, and then continues to the next batch. If $\tau_{k_f} \leq T$, then controller has computed each update within the sample time, so the simulation either moves to the next batch with $k_0 = k_f + 1$ (if $k_f < K$) or terminates (if $k_f = K$). On the other hand, if $\tau_{k_f} > T$, then the control computation that was started at t_{k_f} will not be available at t_{k_f+1}, violating the assumption in Step 1. Thus, SHARC must recompute the system's trajectory starting from x_{k_f} using $u = u_{k_f}$ until the control update finishes. As in Section 7.2.1, let $\tilde{t}_{k_f} := t_{k_f} + \tau_{k_f}$ (which typically is not a sample time) and let $\tilde{\kappa}$ be the smallest integer such that $\tilde{t}_{k_f} \leq \tilde{\kappa} T$. In other words, $\tilde{\kappa} T$ is the first sample time after the computation that started at t_{k_f} finishes. We recompute the portion of the trajectory computed in Step 1 from k_f to $\tilde{\kappa}$ according to (7.3a) with $u_k = u_{k_f}$ held constant:

$$x_{k+1} = f\big(t_k, x_k, u_{k_f}, w(t_k)\big) \quad \forall k \in \{k_f, k_f + 1, \ldots, \tilde{\kappa} - 1\}.$$

Then, the simulation moves to the next batch, using $k_0 := \tilde{\kappa}$ or terminates (if $\tilde{\kappa} \geq K$). The simulation from k_f to $\tilde{\kappa}$ does not require using Scarab since we already know the end time of the pending computation, so it can be computed quickly without parallelization. At $\tilde{\kappa}$, however, the controller will start computing a new control value, so SHARC starts a new batch of parallelization.

Due to the large slowdown incurred by using Scarab, parallelization is important for simulating computationally intensive control algorithms, but parallelization somewhat reduces the fidelity of the simulation. In particular, the parallelized mode is somewhat less accurate in determining computation times because running each time step in a separate process prevents the simulator from accounting for some sequential computational effects between time steps, such as memory caching. In contrast, running SHARC in serial mode allows transient memory effects to persist

[2] In practice, SHARC truncates any batches that would extend past K.

between time steps. The results, however, of Section 7.4.1, below, show that there is only a small difference between the delays calculated by the parallelized and serial modes. Given the large reduction in simulation durations, the trade-off between accuracy and speed often justifies the use of the parallelized mode.

Parallelization is useful for mitigating the 10,000× slowdown incurred by simulating the controller with Scarab. The speedup in the parallelized approach, compared to the serial mode comes from Steps 2 and 3 in Figure 7.5. To quantify the possible improvements gained by parallelizing, Theorem 7.1 describes how much simulation time is reduced by using the parallel approach instead of the serial approach. In particular, it examines how much time it takes to compute N many jobs parallelized across c many CPU cores. In this case, each job corresponds to running Scarab once to determine the computation time of a control input at a particular time step.

Theorem 7.1. *Consider a computational system managing N jobs, each requiring a fixed amount of time—defined as one unit of time—to execute on a single CPU core. The system employs c CPU cores for parallelization where $c \leq N$. Assume that the probability of failure for each job is i.i.d. with probability p and that the system restarts from the job index $k+1$ after each failure at job index k. Then, the average time $\bar{T}$ to complete all the jobs is*

$$\bar{T}(c,p) = \frac{Np}{1-(1-p)^c}. \tag{7.4}$$

Proof. The computational process described herein constitutes a Bernoulli process as it consists of a sequence of independent binary random variables representing job success or failure. To analyze this, we calculate the average number of completed jobs, denoted as $\bar{K}$, for each parallel task. A closed-form expression for $\bar{K}$ is derived as follows:

$$\begin{aligned} \bar{K} &= \sum_{i=0}^{c-1}(i+1)\cdot \Pr\{K=i\} + c\cdot \Pr\{K=c\}, \\ &= p\cdot \sum_{i=0}^{c-1}(i+1)\cdot(1-p)^i + c\cdot(1-p)^c = \frac{1-(1-p)^c}{p}. \end{aligned} \tag{7.5}$$

This result leads to the expression for the average time to complete all N jobs:

$$\bar{T}(c,p) = \frac{N}{\bar{K}} = \frac{Np}{1-(1-p)^c}. \tag{7.6}$$

□

Note that $\bar{T}(1,p) = N$, reflecting the case when all jobs are processed sequentially, and $\lim_{p\to 0} \bar{T}(N,p) = 1$, aligning with the expectation that in the absence of failures, the system completes all jobs in unit time. The parallelization gain, which is the speedup factor for running N jobs in parallel on c cores instead of running N jobs sequentially on one core, is

$$\delta(c,p) \triangleq \frac{\bar{T}(1,p)}{\bar{T}(c,p)} = \frac{N}{\left(\frac{Np}{1-(1-p)^c}\right)} = \frac{1-(1-p)^c}{p}. \tag{7.7}$$

Thus, when using c cores, the parallel approach is faster than the serial approach by a factor of $(1-(1-p)^c)/p$. In the ideal case with unlimited tasks and unlimited computational resources, $\lim_{c\to\infty} \delta(c,p) = 1/p$. Therefore, when using SHARC's parallel mode to simulate a system that has a uniform probability p at each time step of the control delay τ being larger than T, the expected parallelization speedup is never better than $1/p$, regardless of how many CPUs are used to parallelize the simulation.

7.4 Numerical Experiments

In this section, we present two examples of the SHARC simulator applied to systems using a model predictive controller (MPC). In Section 7.4.1, MPC is used for adaptive cruise control of a vehicle on a roadway. The resulting MPC problem is linear and thus can be solved efficiently, allowing for the serial mode of SHARC to run simulations in a reasonable time. We provide a comparison with the parallelized mode to demonstrate the similarity of the results.

7.4.1 Adaptive Cruise Control

In this section, we present an example of applying SHARC to an adaptive cruise control (ACC) system used for longitudinal control of a vehicle on a highway. The dynamical model used in this example is adapted from [76]. In particular, we consider an ego vehicle velocity v and a desired velocity $v_{\text{des}} := 15\,\text{m/s}$. The ego vehicle is following a public *front* vehicle that has velocity $t \mapsto v_{\text{F}}(t)$ that we do not control and is considered as an exogenous input to the system. The headway from the front of the ego vehicle to the rear of the front vehicle is denoted h.

The acceleration of the ego vehicle is

$$\dot{v} = \frac{1}{M}\Big(u^{\mathsf{a}} - u^{\mathsf{b}} - F\Big),$$

where M is the mass of the vehicle, u^{a} is acceleration force, u^{b} is braking force, and F is a resistive force on the ego vehicle due to drag and friction. The controlled quantities are u^{a} and u^{b}. Assuming travel on a level roadway, $v \mapsto F(v) := \beta + \gamma v^2$,where $\beta \geq 0$ and $\gamma \geq 0$ are determined empirically. Values for τ, β, γ, and M can be found in [76, Table 1]. The resulting dynamics are

$$\dot{h} = v_{\mathrm{F}}(t) - v \tag{7.8}$$

$$\dot{v} = \frac{1}{M}\Big(u^{\mathsf{a}} - u^{\mathsf{b}} - F(v)\Big). \tag{7.9}$$

The quadratic friction term $F(v)$ makes the system nonlinear, so we linearize F around $v_0 \geq 0$ as $v \mapsto (\beta - \gamma v_0^2) + 2\gamma v_0 v$. The state of the system is $x := (h, v) \in \mathbb{R}^2$ and the input is $u := (u^{\mathsf{a}}, u^{\mathsf{b}}) \in \mathbb{R}^2$. We write the exogenous inputs to the system as $w := (v_{\mathrm{F}}, 1) \in \mathbb{R}^2$, where v_{F} is the velocity v_{F} of the front vehicle, and "1" in the second component is used to incorporate a constant term $(\gamma v_0^2 - \beta)/M$ arising from F. The resulting system is

$$\dot{x} = \begin{bmatrix} 0 & -1 \\ 0 & -2\gamma v_0/M \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 1/M & -1/M \end{bmatrix} u + \begin{bmatrix} 1 & 0 \\ 0 & (\gamma v_0^2 - \beta)/M \end{bmatrix} w. \tag{7.10}$$

The continuous-time dynamics are discretized with a sample time $T := 0.1\,\mathrm{s}$, resulting in a discrete-time system we write as

$$x_{k+1} = A(v_0)x_k + B(v_0)u_k + B_d(v_0)w_k, \tag{7.11}$$

where x_k, u_k, and w_k are the values of x, u, and w, respectively, at $t = kT$. Note that $A(v_0)$, $B(v_0)$, and $B_d(v_0)$ depend on v_0, the center of the linearization.

7.4.1.1 MPC Problem Formulation

To generate control values at each discrete time $k_0 \in \mathbb{N}$, we apply MPC with a prediction horizon of $N_p \in \mathbb{N}$ time steps. Given any discrete time $k_0 \in \mathbb{N}$, let $\hat{x}_{k_0}$ be a measurement-based estimate of x at k_0, and let $\hat{v}_0$ be the velocity component of $\hat{x}_{k_0}$. For each $k \in \{k_0, k_0 + 1, \ldots, k_0 + N_p\}$, let $k \mapsto u_{k|k_0}$ be planned control values

starting at k_0, and let $k \mapsto x_{k|k_0}$ be the state prediction generated by (7.11) with initial condition $x_{k_0|k_0} = \hat{x}_{k_0}$ and using the control signal $u_{k|k_0}$.

The cost function of the MPC problem is a quadratic function that penalizes the deviance of $v_{k|k_0}$ from the desired velocity v_{des}, the control effort $u_{k|k_0}$, and changes to the control effort, which roughly corresponds to the vehicle's jerk ($\ddot{v}$). A positive definite matrix $R \in \mathbb{R}^{2\times 2}$ defines the control weight matrix and $\alpha \geq 0$ is a jerk penalization parameter.

The ego vehicle must always satisfy the following constraints:

$$\begin{aligned} \text{Headway:} \quad & h \geq h_{\min} := 6\,\text{m} \\ \text{Velocity:} \quad & 0\,\text{m/s} \leq v \leq v_{\max} := 20\,\text{m/s} \\ \text{Acceleration force:} \quad & 0\,\text{N} \leq u^{\mathsf{a}} \leq u^{\mathsf{a}}_{\max} := 4880\,\text{N} \\ \text{Braking force:} \quad & 0\,\text{N} \leq u^{\mathsf{b}} \leq u^{\mathsf{b}}_{\max} := 6507\,\text{N}. \end{aligned}$$

To ensure the headway constraint $h \geq h_{\min}$ is satisfiable past the end of the MPC prediction horizon, we also must include a terminal constraint. In particular, h and v must satisfy

$$h \geq \frac{v^2}{2|a|} - \frac{v_{\text{F}}^2}{2|a_{\text{F}}|} + h_{\min}, \tag{7.12}$$

at the end of the prediction horizon, where $a_{\text{F}} < 0$ is a lower bound on the rate of deceleration of the front vehicle (that is $\dot{v}_{\text{F}} \geq a_{\text{F}}$) and $a < 0$ which is an upper bound on the deceleration of the ego vehicle when maximum braking is applied (that is, $\dot{v} \leq a$ when $u^{\mathsf{a}} = 0$ and $u^{\mathsf{b}} = u^{\mathsf{b}}_{\max}$).

Equation (7.12) is not suitable as a linear MPC constraint, however, because it includes a nonlinear term and depends on the future velocity of the front vehicle, which is unknown. By assuming that the front vehicle applies maximum braking, we estimate its worst-case future velocity as a sequence $k \mapsto \hat{v}_{\text{F}}(k|k_0)$. Then, we use $k \mapsto \hat{w}(k|k_0) := (\hat{v}_{\text{F}}(k|k_0), 1)$ as a worst-case prediction of the w. To remove the v^2 nonlinearity in (7.12), we replace v^2 with $vv_{\max} \geq v^2$, creating a more conservative terminal constraint:

$$h_{(k_0+N_p)|k_0} \geq \frac{v_{\max}}{2|a|} v_{(k_0+N_p)|k_0} - \frac{\hat{v}_{\text{F}}^2(k_0+N_p|k_0)}{2|a_{\text{F}}|} + h_{\min}. \tag{7.13}$$

The resulting MPC problem formulation is shown in Problem 1.

In Figure 7.6, the results of simulating the ACC system are shown with a comparison between the results of serial and parallel simulation schemes. We see that at as the headway decreases, the system hits a point around $t = 3.5\,\text{s}$ when the

Problem 1 (Linear MPC)**.**

$$\begin{aligned}\text{minimize } J\big(x_{(\cdot)|k_0}, u_{(\cdot)|k_0}\big) := &\sum_{k=k_0}^{k_0+N_p} \big(v_{k|k_0} - v_{\text{des}}\big)^2 \\ &+ \sum_{k=k_0}^{k_0+N_p-1} u_{k|k_0}^\top R u_{k|k_0} + \alpha \sum_{k=k_0}^{k_0+N_p-2} \big|u_{k+1|k_0} - u_{k|k_0}\big|^2\end{aligned} \tag{7.14a}$$

with respect to

$$x_{k_0|k_0},\ x_{(k_0+1)|k_0},\ \ldots,\ x_{(k_0+N_p)|k_0} \in \mathbb{R}^2 \tag{7.14b}$$

$$u_{k_0|k_0},\ u_{(k_0+1)|k_0},\ \ldots,\ u_{(k_0+N_p-1)|k_0} \in \mathbb{R}^2 \tag{7.14c}$$

subject to

$$x_{k_0|k_0} = \hat{x}_{k_0}, \tag{7.14d}$$

and for each $k = k_0,\ k_0+1,\ \ldots,\ k_0+N_p-1$,

$$x_{k+1|k_0} = A(\hat{v}_0)x_{k|k_0} + B(\hat{v}_0)u_{k|k_0} + B_d(\hat{v}_0)\hat{w}(k|k_0), \tag{7.14e}$$

and for each $k = k_0,\ k_0+1,\ \ldots,\ k_0+N_p$,

$$0 \le v_{k|k_0} \le v_{\max}, \tag{7.14f}$$

$$0 \le u^{\mathsf{a}}_{k|k_0} \le u^{\mathsf{a}}_{\max}, \tag{7.14g}$$

$$0 \le u^{\mathsf{b}}_{k|k_0} \le u^{\mathsf{b}}_{\max}, \tag{7.14h}$$

$$h_{\min} \le h_{k|k_0}, \tag{7.14i}$$

and for $k = k_0 + N_p$,

$$h_{k|k_0} \ge \frac{v_{\max}}{2|a|} v_{k|k_0} - \frac{\hat{v}_{\text{F}}^2(k|k_0)}{2|a_{\text{F}}|} + h_{\min}. \tag{7.14j}$$

delays significantly increase, rising above the sampling time. This increase causes the updated control values to be delayed by six time steps. The delays increase at this point because more MPC inequality constraints become active, making the optimization problem harder to solve. In this simulation, the vehicle recovers before colliding with the lead vehicle, but if the front vehicle brakes more aggressively, the computational delays could result in a collision.

Figure 7.7 shows a comparison of SHARC simulations using instruction caches of size 1 KB, 8 KB, and 1 MB. We see that computation times increase as the size of the instruction cache shrinks, producing significant deviation in v and h between simulations.

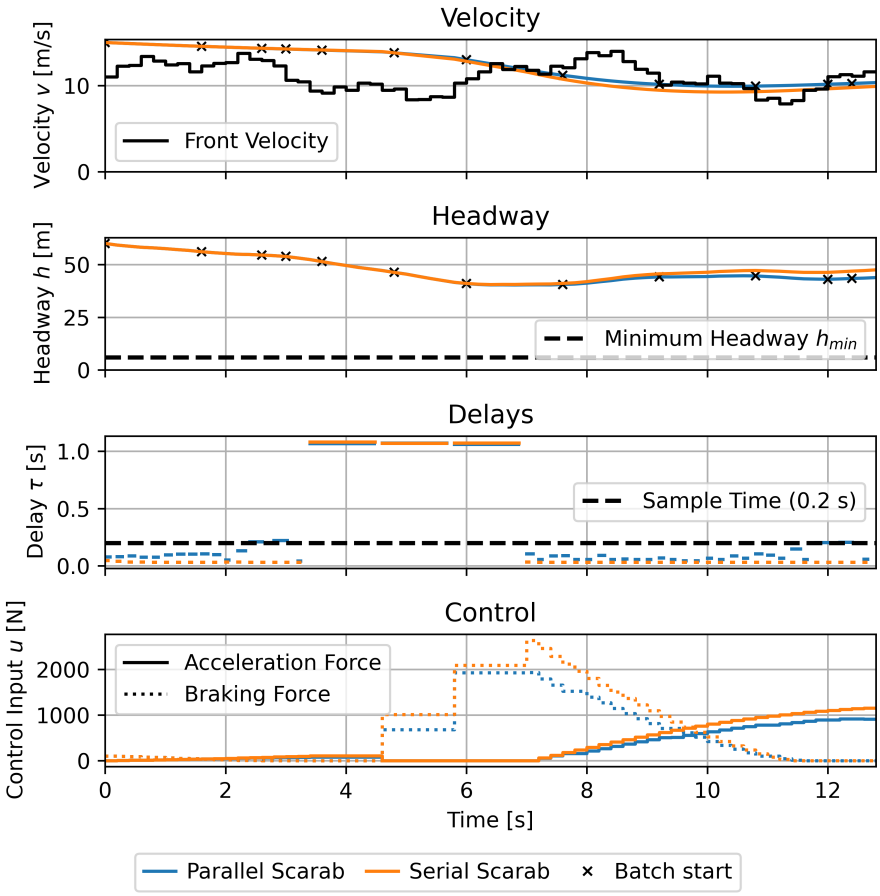

Figure 7.6. Comparison of trajectories for the ACC system from Section 7.4.1 simulated using the serial and parallel modes. In the Delays plot, the horizontal lines extend from the start time of each computation to its completion time.

7.5 Conclusion

In this chapter, we present SHARC as a tool to simulate user-specified control algorithms on a given processor microarchitecture, evaluating how computational constraints affect the performance of the control algorithm and the safety of the physical system. We illustrated the power and usefulness of SHARC via two examples: an adaptive cruise controller implemented with linear MPC and an inverted pendulum system controlled by nonlinear MPC. By providing insight into the impact of computing hardware on the performance of a CPS, SHARC allows for the co-design of control algorithms and the computational hardware on which they are run. Future

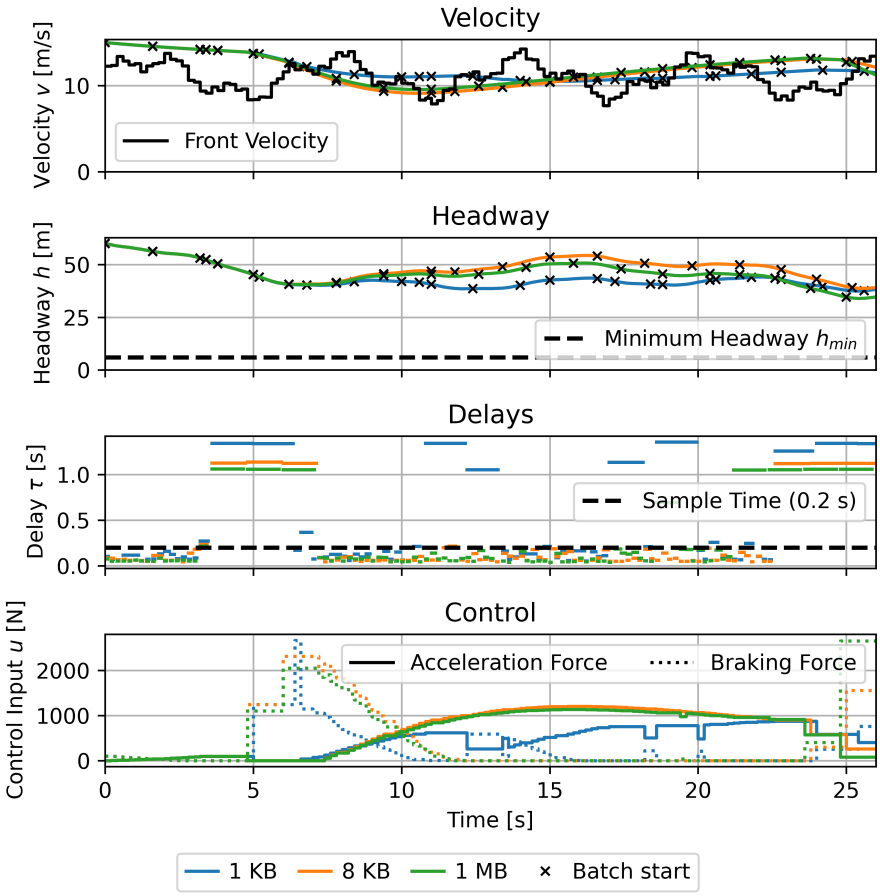

Figure 7.7. Comparison of SHARC simulations for the ACC system in Section 7.4.1 using various sizes of instruction cache.

work includes 1) using SHARC to identify common bottlenecks in particular classes of control algorithms and computational hardware and 2) developing an automated framework for jointly optimizing the parameters of the hardware and the control algorithm.

Bibliography

[1] J.-P. Aubin and H. Frankowska, "Tangent Cones," in *Set-Valued Analysis*, ser. Modern Birkhäuser Classics, Boston: Birkhäuser, 2009, pp. 117–177.

[2] F. H. Clarke, *Optimization and Nonsmooth Analysis* (Classics in Applied Mathematics), 2nd ed. SIAM, 1990, 318 pp.

[3] R. G. Sanfelice, *Hybrid Feedback Control*. Princeton University Press, 2021.

[4] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, 2012, 212 pp. JSTOR: j.ctt7s02z.

[5] M. Maghenem and R. G. Sanfelice, "Sufficient conditions for forward invariance and contractivity in hybrid inclusions using barrier functions," *Automatica*, vol. 124, Feb. 2021.

[6] M. Nagumo, "Über die Lage der Integralkurven gewöhnlicher differentialgleichungen," *Proc. Phys.-Math. Soc. Jpn.*, 3rd ser., vol. 24, pp. 551–559, 1942.

[7] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Hybrid Syst. Comput. Control*, R. Alur and G. J. Pappas, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2004, pp. 477–492.

[8] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. 18th Eur. Control Conf.*, Jun. 2019, pp. 3420–3431.

[9] K. Zhang, J. Sprinkle, and R. G. Sanfelice, "Computationally Aware Switching Criteria for Hybrid Model Predictive Control of Cyber-Physical Systems," *IEEE Trans. Automat. Sci. Eng.*, vol. 13, no. 2, pp. 479–490, Apr. 2016.

[10] J. G. Rivera, A. A. Danylyszyn, C. B. Weinstock, L. R. Sha, and M. J. Gagliardi, "An architectural description of the Simplex Architecture," Defense Technical Information Center, Fort Belvoir, VA, Mar. 1, 1996.

[11] J. Yang, M. A. Islam, A. Murthy, S. A. Smolka, and S. D. Stoller, "A Simplex architecture for hybrid systems using barrier certificates," in *Comput. Saf. Reliab. Secur.*, S. Tonetta, E. Schoitsch, and F. Bitsch, Eds., ser. Lecture Notes in Computer Science, Springer International Publishing, 2017, pp. 117–131.

[12] A. Damare, S. Roy, S. A. Smolka, and S. D. Stoller, "A barrier certificate-based Simplex architecture with application to microgrids," in *Runtime Verification*, T. Dang and V. Stolz, Eds., ser. Lecture Notes in Computer Science, Springer International Publishing, 2022, pp. 105–123.

[13] P. K. Wintz and R. G. Sanfelice, "Forward invariance-based hybrid control using uncertified controllers," in *2023 62nd IEEE Conf. Decis. Control CDC*, Singapore, Singapore: IEEE, Dec. 2023, pp. 864–869.

[14] R. G. Sanfelice, D. A. Copp, and P. Nanez, "A toolbox for simulation of hybrid systems in MATLAB/Simulink: Hybrid Equations (HyEQ) Toolbox," in *Proc. 16th Int. Conf. Hybrid Syst. Comput. Control*, Philadelphia: ACM Press, 2013, pp. 101–106.

[15] A. D. Ames, A. Abate, and S. Sastry, "Sufficient conditions for the existence of zeno behavior in a class of nonlinear hybrid systems via constant approximations," in *Proc. IEEE Conf. Decis. Control*, New Orleans, LA, USA: IEEE, 2007, pp. 4033–4038.

[16] P. K. Wintz and R. G. Sanfelice, "Relaxed lyapunov conditions," in *Am. Control Conf.*, Denver, CO, USA, Jul. 2025.

[17] E. D. Sontag, "Input-to-State Stability," in *Encyclopedia of Systems and Control*, J. Baillieul and T. Samad, Eds., London: Springer London, 2013.

[18] D. N. Tran, B. S. Ruffer, and C. M. Kellett, "Incremental stability properties for discrete-time systems," in *IEEE 55th Conf. Decis. Control*, Las Vegas, NV, USA: IEEE, Dec. 2016, pp. 477–482.

[19] H. K. Khalil, *Nonlinear Systems*, Third. Pearson, 2002.

[20] K. Yamazaki, "The range of maps on classical insertion theorems," *Acta Mathematica Hungarica*, vol. 132, no. 1, pp. 42–48, Jul. 1, 2011.

[21] R. A. Freeman and P. Kokotović, *Robust Nonlinear Control Design*. Boston, MA: Birkhäuser Boston, 1996.

[22] F. H. Clarke, Yu. S. Ledyaev, R. J. Stern, and R. R. Wolenski, *Nonsmooth Analysis and Control Theory* (Graduate Texts in Mathematics). New York, NY: Springer, 1998, vol. 178.

[23] P. K. Wintz and R. G. Sanfelice, "Conical transition graphs for analysis of asymptotic stability in hybrid dynamical systems," in *8th IFAC Conf. Anal. Des. Hybrid Syst.*, vol. 58, IFAC, Jul. 1, 2024, pp. 159–164.

[24] R. Goebel and A. R. Teel, "Preasymptotic stability and homogeneous approximations of hybrid dynamical systems," *SIAM Rev.*, vol. 52, no. 1, pp. 87–109, 2010. JSTOR: 25662361.

[25] P. Prabhakar and M. Garcia Soto, "Abstraction Based Model-Checking of Stability of Hybrid Systems," in *Comput. Aided Verification*, N. Sharygina and H. Veith, Eds., Berlin, Heidelberg: Springer, 2013, pp. 280–295.

[26] S. E. Tuna and A. R. Teel, "Homogeneous hybrid systems and a converse lyapunov theorem," in *Proc. IEEE Conf. Decis. Control*, Dec. 2006, pp. 6235–6240.

[27] R. Goebel and R. G. Sanfelice, "Pointwise asymptotic stability in a hybrid system and well-posed behavior beyond Zeno," *SIAM J. Control Optim.*, vol. 56, no. 2, pp. 1358–1385, Jan. 2018.

[28] A. Lamperski and A. D. Ames, "Lyapunov-like conditions for the existence of zeno behavior in hybrid and lagrangian hybrid systems," in *Proc. IEEE Conf. Decis. Control*, Dec. 2007, pp. 115–120.

[29] S. Nersesov, V. Chellaboina, and W. Haddad, "A generalization of Poincare's theorem to hybrid and impulsive dynamical systems," in *Proc. Am. Control Conf.*, vol. 2, May 2002, pp. 1240–1245.

[30] X. Lou, Y. Li, and R. G. Sanfelice, "On robust stability of limit cycles for hybrid systems with multiple jumps," in *Proc. IFAC Conf. Anal. Des. Hybrid Syst.*, vol. 48, Atlanta, GA, USA: IFAC, Jan. 1, 2015, pp. 199–204.

[31] B. Morris and J. W. Grizzle, "Hybrid Invariant Manifolds in Systems With Impulse Effects With Application to Periodic Locomotion in Bipedal Robots," *IEEE Trans. Automat. Contr.*, vol. 54, no. 8, pp. 1751–1764, Aug. 2009.

[32] M. Philippe, R. Essick, G. E. Dullerud, and R. M. Jungers, "Stability of discrete-time switching systems with constrained switching sequences," *Automatica*, vol. 72, pp. 242–250, Oct. 1, 2016.

[33] A. Kundu and D. Chatterjee, "A graph theoretic approach to input-to-state stability of switched systems," *Eur. J. Control*, vol. 29, pp. 44–50, May 1, 2016.

[34] R. Langerak and J. Polderman, "Tools for stability of switching linear systems: Gain automata and delay compensation," in *Proc. IEEE Conf. Decis. Control*, Dec. 2005, pp. 4867–4872.

[35] S. Bogomolov, M. Giacobbe, T. A. Henzinger, and H. Kong, "Conic Abstractions for Hybrid Systems," in *Formal Modeling and Analysis of Timed Systems*, A. Abate and G. Geeraerts, Eds., vol. 10419, Cham: Springer International Publishing, 2017, pp. 116–132.

[36] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," California Institute of Technology, May 7, 2004.

[37] A. Papachristodoulou and S. Prajna, "On the construction of Lyapunov functions using the sum of squares decomposition," in *Proc. IEEE Conf. Decis. Control*, vol. 3, Dec. 2002, pp. 3482–3487.

[38] S. Kundu and M. Anghel, "Stability and control of power systems using vector lyapunov functions and sum-of-squares methods," in *Eur. Control Conf.*, Jul. 2015, pp. 253–259.

[39] S. Kundu and M. Anghel, "A sum-of-squares approach to the stability and control of interconnected systems using vector Lyapunov functions," in *Proc. Am. Control Conf.*, Jul. 2015, pp. 5022–5028.

[40] C. Murti, "Analysis of Zeno stability in hybrid systems using sum-of-squares programming," M.S. thesis, Illinois Institute of Technology, 2012.

[41] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "HyTech: A model checker for hybrid systems," in *Comput. Aided Verification*, O. Grumberg, Ed., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 1997, pp. 460–463.

[42] G. Frehse, "PHAVer: Algorithmic Verification of Hybrid Systems Past HyTech," in *Hybrid Syst. Comput. Control*, M. Morari and L. Thiele, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2005, pp. 258–273.

[43] E. Asarin, T. Dang, and O. Maler, "The d/dt tool for verification of hybrid systems," vol. 3, Jan. 1, 2002, pp. 365–370.

[44] R. Diestel, "The Basics," in *Graph Theory*, ser. Graduate Texts in Mathematics, 5th ed., vol. 173, Berlin, Heidelberg: Springer, 2017.

[45] R. T. Farouki, H. P. Moon, and B. Ravani, "Minkowski geometric algebra of complex sets," *Geom. Dedicata*, vol. 85, pp. 283–315, 2001.

[46] D. B. Johnson, "Finding all the elementary circuits of a directed graph," *SIAM J. Comput.*, vol. 4, no. 1, pp. 77–84, Mar. 1975.

[47] K. G. Larsen and W. Yi, "Time abstracted bisimulation: Implicit specifications and decidability," in *Mathematical Foundations of Programming Semantics*, S. Brookes, M. Main, A. Melton, M. Mislove, and D. Schmidt, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 160–176.

[48] M. Henzinger, T. Henzinger, and P. Kopke, "Computing simulations on finite and infinite graphs," in *Proceedings of IEEE 36th Annual Foundations of Computer Science*, 1995, pp. 453–462.

[49] P. K. Wintz et al., "SHARC: Simulator for hardware architecture and real-time control," in *Proceedings of the 28th ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '25, Irvine, CA, USA: Association for Computing Machinery, 2025.

[50] P. Leteinturier, S. Brewerton, and K. Scheibert, "MultiCore Benefits & Challenges for Automotive Applications," SAE International, Warrendale, PA, SAE Technical Paper 2008-01-0989, Apr. 14, 2008.

[51] L. M. Kinnan, "Use of multicore processors in avionics systems and its potential impact on implementation and certification," in *2009 IEEEAIAA 28th Digit. Avion. Syst. Conf.*, Oct. 2009, 1.E.4-1-1.E.4–6.

[52] D. Pradhan, "Multicore processors bring innovation to medical imaging," Texas Instruments, May 2010.

[53] R. N. Charette, "This Car Runs on Code," *IEEE Spectrum*, Feb. 1, 2009.

[54] A. Bonci, S. Longhi, G. Nabissi, and G. A. Scala, "Execution Time of Optimal Controls in Hard Real Time, a Minimal Execution Time Solution for Nonlinear SDRE," *IEEE Access*, vol. 8, pp. 158 008–158 025, 2020.

[55] M. Caccamo, G. Buttazzo, and L. Sha, "Handling execution overruns in hard real-time control systems," *IEEE Trans. Comput.*, vol. 51, no. 7, pp. 835–849, Jul. 2002.

[56] D. Arnström, D. Broman, and D. Axehill, "Exact worst-case execution-time analysis for implicit model predictive control," *IEEE Trans. Autom. Control*, vol. 69, no. 10, pp. 7190–7196, Oct. 2024.

[57] D. Arnström, *Real-Time Certified MPC: Reliable Active-Set QP Solvers*. Linköping: Department of Electrical Engineering, Linköping University, 2023, 190 pp.

[58] Y. V. Pant, H. Abbas, K. Mohta, T. X. Nghiem, J. Devietti, and R. Mangharam, "Co-design of Anytime Computation and Robust Control," in *IEEE Real-Time Syst. Symp.*, San Antonio, TX, USA: IEEE, Dec. 2015, pp. 43–52.

[59] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms* (Communications and Control Engineering). London: Springer London, 2011.

[60] V. M. Zavala and L. T. Biegler, "The advanced-step NMPC controller: Optimality, stability and robustness," *Automatica*, vol. 45, no. 1, pp. 86–93, Jan. 2009.

[61] N. Binkert et al., "The gem5 simulator," *SIGARCH Comput Arch. News*, vol. 39, no. 2, pp. 1–7, Aug. 31, 2011.

[62] N. Gober et al. "The Championship Simulator: Architectural Simulation for Education and Competition." arXiv: `2210.14324 [cs]`, Accessed: Mar. 11, 2025, pre-published.

[63] D. Sanchez and C. Kozyrakis, "ZSim: Fast and accurate microarchitectural simulation of thousand-core systems," *SIGARCH Comput Arch. News*, vol. 41, no. 3, pp. 475–486, Jun. 23, 2013.

[64] K. Vollmar and P. Sanderson, "MARS: An education-oriented MIPS assembly language simulator," *SIGCSE Bull.*, vol. 38, no. 1, pp. 239–243, Mar. 3, 2006.

[65] T. E. Carlson, W. Heirman, S. Eyerman, I. Hur, and L. Eeckhout, "An Evaluation of High-Level Mechanistic Core Models," *ACM Trans Arch. Code Optim*, vol. 11, no. 3, 28:1–28:25, Aug. 25, 2014.

[66] M. Ishikawa, D. McCune, G. Saikalis, and S. Oho, "CPU Model-Based Hardware/Software Co-design, Co-simulation and Analysis Technology for Real-Time Embedded Control Systems," in *IEEE Real Time Embed. Technol. Appl. Symp.*, IEEE, Apr. 2007, pp. 3–11.

[67] M. Torngren, D. Henriksson, K.-E. Arzen, A. Cervin, and Z. Hanzalek, "Tool supporting the co-design of control systems and their real-time implementation: Current status and future directions," in *IEEE Conf. Comput. Aided Control Syst. Des.*, Munich, Germany: IEEE, Oct. 2006, pp. 1173–1180.

[68] C. Faure, M. Ben Gaid, N. Pernet, M. Fremovici, G. Font, and G. Corde, "Methods for real-time simulation of cyber-physical systems: Application to automotive domain," in *2011 7th Int. Wirel. Commun. Mob. Comput. Conf.*, Jul. 2011, pp. 1105–1110.

[69] F. MihaliUnexpected case., M. TruntiUnexpected case., and A. Hren, "Hardware-in-the-loop simulations: A historical overview of engineering challenges," *Electronics*, vol. 11, no. 15, p. 2462, Aug. 8, 2022.

[70] *Scarab*.

[71] S. Oh, M. Xu, T. A. Khan, B. Kasikci, and H. Litz, "UDP: Utility-Driven Fetch Directed Instruction Prefetching," in *ACMIEEE 51st Annu. Int. Symp. Comput. Archit.*, IEEE, Jun. 2024, pp. 1188–1201.

[72] A. Seznec and P. Michaud, "A case for (partially) tagged geometric history length branch prediction," *J. Instr.-Level Parallelism*, vol. 8, 2006.

[73] R. M. Tomasulo, "An Efficient Algorithm for Exploiting Multiple Arithmetic Units," *IBM J. Res. Dev.*, vol. 11, no. 1, pp. 25–33, Jan. 1967.

[74] Y. Kim, W. Yang, and O. Mutlu, "Ramulator: A Fast and Extensible DRAM Simulator," *IEEE Comput. Archit. Lett.*, vol. 15, no. 1, pp. 45–49, Jan. 2016.

[75] D. Bruening, T. Garnett, and S. Amarasinghe, "An infrastructure for adaptive dynamic optimization," in *Proc Int. Symp. Code Gener. Optim. Feedback-Dir. Runtime Optim.*, ser. CGO '03, USA: IEEE Computer Society, Mar. 23, 2003, pp. 265–275.

[76] S. W. Smith et al., "Improving Urban Traffic Throughput With Vehicle Platooning: Theory and Experiments," *IEEE Access*, vol. 8, pp. 141 208–141 223, 2020.

Appendix A

Hybrid Equations Toolbox

The Hybrid Equation (HyEQ) Toolbox provides methods in MATLAB and Simulink for computing and plotting numerical solutions to hybrid dynamical systems. During the beginning of my PhD studies, I rewrote a large portion of the HyEQ Toolbox to improve the toolbox's design and capabilities to aid in the simulation, analysis of hybrid systems and the plotting of hybrid arcs. Most of the examples throughout this dissertation were simulated and plotted an using this toolbox.

New Features in Version 3.0

This chapter summarizes my contributions to the HyEQ Toolbox, released in versions 3.0 and 3.1 of the HyEQ Toolbox.

HyEQ MATLAB Library

Object-Oriented Definitions of Hybrid Systems. A hybrid system can now be defined in a single file by creating a subclass of the `HybridSystem` class. This allows for the definition of multiple hybrid systems without name conflicts and enables the definition of system parameters without using global variables.

Interconnected Hybrid Systems. It is now possible, in MATLAB, to define several hybrid subsystem systems with inputs and outputs, such as a plant and a controller, then link them together to form a composite hybrid system. Solutions to the composite system can be computed like any other system.

More Informative Solutions. The new `HybridSolution` class includes additional useful information about solutions such as the duration of each interval of flow

and the reason the solution terminated. Methods are provided for modifying solution objects by, e.g., applying a transformation the state values or truncating the time span.

Improved Progress Updates. While computing solutions, a progress bar displays the percent completed and the current hybrid time. The progress updates during both flows and at jumps (in v2.04, progress updates were only printed to the command line at jumps).

Improved Plotting. Plotting hybrid arcs is easier and allows more control over the appearance of plots. New features include:

- Easy control of the marker and line styles for flows and jumps.
- Support for legends.
- Ability to hide portions of hybrid arcs from plots using a filter (useful for plotting different modes in different styles).
- Automatic creation of subplots for hybrid arcs with multiple components.
- Ability to set default plot settings.

Plotting methods are up to 200x faster than in v2.04 for hybrid arcs with many jumps.

Validation and error reporting. New error checking features catch programming mistakes earlier when using the toolbox. Over 350 automated tests verify the correctness of the toolbox's code.

Code Autocompletion. The Hybrid Equations Toolbox supports MATLAB's autocompletion feature (introduced to the MATLAB code editor in R2021b).

HyEQ Simulink Library

The following improvements were made to the Simulink-based Hybrid System solver:

Hybrid System with External Functions and Inputs. A new Simulink block allows for a hybrid system with an input to be defined using plaintext `.m` MATLAB function files to specify f, g, C, and D.

Block Parameters. Simulink block masks were added to the HyEQ blocks to allow users to set block parameters without needing to modify anything inside the block. Parameters are now set in a popup dialog that opens when each block is clicked.

Instructions for How To Use Blocks. Each block in the HyEQ Simulink library now includes instructions in the block's popup dialog that explains how to use the block.

Signal Sizes. HyEQ Simulink library blocks now check the signal sizes for inputs and outputs to help identify errors and aid in debugging.

General Improvements

The following updates apply to the entire toolbox:

Easier Installation and Updates. Version 3.0 is packaged using MATLAB's toolbox packaging, so it can be installed and updated automatically through MATLAB's Add-on manager.

Backward Compatibility. All code that works in Toolbox version 2.04 is expected to work in v3.0 without modification. Version 3.0 is compatible with—and tested on—MATLAB versions back to R2014b.

Improved Help Files and Example. All documentation for the HyEQ Toolbox has been redone to make it easier to access and navigate in MATLAB Help.

Appendix B

Additional Results and Proofs

B.1 Additional Proofs from Chapter 4

This section contains proofs that were omitted from Chapter 4.

B.1.1 Proof of Lemma 4.1 (Hybrid Basic Conditions)

For clarity, we split the proof of Lemma 4.1 into several parts.

Proof that C and D are closed. The sets $C_{\mathrm{P}}^{\mathrm{CL}}$ and $D_{\mathrm{P}}^{\mathrm{CL}}$ are defined in (4.15) as set closures. The sets $C_{\mathrm{K}_0}^{\mathrm{CL}}$, $C_{\mathrm{K}_1}^{\mathrm{CL}}$, $C_{\mathrm{S}}^{\mathrm{CL}}$, $D_{\mathrm{K}_0}^{\mathrm{CL}}$, $D_{\mathrm{K}_1}^{\mathrm{CL}}$, and $D_{\mathrm{S}}^{\mathrm{CL}}$ are closed because they are defined as equal to or as the union and Cartesian product of C_{K_0}, D_{K_0}, D_{K_1}, C_{K_1}, C_{S}, D_{S}, $\mathcal{E}_1$, and $\mathcal{V}$, which are closed by assumption (B2), as well as the finite set $\{0,1\}$, which is also closed. Therefore, C and D are closed. □

Proof of $\mathrm{dom}(\widehat{F}) \supset C$ *and* $\mathrm{dom}(\widehat{G}) \supset D$. Take any $x := (z, \eta_0, \eta_1, v, q) \in C$. Since $C = C_{\mathrm{P}}^{\mathrm{CL}} \cap C_{\mathrm{K}_0}^{\mathrm{CL}} \cap C_{\mathrm{K}_1}^{\mathrm{CL}} \cap C_{\mathrm{S}}^{\mathrm{CL}}$, we have that x is in $C_{\mathrm{P}}^{\mathrm{CL}}$, $C_{\mathrm{K}_0}^{\mathrm{CL}}$, $C_{\mathrm{K}_1}^{\mathrm{CL}}$, and $C_{\mathrm{S}}^{\mathrm{CL}}$. We want to show $x \in \mathrm{dom}(\widehat{F}) = \bigcap \mathrm{dom}(F_\star^{\mathrm{CL}})$. To do this, we will show for each $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$ that $x \in \mathrm{dom}(F_\star^{\mathrm{CL}})$, since

$$\mathrm{dom}(F_\star^{\mathrm{CL}}) \subset \mathrm{dom}(\widehat{F_\star^{\mathrm{CL}}}).$$

For $\star = \mathrm{P}$, we have that x is in $C_{\mathrm{P}}^{\mathrm{CL}}$, which is defined as $\mathrm{dom}(F_{\mathrm{P}}^{\mathrm{CL}})$ in (4.8), so

$$x \in C_{\mathrm{P}}^{\mathrm{CL}} = \mathrm{dom}(F_{\mathrm{P}}^{\mathrm{CL}}) \subset \mathrm{dom}(\widehat{F_{\mathrm{P}}^{\mathrm{CL}}}).$$

For $\star = \mathrm{K}_0$, we have $C_{\mathrm{K}_0}^{\mathrm{CL}} = C_{\mathrm{K}_0}^{\mathrm{CL}}$ (by definition), so $x \in C_{\mathrm{K}_0}^{\mathrm{CL}} = C_{\mathrm{K}_0}^{\mathrm{CL}}$. From the definition of $C_{\mathrm{K}_0}^{\mathrm{CL}}$ in (4.8), we see that $x = (z, \eta_0, \eta_1, v, q) \in C_{\mathrm{K}_0}^{\mathrm{CL}} = C_{\mathrm{K}_0} \times \mathcal{E}_1 \times \mathcal{V} \times \{0,1\}$,

so (z, η_0) is in C_{K_0}. By (B3), we have that $\mathrm{dom}(F_{\mathrm{K}_0}) = C_{\mathrm{K}_0}$, so $(z, \eta_0) \in \mathrm{dom}(F_{\mathrm{K}_0})$, so $F_{\mathrm{K}_0}(z, \eta_0) = F^{\mathrm{CL}}_{\mathrm{K}_0}(x) \neq \varnothing$, from the definition of $F^{\mathrm{CL}}_{\mathrm{K}_0}$ in (4.4). Thus,

$$x \in \mathrm{dom}(F^{\mathrm{CL}}_{\mathrm{K}_0}) \subset \mathrm{dom}(\widehat{F^{\mathrm{CL}}_{\mathrm{K}_0}}).$$

For $\star = \mathrm{K}_1$, we have that $x \in C^{\mathrm{CL}}_{\mathrm{K}_1} = C^{\mathrm{CL}}_{\mathrm{K}_1} = C_{\mathrm{K}_1}$ (per (4.8)). By (B3), we have that $\mathrm{dom}(F_{\mathrm{K}_1}) = C_{\mathrm{K}_1}$, so

$$x \in \mathrm{dom}(F_{\mathrm{K}_1}) = \mathrm{dom}(F^{\mathrm{CL}}_{\mathrm{K}_1}) = \mathrm{dom}(\widehat{F^{\mathrm{CL}}_{\mathrm{K}_1}}).$$

For $\star = \mathrm{S}$, we have $x \in C^{\mathrm{CL}}_{\mathrm{S}} = C^{\mathrm{CL}}_{\mathrm{S}} = C_{\mathrm{S}}$. By (B3), $\mathrm{dom}(f_{\mathsf{v}}) = C_{\mathrm{S}}$, so $x \in \mathrm{dom}(f_{\mathsf{v}})$. The domain of $F^{\mathrm{CL}}_{\mathrm{S}}$ is defined as $\mathrm{dom}(f_{\mathsf{v}})$, so

$$x \in C_{\mathrm{S}} = \mathrm{dom}(f_{\mathsf{v}}) = \mathrm{dom}(F^{\mathrm{CL}}_{\mathrm{S}}) = \mathrm{dom}(\widehat{F^{\mathrm{CL}}_{\mathrm{S}}}).$$

Therefore, $x \in \mathrm{dom}(\widehat{F})$, so $\mathrm{dom}(\widehat{F}) \supset C$.

By a similar process, one can show that $\mathrm{dom}(\widehat{G}) \supset D$. □

Proof that $\widehat{F}$ and $\widehat{G}$ are OSC. To show $\widehat{F}$ and $\widehat{G}$ are OSC we must show that $\widehat{F^{\mathrm{CL}}_{\star}}$ and $\widehat{G^{\mathrm{CL}}_{\star}}$ are OSC for each $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$. By (B4), f_{v} and g_{v} are continuous, and F_{K_0} and G_{K_0} are OSC. It follows directly that

$$\begin{aligned}
x &\mapsto \widehat{F^{\mathrm{CL}}_{\mathrm{K}_0}}(x) = F^{\mathrm{CL}}_{\mathrm{K}_0}(x) = F_{\mathrm{K}_0}(z, \eta_0) \\
x &\mapsto \widehat{G^{\mathrm{CL}}_{\mathrm{K}_0}}(x) = G^{\mathrm{CL}}_{\mathrm{K}_0}(x) = G_{\mathrm{K}_0}(z, \eta_0) \\
x &\mapsto \widehat{F^{\mathrm{CL}}_{\mathrm{S}}}(x) = F^{\mathrm{CL}}_{\mathrm{S}}(x) = \begin{bmatrix} f_{\mathsf{v}}(x) \\ 0 \end{bmatrix} \\
x &\mapsto \widehat{G^{\mathrm{CL}}_{\mathrm{S}}}(x) = G^{\mathrm{CL}}_{\mathrm{S}}(x) = \begin{bmatrix} g_{\mathsf{v}}(x) \\ 0 \end{bmatrix}
\end{aligned}$$

are OSC. Additionally, by (B4),

$$(z, \eta_0) \mapsto F_{\mathrm{P}}(z, \kappa_0(z, \eta)) \quad \text{and} \quad (z, \eta_0) \mapsto G_{\mathrm{P}}(z, \kappa_0(z, \eta))$$

are OSC. When restricted to $x \in \mathcal{X}_0$, we have that

$$\widehat{F^{\mathrm{CL}}_{\mathrm{P}}}(x) = F_{\mathrm{P}}(x) = F_{\mathrm{P}}(z, \kappa_0(z, \eta)) \quad \text{and} \quad \widehat{G^{\mathrm{CL}}_{\mathrm{P}}}(x) = G_{\mathrm{P}}(x) = G_{\mathrm{P}}(z, \kappa_0(z, \eta)),$$

are OSC, so $\widehat{F^{\mathrm{CL}}_{\mathrm{P}}}$ and $\widehat{G^{\mathrm{CL}}_{\mathrm{P}}}$ are OSC on $\mathcal{X}_1$. Alternatively, $\widehat{F^{\mathrm{CL}}_{\mathrm{P}}}$ and $\widehat{G^{\mathrm{CL}}_{\mathrm{P}}}$ are OSC on $\mathcal{X}_1$ by [4, Lemma 5.16]. Therefore, $\widehat{F^{\mathrm{CL}}_{\mathrm{P}}}$ and $\widehat{G^{\mathrm{CL}}_{\mathrm{P}}}$ are OSC.

Similarly, $\widehat{F^{\mathrm{CL}}_{\mathrm{K}_1}}$ and $\widehat{G^{\mathrm{CL}}_{\mathrm{K}_1}}$ are OSC, again by [4, Lemma 5.16]. □

Proof that $\widehat{F}$ and $\widehat{G}$ are locally bounded. By (B5), F_{P}, F_{K_0}, F_{K_1}, G_{P}, G_{K_0}, G_{K_1}, and κ_1 are locally bounded, whereas κ_0 is locally bounded because it is continuous (B4). Thus, it follows that the closed-loop functions $F_{\mathrm{P}}^{\mathrm{CL}}$, $F_{\mathrm{K}_0}^{\mathrm{CL}}$, $F_{\mathrm{K}_1}^{\mathrm{CL}}$, $G_{\mathrm{P}}^{\mathrm{CL}}$, $G_{\mathrm{K}_0}^{\mathrm{CL}}$, $G_{\mathrm{K}_1}^{\mathrm{CL}}$ are locally bounded. Furthermore, since f_{v} and V_{P} are continuous, $F_{\mathrm{S}}^{\mathrm{CL}}$ and $G_{\mathrm{S}}^{\mathrm{CL}}$ are also locally bounded. The regularized functions in (4.15) are also locally bounded, per [4, Lemma 5.16]. □

Proof that $F(x)$ is convex for all $x \in C$. Take any $x := (z, \eta_0, \eta_1, v, q) \in C$. By assumption (B6), $\widehat{F_{\mathrm{K}_0}^{\mathrm{CL}}}(x) = F_{\mathrm{K}_0}^{\mathrm{CL}}(x) = F_{\mathrm{K}_0}(z, \eta_0)$ is convex. The set $\widehat{F_{\mathrm{P}}^{\mathrm{CL}}}(x)$ is also convex: If $q = 0$, then $F_{\mathrm{P}}^{\mathrm{CL}}(x) = F_{\mathrm{P}}(z, \kappa(x))$, again per (B6), whereas if $q = 1$, then $\widehat{F_{\mathrm{P}}^{\mathrm{CL}}}(x)$ is constructed as the intersection of convex sets and therefore is convex. Similarly, $\widehat{F_{\mathrm{K}_1}^{\mathrm{CL}}}(x)$ is also convex, by construction. The set $\widehat{F_{\mathrm{S}}^{\mathrm{CL}}}(x)$ is a singleton, so it is convex. Therefore, $\widehat{F}(x)$ is convex because it is the Cartesian product of convex sets. □

B.1.2 Proof that V is a Lyapunov function candidate

Lemma B.1. *Suppose that $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, and $\mathcal{H}_{\mathrm{K}_1}$ satisfy Assumption 4.1 and that $\mathcal{A}_{\mathrm{P}}$ and $\mathcal{H}_{\mathrm{P}\times 0}$ satisfy Assumption 4.5 with Lyapunov function V_{P}. Suppose additionally that $\gamma \in (0, 1]$, $\mu > 0$, and $\delta : \mathcal{E}_{\mathrm{P},0} \to \mathbb{R}_{>0}$ is continuous and strictly positive. Then, V is a Lyapunov function candidate (Definition 1.5) with respect to $\mathcal{A}$ for $\mathcal{H}$.*

Proof. From (L2) and (L4), we have that $\mathrm{dom}(V_{\mathrm{P}}) = \mathrm{dom}(\sigma_0) = \mathcal{E}_{\mathrm{P},0}$, so V is well-defined on $\mathrm{dom}\, V = \mathcal{X}$. By Lemmas 4.1–4.2, we have that C is closed and $G(D) \subset C \cup D = \mathcal{X}$. Thus, V satisfies (LFC1):

$$\overline{C} \cup D \cup G(C) = C \cup D = \mathcal{X} = \mathrm{dom}(V).$$

Next, we show V is continuous and there exists an open neighborhood of C where V is locally Lipschitz. The function V is continuous because it is defined as the point-wise maximum of two continuous functions, V_{P} (which is continuous because V_{P} is a Lyapunov function candidate) and $v \mapsto v$.

We want to show that V is locally Lipschitz on an open neighborhood of C. By assumption, there exists an open neighborhood $U_{\mathrm{P},0}$ of C_{K_0} where V_{P} is locally Lipschitz. Consider the open set

$$U := U_{\mathrm{P},0} \times \mathbb{R}^{n_1} \times \mathbb{R} \times \mathbb{R}.$$

Recall that $C_{\mathrm{K}_0}^{\mathrm{CL}} = C_{\mathrm{K}_0} \times \mathcal{E}_1 \times \mathbb{R}_{\geq 0} \times \{0,1\}$ and $C_{\mathrm{K}_0} \subset U_{\mathrm{P},0}$, so U is an open neighborhood of $C_{\mathrm{K}_0}^{\mathrm{CL}}$. Since $C \subset C_{\mathrm{K}_0}^{\mathrm{CL}}$, the set U is an open neighborhood of C. Furthermore, V is locally Lipschitz continuous on U because V_{P} is locally Lipschitz on $U_{\mathrm{P},0}$, the function $v \mapsto v$ is Lipschitz everywhere, and the max operator preserves Lipschitz continuity. Thus, V satisfies (LFC2).

Finally, we show that V is positive definite on $C \cup D \cup G(D)$ with respect to $\mathcal{A}$. As shown above, $C \cup D \cup G(D) = C \cup D = \mathcal{X}$, so we must show that for all $x \in \mathcal{X}$, if $x \in \mathcal{A}$, then $V(x) = 0$, and if $x \notin \mathcal{A}$, then $V(x) > 0$. Take any $x \in \mathcal{A}$. Then, $(z, \eta_0) \in \mathcal{A}_{\mathrm{P}}$ and $v = 0$, so $V_{\mathrm{P}}(z, \eta_0) = 0$ and $V(x) = 0$. Alternatively, take any $x \in \mathcal{X} \setminus \mathcal{A}$. Then, either $(z, \eta_0) \notin \mathcal{A}_{\mathrm{P}}$ or $v \neq 0$. We get that $V_{\mathrm{P}}(z, \eta_0) > 0$ or $v > 0$, respectively, so $V(x) > 0$. Therefore, V satisfies (LFC3). □

B.1.3 Proof of Lemma 4.2

Before proving Lemma 4.2, it is useful to first prove the following lemma.

Lemma B.2. *Suppose $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, and $\mathcal{H}_{\mathrm{S}}$ satisfy Assumption 4.1. Then, $C_\star^{\mathrm{CL}} \cup D_\star^{\mathrm{CL}} = \mathcal{X}$ for each $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$.*

Proof. Take any subsystem $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$. From definitions, $C_\star^{\mathrm{CL}} \cup D_\star^{\mathrm{CL}} \subset \mathcal{X}$. Thus, all that remains is to show $\mathcal{X} \subset C_\star^{\mathrm{CL}} \cup D_\star^{\mathrm{CL}}$. To this end, take any $x := (z, \eta_0, \eta_1, v, q) \in \mathcal{X}$; we will show that $x \in C_\star^{\mathrm{CL}} \cup D_\star^{\mathrm{CL}}$.

Consider the plant subsystem, $\star = \mathrm{P}$. By (B1), we have that $(z, \kappa(x)) \in C_{\mathrm{P}} \cup D_{\mathrm{P}}$. Per (B3), we find that $\mathrm{dom}(F_{\mathrm{P}}) = C_{\mathrm{P}}$ and $\mathrm{dom}(G_{\mathrm{P}}) = D_{\mathrm{P}}$ so $x \in \mathrm{dom}(F_{\mathrm{P}}^{\mathrm{CL}}) \cup \mathrm{dom}(G_{\mathrm{P}}^{\mathrm{CL}})$. Using the definitions of $C_{\mathrm{P}}^{\mathrm{CL}}$, $D_{\mathrm{P}}^{\mathrm{CL}}$, $C_{\mathrm{P}}^{\mathrm{CL}}$, and $D_{\mathrm{P}}^{\mathrm{CL}}$ in (4.8) and (4.15), we find that

$$x \in \mathrm{dom}(F_{\mathrm{P}}^{\mathrm{CL}}) \cup \mathrm{dom}(G_{\mathrm{P}}^{\mathrm{CL}}) = C_{\mathrm{P}}^{\mathrm{CL}} \cup D_{\mathrm{P}}^{\mathrm{CL}}.$$

For $\star = \mathrm{K}_0$, we have

$$x \in \mathcal{X} = (C_{\mathrm{K}_0} \cup D_{\mathrm{K}_0}) \times \mathcal{E}_1 \times \mathcal{V} \times \{0,1\} = C_{\mathrm{K}_0}^{\mathrm{CL}} \cup D_{\mathrm{K}_0}^{\mathrm{CL}}.$$

Similarly, for $\star = \{\mathrm{K}_1, \mathrm{S}\}$, we have that $\mathcal{X} = C_{\mathrm{K}_1} \cup D_{\mathrm{K}_1} = C_{\mathrm{K}_1}^{\mathrm{CL}} \cup D_{\mathrm{K}_1}^{\mathrm{CL}} = C_{\mathrm{K}_1}^{\mathrm{CL}} \cup D_{\mathrm{K}_1}^{\mathrm{CL}}$ and $\mathcal{X} = C_{\mathrm{S}} \cup D_{\mathrm{S}} = C_{\mathrm{S}}^{\mathrm{CL}} \cup D_{\mathrm{S}}^{\mathrm{CL}} = C_{\mathrm{S}}^{\mathrm{CL}} \cup D_{\mathrm{S}}^{\mathrm{CL}}$ by (B1), so

$$x \in C_{\mathrm{K}_1}^{\mathrm{CL}} \cup D_{\mathrm{K}_1}^{\mathrm{CL}} \quad \text{and} \quad x \in C_{\mathrm{S}}^{\mathrm{CL}} \cup D_{\mathrm{S}}^{\mathrm{CL}}.$$ □

We now prove Lemma 4.2.

Proof that $C \cup D = \mathcal{X}$. Take any $x \in \mathcal{X}$. For each $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$, we either have $x \in C_\star^{\mathrm{CL}}$ or $x \in D_\star^{\mathrm{CL}}$. If $x \in C_\star^{\mathrm{CL}}$ for all $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$, then $x \in C$. Otherwise, $x \in D_\star^{\mathrm{CL}}$ for some $\star \in \{\mathrm{P}, \mathrm{K}_0, \mathrm{K}_1, \mathrm{S}\}$, so $x \in D$. Therefore, $C \cup D = \mathcal{X}$. □

Proof that $\widehat{G}(D) \subset C \cup D$. To show $\widehat{G}(D) \subset C \cup D$, we will instead show that $G(D) \subset C \cup D$. The motivation for this is that[1] $\widehat{G}(D) = \overline{G(D)}$, and $C \cup D$ is closed, so $G(D) \subset C \cup D$ implies

$$\widehat{G}(D) = \overline{G(D)} \subset C \cup D.$$

We now prove that $G(D) \subset C \cup D$. Take any

$$x := (z, \eta_0, \eta_1, v, q) \in D \quad \text{and} \quad g := (g_z, g_{\eta_0}, g_{\eta_1}, g'_{\mathsf{v}}, g_q) \in \widehat{G}(x)$$

(the prime notation on g'_{v} is used to distinguish it from the function g_{v}). We want to show $g \in C \cup D = \mathcal{X}$. From the definition of D, the vector x must be in $D_{\mathrm{P}}^{\mathrm{CL}}$, $D_{\mathrm{K}_0}^{\mathrm{CL}}$, $D_{\mathrm{K}_1}^{\mathrm{CL}}$, or $D_{\mathrm{S}}^{\mathrm{CL}}$. We will consider these four cases, which correspond to jumps caused by $\mathcal{H}_{\mathrm{P}}$, $\mathcal{H}_{\mathrm{K}_0}$, $\mathcal{H}_{\mathrm{K}_1}$, or $\mathcal{H}_{\mathrm{S}}$, respectively.

Case 1. Suppose $x \in D_{\mathrm{P}}^{\mathrm{CL}}$ and $g \in \widehat{G_{\mathrm{P}}^{\mathrm{CL}}}(x)$, which indicates a jump in plant state from $(z, \kappa(x)) \in D_{\mathrm{P}}$ to $g_z \in G_{\mathrm{P}}(z, \kappa(x))$, while the other components are unchanged; that is, $g_{\eta_0} = \eta_0$, $g_{\eta_1} = \eta_1$, $g'_{\mathsf{v}} = v$, and $g_q = q$. By Assumption 4.2 and (B1) of Assumption 4.1, $(g_z, \eta_0) \in C_{\mathrm{K}_0} \cup D_{\mathrm{K}_0} = \mathcal{E}_{\mathrm{P},0}$. Since $g_{\eta_1} = \eta_1 \in \mathcal{E}_1$, $g'_{\mathsf{v}} = v \in \mathcal{V}$, and $g_q = q \in \{0, 1\}$ are unchanged, we have that

$$(g_z, g_{\eta_0}, g_{\eta_1}, g'_{\mathsf{v}}, g_q) \in \mathcal{E}_{\mathrm{P},0} \times \mathcal{E}_1 \times \mathcal{V} \times \{0, 1\} = \mathcal{X}.$$

Case 2. Suppose $x \in D_{\mathrm{K}_0}^{\mathrm{CL}}$ and $g \in \widehat{G_{\mathrm{K}_0}^{\mathrm{CL}}}(x)$. We have $(z, \eta_0) \in D_{\mathrm{K}_0}$ and $g_{\eta_0} \in G_{\mathrm{K}_0}(z, \eta_0)$. By assumption Assumption 4.2, we have $(z, g_{\eta_0}) \in \mathcal{E}_{\mathrm{P},0}$ and the other components are unchanged, so, as before, g is in $\mathcal{X}$.

Case 3. Suppose $x \in D_{\mathrm{K}_1}^{\mathrm{CL}}$ and $g \in \widehat{G_{\mathrm{K}_1}^{\mathrm{CL}}}(x)$. By assumption Assumption 4.2, we have $g_{\eta_1} \in \mathcal{E}_1$ with the other components unchanged, so $g \in \mathcal{X}$.

[1] Consider $\mathcal{H}_{\mathrm{P}}$. In the definition of $\widehat{G_{\mathrm{P}}^{\mathrm{CL}}}$, we regularize $G_{\mathrm{P}}^{\mathrm{CL}}$ using $\bigcap_{\delta>0} \overline{G_{\mathrm{P}}^{\mathrm{CL}}(x + \delta\mathbb{B})}$. Since $\mathrm{dom}(G_{\mathrm{P}}^{\mathrm{CL}}) = D_{\mathrm{P}}^{\mathrm{CL}}$, however, $G_{\mathrm{P}}^{\mathrm{CL}}(x + \delta\mathbb{B}) = G_{\mathrm{P}}^{\mathrm{CL}}\big((x + \delta\mathbb{B}) \cap D_{\mathrm{P}}^{\mathrm{CL}}\big) \subset G_{\mathrm{P}}^{\mathrm{CL}}(D_{\mathrm{P}}^{\mathrm{CL}})$ for all $\delta > 0$. Therefore, $\widehat{G_{\mathrm{P}}^{\mathrm{CL}}}(x) \subset \overline{G_{\mathrm{P}}^{\mathrm{CL}}(D_{\mathrm{P}}^{\mathrm{CL}})}$

Case 4. Suppose $x \in D_{\mathrm{S}}^{\mathrm{CL}}$ and $g \in \widehat{G_{\mathrm{S}}^{\mathrm{CL}}}(x)$. We have that $g'_{\mathsf{v}} = g_{\mathsf{v}}(x) \in \mathcal{V}$ per Assumption 4.2, and $q = 1 - q \in \{0, 1\}$ with the other components unchanged, so g is in $\mathcal{X}$.

Therefore, $\widehat{G}(D) \subset \mathcal{X} = C \cup D$. □

B.1.4 Construction of Class $\mathcal{K}_\infty$ bounds on V

Lemma B.3. *Suppose the assumptions of Lemma 4.2 hold, and that there exists $\alpha_p \in \mathcal{K}_\infty$ such that $\alpha_p(|(z, \eta_0)|_{\mathcal{A}_{\mathrm{P}}}) \leq V_{\mathrm{P}}(z, \eta_0)$ for all $(z, \eta_0) \in C_{\mathrm{K}_0} \cup D_{\mathrm{K}_0}$, as per* (L3) *of Assumption 4.5, and let $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be defined by*

$$\alpha(r) := \min\Big\{\alpha_p(r/\sqrt{2}), r/\sqrt{2}\Big\} \qquad \forall r \geq 0. \tag{B.1}$$

Then, α is a class-$\mathcal{K}_\infty$ and $\alpha(|x|_{\mathcal{A}}) \leq V(x)$ for all $x \in C \cup D \cup G(D)$.

Proof. Per Lemma 4.2, $\mathcal{X} = C \cup D \cup G(D)$. Take any $x := (z, \eta_0, \eta_1, v, q) \in \mathcal{X}$. From Lemma B.4,

$$|x|_{\mathcal{A}} = \sqrt{|(z, \eta_0)|^2_{\mathcal{A}_{\mathrm{P}}} + v^2} \leq \sqrt{2} \max\{|(z, \eta_0)|_{\mathcal{A}_{\mathrm{P}}}, v\}.$$

Then,

$$\begin{aligned}\alpha(|x|_{\mathcal{A}}) &= \min\Big\{\alpha_p(|x|_{\mathcal{A}}/\sqrt{2}),\ |x|_{\mathcal{A}}/\sqrt{2}\Big\} \\ &\leq \min\Big\{\alpha_p\Big(\tfrac{\sqrt{2}}{\sqrt{2}} \max\big\{|(z, \eta_0)|_{\mathcal{A}}, v\big\}\Big),\ \tfrac{\sqrt{2}}{\sqrt{2}} \max\big\{|(z, \eta_0)|_{\mathcal{A}}, v\big\}\Big\}. \end{aligned} \tag{B.2}$$

Consider two cases:

Case 1. Suppose $|(z, \eta_0)|_{\mathcal{A}} \geq v$. Then, $\max\big\{|(z, \eta_0)|_{\mathcal{A}}, v\big\} = |(z, \eta_0)|_{\mathcal{A}}$, so

$$\begin{aligned}\alpha(|x|_{\mathcal{A}}) &= \min\{\alpha_p(|(z, \eta_0)|_{\mathcal{A}}),\ |(z, \eta_0)|_{\mathcal{A}}\} \\ &\leq \alpha_p(|(z, \eta_0)|_{\mathcal{A}}) \leq V_{\mathrm{P}}(z, \eta_0) \leq \max\{V_{\mathrm{P}}(z, \eta_0), v\} = V(x). \end{aligned} \tag{B.3}$$

Case 2. Suppose $|(z, \eta_0)|_{\mathcal{A}} < v$. Then, $\max\big\{|(z, \eta_0)|_{\mathcal{A}}, v\big\} = v$, so

$$\alpha(|x|_{\mathcal{A}}) = \min\{\alpha_p(v), v\} \leq v \leq \max\{V_{\mathrm{P}}(z, \eta_0), v\} = V(x). \tag{B.4}$$

Therefore, by cases, we have that $\alpha(|x|_{\mathcal{A}}) \leq V(x)$ for all $x \in \mathcal{X}$. Finally, α is a class-$\mathcal{K}_\infty$ function because it is defined as the pointwise minimum of two $\mathcal{K}_\infty$ functions, so it is class-$\mathcal{K}_\infty$. □

Lemma B.4. *For any $a \geq 0$ and $b \geq 0$,*

$$\max\{a,\ b\} \leq \sqrt{a^2 + b^2} \leq \sqrt{2}\max\{a,\ b\}.$$

Proof. Take any $a \geq 0$ and $b \geq 0$. We have that $a = \sqrt{a^2} \leq \sqrt{a^2 + b^2}$ and $b = \sqrt{b^2} \leq \sqrt{a^2 + b^2}$, so

$$\max\{a,\ b\} \leq \sqrt{a^2 + b^2}.$$

Since $\max\{\cdot, \cdot\}$ and $\sqrt{(\cdot)^2 + (\cdot)^2}$ are symmetrical, we can assume without loss of generality that $a \geq b$. Then, $\max\{a,\ b\} = a$ and

$$\sqrt{a^2 + b^2} \leq \sqrt{a^2 + a^2} = \sqrt{2}\, a = \sqrt{2}\max\{a,\ b\}. \qquad \square$$

B.1.5 Tangent Cone Results

This section contains lemmas regarding tangent cones.

Lemma B.5. *Given sets $S \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^n$, suppose $\mathcal{U}$ is open. For every point $x \in S \cap \mathcal{U}$,*

$$T_S(x) = T_{S \cap \mathcal{U}}(x).$$

Proof. Take any $x \in S \cap \mathcal{U}$. We immediately have that $T_{S\cap\mathcal{U}}(x) \subset T_S(x)$, since $S \cap \mathcal{U} \subset S$. To show equality, we want to show $T_S(x) \subset T_{S\cap U}$. Take any $w \in T_S(x)$. By definition, there exists convergent sequences $w_i \to w$ in $\mathbb{R}^n$ and $h_i \to 0^+$ in $\mathbb{R}_{\geq 0}$ such that $x + h_i w_i \in S$ for all $i \in \mathbb{N}$. Since $\mathcal{U}$ is open and contains x, there exists $\varepsilon > 0$ such that $x + \varepsilon\mathbb{B} \subset \mathcal{U}$. The sequence $i \mapsto w_i$ is bounded, so $h_i w_i \to 0$. Thus, there exists $i_0 \in \mathbb{N}$ such that $|h_i w_i| \leq \varepsilon$ all $i \geq i_0$. That is, the tail of $i \mapsto h_i w_i$ is contained in $x + \varepsilon\mathbb{B} \subset \mathcal{U}$, so the sequences $h_{i-i_0} \to 0^+$ and $w_{i-i_0} \to w$ satisfy $x + h_{i-i_0} w_{i-i_0} \in S \cap U$ for all $i \in \mathbb{N}$. Therefore, $w \in T_{S\cap\mathcal{U}}(x)$. $\square$

Lemma B.6. *Consider a closed set $S \in \mathbb{R}^n$ and a finite set $Q \subset \mathbb{R}$. For $x := (s, q) \in S \times Q$, the tangent cone to $S \times Q$ at x is*

$$T_{S\times Q}(x) = T_S(s) \times \{0\}.$$

Proof. Take any $x := (s, q) \in S \times Q$. We want to show $T_{S\times Q}(x) \subset T_S(s) \times \{0\}$ and $T_{S\times Q}(x) \supset T_S(s) \times \{0\}$.

("$\subset$") Take any $(w, v) \in T_{S\times Q}(x, q)$. We want to show $(w, v) \in T_S(s) \times \{0\}$. By definition of the tangent cone, there exists $(w_i, v_i) \to (w, v)$ and $h_i \to 0^+$ such that $x + h_i(w_i, v_i) \in S \times Q$ for all $i \in \mathbb{N}$, which is to say, $s + h_i w_i \in S$ and $q + h_i v_i \in Q$. Thus, $w \in T_S(s)$. Since Q is finite, however, each point in Q is isolated, meaning there is an open neighborhood of q such that $\mathcal{U} \cap Q = \{q\}$. Eventually, $v_i = 0$ for all i larger than some $i_0 \in \mathbb{N}$, so $v_i \to 0$. Thus, $v = 0$, concluding the proof that $(w, v) \in T_S(s) \times \{0\}$.

("$\supset$") Take any $(w, v) \in T_S(s) \times \{0\}$, which is to say, $w \in T_S(s)$ and $v = 0$. There exists a sequence $w_i \to w$ and $h_i \to 0^+$ such that $s + h_i w_i \in S$. Let $i \mapsto v_i := 0$ for each $i \in \mathbb{N}$. We have that $(s, q) + h_i(w_i, v_i) = (s + h_i w_i, q) \in S \times F$ for all $i \in \mathbb{N}$. Therefore, $(w, 0) \in T_{S\times F}(x)$. □

B.2 Additional Results for Relaxed Lyapunov Conditions

B.2.1 Additional Results for Hybrid Time Domains

This section contains results for relaxing the assumptions imposed on the hybrid time domains of solutions in parts (b)-(f) of [3, Thm. 3.19(3)]

Lemma B.7. *Let $\mathcal{H}$ be a hybrid system. The following are equivalent:*

(L7.1) *For each $r > 0$, there exist $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ such that for every solution φ to $\mathcal{H}$ and every $T \geq 0$,*

$$|\varphi(0,0)|_{\mathcal{A}} \in (0, r],\ (t, j) \in \operatorname{dom}\varphi,\ t + j \geq T \quad \implies \quad t \geq \gamma_r(T) - N_r.$$

(L7.2) *For each $r > 0$, there exist $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ such that for every solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$,*

$$(t, j) \in \operatorname{dom}\varphi \quad \implies \quad t \geq \gamma_r(t + j) - N_r$$

Proof. First, we show (L7.1) $\implies$ (L7.2). Take any $r > 0$ and take $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ from (L7.1). Take any solution φ to $\mathcal{H}$ such that $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$ and any $(t_0, j_0) \in \operatorname{dom}\varphi$. Let $T_0 := t_0 + j_0$. By (L7.1),

$$t_0 \geq \gamma_r(T_0) - N_r = \gamma_r(t_0 + j_0) - N_r.$$

Therefore, (L7.2) holds.

Conversely, we show (L7.1) $\Longleftarrow$ (L7.2). Suppose (L7.2). Take any $r > 0$ and take $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ from (L7.2). Take any solution φ to $\mathcal{H}$ such that $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$, any $T_1 \geq 0$, and any $(t_1, j_1) \in \operatorname{dom} \varphi$ such that $t_1 + j_1 \geq T_1$. By (L7.2) and the monotonicity of γ_r,

$$t_1 \geq \gamma_r(t_1 + j_1) - N_r \geq \gamma_r(T_1) - N_r.$$

Therefore, (L7.1) holds. □

The following lemma and proof is nearly identical to the preceding one. The only difference is "t" is replaced by "j" in the left-hand side of the "$\star \geq \gamma_r(\star) - N_r$" inequalities.

Lemma B.8. *Let $\mathcal{H}$ be a hybrid system. The following are equivalent:*

(L8.1) *For each $r > 0$, there exist $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ such that for every solution φ to $\mathcal{H}$ and every $T \geq 0$,*

$$|\varphi(0,0)|_{\mathcal{A}} \in (0, r],\ (t, j) \in \operatorname{dom} \varphi,\ t + j \geq T \quad \Longrightarrow \quad j \geq \gamma_r(T) - N_r.$$

(L8.2) *For each $r > 0$, there exist $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ such that for every solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$,*

$$(t, j) \in \operatorname{dom} \varphi \quad \Longrightarrow \quad j \geq \gamma_r(t + j) - N_r$$

Proof. First, we show (L8.1) $\Longrightarrow$ (L8.2). Suppose (L8.1). Take any $r > 0$ and take $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ from (L8.1). Take any solution φ to $\mathcal{H}$ such that $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$ and any $(t_0, j_0) \in \operatorname{dom} \varphi$. Let $T_0 := t_0 + j_0$. By (L8.1),

$$j_0 \geq \gamma_r(T_0) - N_r = \gamma_r(t_0 + j_0) - N_r.$$

Therefore, (L7.2) holds.

Conversely, we show (L8.1) $\Longleftarrow$ (L8.2). Suppose (L8.2). Take any $r > 0$ and take $\gamma_r \in \mathcal{K}_\infty$ and $N_r \geq 0$ from (L8.2). Take any solution φ to $\mathcal{H}$ such that $|\varphi(0,0)|_{\mathcal{A}} \in (0, r]$, any $T_1 \geq 0$, and any $(t_1, j_1) \in \operatorname{dom} \varphi$ such that $t_1 + j_1 \geq T_1$. By (L8.2) and the monotonicity of γ_r,

$$j_1 \geq \gamma_r(t_1 + j_1) - N_r \geq \gamma_r(T_1) - N_r.$$

Therefore, (L8.1) holds. □

The next result simplifies the condition for the "finite number of jumps" case in [3, Thm. 3.19]

Lemma B.9. *Let $\mathcal{H}$ be a hybrid system. The following are equivalent:*

1. *For every $r \geq 0$, there exists $J_r \in \mathbb{N}$ such that for every solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \leq r$,*
$$\operatorname{dom} \varphi \subset \mathbb{R} \times \{0, 1, \ldots, J_r\}.$$
2. *There exist $\gamma \in \mathcal{K}$ and $J > 0$ such that for every solution φ to $\mathcal{H}$ and every $(t, j) \in \operatorname{dom} \varphi$,*
$$j \leq \gamma(|\varphi(0,0)|_{\mathcal{A}}) + J.$$

Proof. (*1.* $\implies$ *2.*) Suppose that for every $r \geq 0$, there exists $J_r \in \mathbb{N}$ such that for every solution φ to $\mathcal{H}$ such that $|\varphi(0,0)|_{\mathcal{A}} \leq r$,

$$\operatorname{dom} \varphi \subset \mathbb{R} \times \{0, 1, \ldots, J_r\}.$$

Let $\tilde{\jmath} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a nondecreasing function chosen such that for each $r \geq 0$ and $\operatorname{dom} \varphi \subset \mathbb{R} \times \{0, 1, \ldots, \tilde{\jmath}(r)\}$ for each solution φ to $\mathcal{H}$ with $|\varphi(0,0)|_{\mathcal{A}} \leq r$.

Because $\tilde{\jmath}(r)$ exists for all $r \geq 0$ and is nondecreasing, it is locally bounded, so the supremum over any compact interval is finite. We will define a continuous function $\sigma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that upper bounds $\tilde{\jmath}$ by first defining σ at 0, 1, 2, etc., and then using linear interpolation between those points. For $r_0 = 0$, let

$$\sigma(r_0) := \sup_{r \in [0,1]} \tilde{\jmath}(r).$$

For each $r_i \in \{1, 2, \ldots\}$, let

$$\sigma(r_i) := \max\left\{ \sup_{r \in [r_i, r_i + 1]} \tilde{\jmath}(r),\ \sigma(r_{i-1}) + 1 \right\}.$$

Since $\tilde{\jmath}(r_{i+1}) \geq \tilde{\jmath}(r_i) + 1$, the sequence $r_i \mapsto \sigma(r_i)$ is strictly increasing. Then, for $r \notin \mathbb{N}$, let

$$\sigma(r) = (r - \lfloor r \rfloor)\sigma(\lfloor r \rfloor) + (\lceil r \rceil - r)\sigma(\lceil r \rceil),$$

which is the linear interpolation between $\sigma(\lfloor r \rfloor)$ and $\sigma(\lceil r \rceil)$ and, as such, σ is continuous. For all $r \geq 0$, we have that $\sigma(r) \geq \tilde{\jmath}(r)$. Let $J := \sigma(0)$ and let $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be defined, for each $r \geq 0$, by

$$\gamma(r) := \sigma(r) - J.$$

Because σ is continuous and increasing, γ is also. Additionally, $\gamma(0) = \sigma(0) - J = 0$. Therefore, $\gamma \in \mathcal{K}$.

Take any solution φ to $\mathcal{H}$. Because

$$\tilde{j}(|\varphi(0,0)|_{\mathcal{A}}) \leq \sigma(|\varphi(0,0)|_{\mathcal{A}}) = \gamma(|\varphi(0,0)|_{\mathcal{A}}) + J,$$

we have that

$$\operatorname{dom}\varphi \subset \mathbb{R} \times \{0,\ 1,\ \ldots,\ \tilde{j}(|\varphi(0,0)|_{\mathcal{A}})\} \subset \mathbb{R} \times \{0,\ 1,\ \ldots,\ \gamma(|\varphi(0,0)|_{\mathcal{A}}) + J\}.$$

Thus, for all $(t,j) \in \operatorname{dom}\varphi$,

$$j \leq \gamma(|\varphi(0,0)|_{\mathcal{A}}) + J.$$

(*1.* $\Longleftarrow$ *2.*) Suppose that there exist $\gamma \in \mathcal{K}$ and $J > 0$ such that for every solution φ to $\mathcal{H}$ and every $(t,j) \in \operatorname{dom}\varphi$,

$$j \leq \gamma(|\varphi(0,0)|_{\mathcal{A}}) + J.$$

Take any $r \geq 0$ and let $J_r := \gamma(r) + J$. Then, for every solution φ to $\mathcal{H}$ with $|\varphi|_{\mathcal{A}}$, we have that

$$\gamma(|\varphi(0,0)|_{\mathcal{A}}) + J \leq \gamma(r) + J = J_r.$$

Therefore, for all $(t,j) \in \operatorname{dom}\varphi$,

$$j \leq J_r. \qquad \square$$

B.3 Additional Results for CTG

This section contains results omitted from Chapter 6.

Lemma B.10. *Let $\mathcal{H}$ be a conical hybrid system with constant flows and let φ be any solution to $\mathcal{H}$. For all $r > 0$, the hybrid arc $\psi(t,j) := r\varphi(t/r, j)$ for all $(t,j) \in \operatorname{dom}(\psi) := \{(t,j) \mid (t/r, j) \in \operatorname{dom}(\varphi)\}$ is a solution to $\mathcal{H}$.*

Proof. Let $\mathcal{H} = (C, f, D, g)$ be a conical hybrid system with constant flows, let φ be any solution to $\mathcal{H}$, and for any $r > 0$, let $\psi(t,j) := r\varphi(t/r, j)$ for all $(t,j) \in \operatorname{dom}(\psi) := \{(t,j) \mid (t/r, j) \in \operatorname{dom}(\varphi)\}$.

We have that t_j is a jump time in $\mathrm{dom}(\psi)$ if and only if t_j/r is a jump time in $\mathrm{dom}(\varphi)$, so $\varphi(t_j/r, j-1) \in D$. Since D is a cone, $\psi(t_j, j-1) = r\varphi(t_j/r, j-1)$ is also in D. By the linearity of g,

$$\begin{aligned} g(\psi(t_j, j-1)) &= g(r\varphi(t_j/r, j-1)) \\ &= rg(\varphi(t_j/r, j-1)) \\ &= r\varphi(t_j/r, j) \\ &= \psi(t_j, j), \end{aligned}$$

so ψ satisfies the jump conditions.

Take any pair of consecutive jump times t_j and t_{j+1} in $\mathrm{dom}(\psi)$ such that $t_{j+1} > t_j$, meaning $I := [t_j, t_{j+1}]$ is an interval of flow in $\mathrm{dom}(\psi)$. Then, $[t_j/r, t_{j+1}/r]$ is also an interval of flow in $\mathrm{dom}(\varphi)$. For each $t \in I$, we have that $\psi(t, j) = r\varphi(t/r, j) \in C$ because C is a closed cone. From flow condition (1.6) in the definition of hybrid solutions, we have that $\dot{\varphi}(t, j) = f(\varphi(t, j))$ for almost all $t \in I$. Thus, using the chain rule and the fact that f is constant-valued, we find that

$$\begin{aligned} \dot{\psi}(t, j) &= \frac{d}{dt}(r\dot{\varphi}(t/r, j)) \\ &= r\left(\left.\frac{d}{dt}\right|_{t=t/r} \varphi(t, j)\right)\left(\left.\frac{d}{dt}\right|_{t} (t/r)\right) \\ &= rf(\varphi(t/r, j))(1/r) \\ &= f(\psi(t, j)) \end{aligned}$$

for almost all $t \in I$, so ψ satisfies the flow conditions in the definition of a hybrid solution. Therefore, ψ is a solution to $\mathcal{H}$. □

Lemma B.11. *Let $\mathcal{H}$ be a conical hybrid system with linear flows and let φ be any solution to $\mathcal{H}$. For all $r > 0$, the hybrid arc ψ defined by $\psi(t, j) := r\varphi(t, j)$ for all $(t, j) \in \mathrm{dom}(\psi) := \mathrm{dom}(\varphi)$ is a solution to $\mathcal{H}$.*

Proof. Let $\mathcal{H} = (C, f, D, g)$ be a conical hybrid system with linear flows, let φ be any solution to $\mathcal{H}$, and for any $r > 0$, let $\psi(t, j) := r\varphi(t, j)$ for all $(t, j) \in \mathrm{dom}(\psi) := \mathrm{dom}(\varphi)$.

For each jump time t_j in $\mathrm{dom}(\varphi)$, we have that t_j is a jump time in $\mathrm{dom}(\psi)$ and $\varphi(t_j, j-1) \in D$. Since D is a cone, $\psi(t_j, j-1) = r\varphi(t_j, j-1)$ is also in D. By

the linearity of g,

$$g(\psi(t_j, j-1)) = g(r\varphi(t_j, j-1)) = rg(\varphi(t_j, j-1)) = r\varphi(t_j, j) = \psi(t_j, j),$$

so ψ satisfies the jump conditions.

Take any pair of consecutive jump times t_j and t_{j+1} in $\mathrm{dom}(\varphi)$ such that $t_{j+1} > t_j$, meaning $I := [t_j, t_{j+1}]$ is an interval of flow in $\mathrm{dom}(\varphi)$, and also in $\mathrm{dom}(\psi)$. For each $t \in I$, we have that $\psi(t, j) = r\varphi(t, j) \in C$ because C is a closed cone. Let A be the linear map defining the flow dynamics $\dot{x} = f(x) = Ax$. From flow condition (1.6) in the definition of hybrid solutions, we have that $\dot{\varphi}(t, j) = A\varphi(t, j) = f(\varphi(t, j))$ for almost all $t \in I$. Thus,

$$\dot{\psi}(t, j) = r\dot{\varphi}(t, j) = rA\varphi(t, j) = A(r\varphi(t, j)) = A\psi(t, j)$$

for almost all $t \in I$, so ψ satisfies the flow conditions in the definition of a hybrid solution. Therefore, ψ is a solution to $\mathcal{H}$. □