A "Throw-and-Catch" Hybrid Control Strategy for Robust Global Stabilization of Nonlinear Systems*

Ricardo G. Sanfelice and Andrew R. Teel†

***Abstract*— We present a control strategy that combines local state feedback laws and open-loop schedules to robustly globally asymptotically stabilize a compact subset (typically a point) of the state space for a nonlinear system. The control algorithm is illustrated on the problem of global stabilization of the upright position of the pendubot and implemented in a hybrid controller containing logic variables and logic rules with hysteresis. We also present the design procedure of the hybrid controller for general nonlinear systems. Recent results in the literature on robustness of asymptotic stability in hybrid systems are used in establishing that the closed-loop system is robust to measurement noise and other external disturbances.**

I. Introduction

In this paper, we develop a novel hybrid feedback control strategy for the problem of globally asymptotically stabilizing a point (or a set). Our control strategy combines local feedback stabilizers and open-loop control signals (or schedules) to steer the trajectories toward the desired point from other particular points in the state space, and a "bootstrap" feedback controller that is capable of steering the trajectories to a neighborhood of one of these points from which the local feedback stabilizers and the open-loop controls can be used. A switching logic between these control laws with hysteresis is implemented in a hybrid controller with logic variables and logic rules. We follow the formalism for hybrid systems used in [4], [5] where some of the first general results on robustness of hybrid control systems were obtained.

We will use the problem of global stabilization of the upright position for the pendubot to explain the control strategy and clarify the assumptions for the general case. For the purposes of the discussion, call the upright equilibrium point $\mathcal{A}_u$, the straight-down equilibrium point $\mathcal{A}_r$, and the two other equilibriums, corresponding to the first link up and the second link down, and vice versa, A_{ur} and A_{ru}, respectively. By linearizing the system at the points $\mathcal{A}_u$ and $\mathcal{A}_r$, we construct local stabilizers for neighborhoods of the points $\mathcal{A}_u$ and $\mathcal{A}_r$, respectively. We also construct an open-loop control signal to take the state to a neighborhood of the point $\mathcal{A}_u$ from a neighborhood of the point $\mathcal{A}_r$, and two different open-loop controls to take the state to a neighborhood of the point $\mathcal{A}_r$ from neighborhoods of the points A_{ur} and A_{ru}. These control laws can be constructed by solving a two-point boundary value problem, using trial and error, or relying on "human inspired" control signals as in [3]. Finally, we construct a feedback controller that steers the state to the union of $\mathcal{A}_u$, $\mathcal{A}_r$, A_{ur}, and A_{ru}. Since the system will have some natural damping, the zero control would suffice. Alternatively, additional damping can be added through feedback. In this work, we will show how these ingredients can be used to build a robust, global hybrid feedback stabilizer. To the best of our knowledge, this constitutes the first robust global feedback stabilizer for the pendubot; cf. [2], [1], [6]. Moreover, the proposed control strategy is applicable to general multi-link pendulums.

In a sense, our work can be thought of as a generalization of the work in [7] where a local controller is assumed to be known for the desired equilibrium point and, in addition, the "bootstrap" feedback controller steers the system to near this point. When the bootstrap controller has this especially strong property, no additional open-loop controls and local stabilizers are needed. However, this type of assumption is not reasonable for the pendubot. Indeed, topological considerations easily reveal the impossibility of building a robust feedback stabilizer to take the pendubot from every initial condition to a neighborhood of the straight-up position. This obstruction motivates the relaxation considered herein.

II. Control approach: Robust "throw" and "catch"

A. Robust Global Stabilization of the Pendubot

Consider the dynamical system given in Figure 1 consisting of a pendulum with two links, the *pendubot*. We are interested on the problem of stabilizing both links of the pendubot to the upright position using only torque actuation in the first link. Several control strategies to accomplish this task have appeared in the literature; these include energy

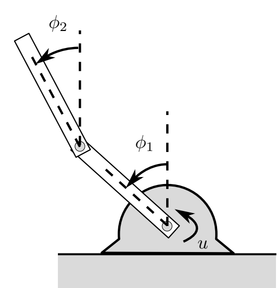

Fig. 1. *The pendubot system: a two-link pendulum with torque actuation u in the first link.*

*Research partially supported by the Army Research Office under Grant no. DAAD19-03-1-0144, the National Science Foundation under Grant no. CCR-0311084 and Grant no. ECS-0622253, and by the Air Force Office of Scientific Research under Grant no. F9550-06-1-0134.

†{rsanfelice,teel}@ece.ucsb.edu, Center for Control Engineering and Computation, Electrical and Computer Engineering Department, University of California, Santa Barbara, CA 93106-9560.

pumping [2], trajectory tracking [6], and jerk control [1], to just list a few. Our goal is to design a control algorithm that accomplishes the stabilization task *globally* (by this, we mean for every initial condition of the pendubot), and *robustly* with respect to measurement noise and external disturbances.

Let ϕ_1 and ϕ_2 denote the angles relative to the upright position, ω_1 and ω_2 the angular velocities, and $u \in \mathbb{R}$ the control input. The dynamical model of this system can be obtained with the Lagrange method. The resulting equations are of the form

$$\begin{array}{rcl} \dot{\phi}_1 & = & \omega_1, \;\; \dot{\omega}_1 = f_1(x,u) \\ \dot{\phi}_2 & = & \omega_2, \;\; \dot{\omega}_2 = f_2(x,u) \;, \end{array} \tag{1}$$

where $x := [\phi_1 \;\; \omega_1 \;\; \phi_2 \;\; \omega_2]^T \in \mathbb{R}^4$ and $f_1, f_2 : \mathbb{R}^4 \times \mathbb{R} \to \mathbb{R}$ are nonlinear, locally Lipschitz functions that define the dynamics of the pendulum. Let $f(x,u) := [\omega_1 \;\; f_1(x,u) \;\; \omega_2 \;\; f_2(x,u)]^T$. We consider that ϕ_1 and ϕ_2 are given by the angle of a vector in the unit circle $Z := \left\{ z \in \mathbb{R}^2 \mid \|z\|_2 = 1 \right\}$. More precisely, for each $i = 1, 2$, ϕ_i is given by the angle of the vector $z_i \in Z$. Note that, with this embedding technique, the problem of globally stabilizing the pendubot to the upright position is equivalent to globally stabilizing the system to the compact set defined by $z_1 = z_2 = [1 \;\; 0]^T$, $\omega_1 = \omega_2 = 0$.

The pendubot system has four equilibrium points:

- Resting ($\mathcal{A}_r$): $\phi_1 = -\pi$, $\omega_1 = 0$, $\phi_2 = -\pi$, $\omega_2 = 0$;
- Upright ($\mathcal{A}_u$): $\phi_1 = \omega_1 = \phi_2 = \omega_2 = 0$;
- Upright/Resting ($\mathcal{A}_{ur}$): $\phi_1 = \omega_1 = 0$, $\phi_2 = -\pi$, $\omega_2 = 0$;
- Resting/Upright ($\mathcal{A}_{ru}$): $\phi_1 = -\pi$, $\omega_1 = \phi_2 = \omega_2 = 0$.

These equilibrium points are depicted in Figure 2.

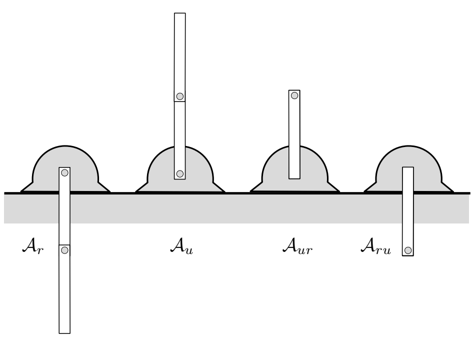

Fig. 2. *Equilibrium configurations of the pendubot.*

B. Control Strategy

We build our feedback control strategy upon the following state-feedback and open-loop control laws.

1) Local state-feedback stabilizers κ_u and κ_r: The construction of local state-feedback stabilizers κ_u for the upright equilibrium $\mathcal{A}_u$ and κ_r for the resting equilibrium $\mathcal{A}_r$ are designed to steer points nearby $\mathcal{A}_u$ and $\mathcal{A}_r$ to the equilibrium point itself, respectively. Such controllers can be designed by linearization and pole placement. For example, for κ_u, let $A := \partial f(x,u)/\partial x|_{x=\mathcal{A}_u, u=0}$ and $B := \partial f(x,u)/\partial u|_{x=\mathcal{A}_u, u=0}$, choose $K \in \mathbb{R}^4$ and $P \in \mathbb{R}^{4\times 4}$, $P = P^T > 0$, such that

$$(A - BK^T)^T P + P(A - BK^T) < 0 \;, \tag{2}$$

and let $\kappa_u(x) := K^T x$. (Such K and P exist as (A, B) is controllable.) The basin of attraction of this controller can be estimated with a sublevel set $L_{V_u}(r_u)$ of the Lyapunov function $V_u(x) := x^T P x$.

2) Open-loop control laws for steering from/to neighborhoods of points to $\mathcal{A}_r, \mathcal{A}_u, \mathcal{A}_{ur}$, and $\mathcal{A}_{ru}$: Construct open-loop controllers $\alpha_{r\to u}$, $\alpha_{ur\to r}$, and $\alpha_{ru\to r}$ such that

a) $\alpha_{r\to u}(t)$ steers the trajectories of (1) from points nearby the *resting* equilibrium $\mathcal{A}_r$ to points nearby the *upright* equilibrium $\mathcal{A}_u$;

b) $\alpha_{ur\to r}(t)$ steers the trajectories of (1) from points nearby the *upright/resting* equilibrium $\mathcal{A}_{ur}$ to points nearby the *resting* equilibrium $\mathcal{A}_r$;

c) $\alpha_{ru\to r}(t)$ steers the trajectories of (1) from points nearby the *resting/upright* equilibrium $\mathcal{A}_{ru}$ to points nearby the *resting* equilibrium $\mathcal{A}_r$.

For example, for item a), we construct a piecewise-continuous function of time $\alpha_{r\to u} : \mathbb{R}_{\geq 0} \to \mathbb{R}$ such that for the initial condition $x^0 = \mathcal{A}_r, t^0 = 0$, the solution to $\dot{x} = f(x, \alpha_{r\to u}(t))$ is in a small neighborhood of $\mathcal{A}_u$. Then, by continuity with respect of initial conditions to (1), there exists a neighborhood S of $\mathcal{A}_r$ and a neighborhood E of $\mathcal{A}_u$ such that solutions to $\dot{x} = f(x, \alpha_{r\to u}(t))$ starting from S reach E in finite time $\tau^*_{r\to u} > 0$. We design $\alpha_{ur\to r}$ and $\alpha_{ru\to r}$ similarly. One technique that can be used to design these open-loop controllers is to define a parameterized basis function for the control law and then determine its parameters by trial and error. A different approach is to solve a two-point boundary value problem (or some other constrained optimal control problem) with boundary constraints corresponding to neighborhoods of $\mathcal{A}_r, \mathcal{A}_u, \mathcal{A}_{ur}$, and $\mathcal{A}_{ru}$.

3) Bootstrap stabilizer κ_0: The main task of this controller is to steer trajectories starting from every point not in $\mathcal{A}_r \cup \mathcal{A}_u \cup \mathcal{A}_{ur} \cup \mathcal{A}_{ru}$ to an small enough neighborhood of $\mathcal{A}_r \cup \mathcal{A}_u \cup \mathcal{A}_{ur} \cup \mathcal{A}_{ru}$. One such a controller is $\kappa_0 \equiv 0$ as the natural damping present in the system steer the trajectories to $\mathcal{A}_r \cup \mathcal{A}_u \cup \mathcal{A}_{ur} \cup \mathcal{A}_{ru}$ with zero control input. In the next section, to obtain better performance, we use a more sophisticated control law which removes energy from the system much faster.

With the control ingredients designed in 1), 2), and 3), the basic tasks that our control strategy performs are:

- For points nearby $\mathcal{A}_r$, apply the state-feedback law κ_r to steer the state to the set S corresponding to $\alpha_{r\to u}$ and then apply $\alpha_{r\to u}$ to steer the trajectories to a neighborhood of $\mathcal{A}_u$;
- For points nearby $\mathcal{A}_u$, apply the state feedback law κ_u to stabilize the trajectories to $\mathcal{A}_u$;
- For points nearby $\mathcal{A}_{ur}$ and $\mathcal{A}_{ru}$, apply the open-loop control laws $\alpha_{ur\to r}$ and $\alpha_{ru\to r}$, respectively, to steer the trajectories to a neighborhood of $\mathcal{A}_r$;

- For any other point in $\mathbb{R}^4$, apply the law κ_0 to steer the trajectories to a neighborhood of $\mathcal{A}_r \cup \mathcal{A}_u \cup \mathcal{A}_{ur} \cup \mathcal{A}_{ru}$.

In Figure 3, we show the combination of these tasks to accomplish global stabilization to the point $\mathcal{A}_u$ of the pendubot. When the open-loop control laws are applied, we say that there is a "throw" between neighborhoods of the equilibrium points, and when the feedback stabilizers are applied, we say that there is a "catch" to one of the equilibrium points.

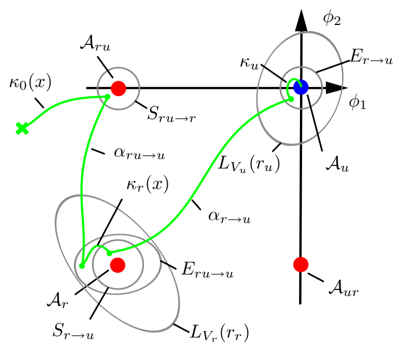

Fig. 3. *Control strategy for robust global stabilization of the pendubot to the point $\mathcal{A}_u$. A sample trajectory in the ϕ_1, ϕ_2 plane resulting from our control strategy is depicted. From the initial point $\times$, the trajectory is steered to the neighborhood $S_{ru\to r}$ of $\mathcal{A}_{ru}$ with $\kappa_0(x)$, from which it is "thrown" to the neighborhood $E_{ru\to r}$ of $\mathcal{A}_r$ with the control law $\alpha_{ru\to r}$. The local stabilizer κ_r "catches" the state to a point in $S_{r\to u}$ from where the open-loop law $\alpha_{r\to u}$ is applied. Finally, after the "throw", the state reaches a point in $E_{r\to u}$ and the last "catch" by the local stabilizer κ_u steers the trajectory to $\mathcal{A}_u$.*

C. Hybrid Controller

The control strategy outlined in Section II-B is implemented in a hybrid controller with logic variables and logic rules with hysteresis. In contrast to the discontinuous control law case, our implementation as a hybrid system guarantees robustness properties of the closed-loop system.

First note that our control strategy can be interpreted as a *directed tree* or *graph* with nodes given by the equilibrium points. The directed tree consists of two paths given by $\mathcal{A}_{ur} \to \mathcal{A}_r \to \mathcal{A}_u$ and $\mathcal{A}_{ru} \to \mathcal{A}_r \to \mathcal{A}_u$.

We number the nodes in each of the paths, starting from $\mathcal{A}_{ur}$ and $\mathcal{A}_{ru}$ and finishing at $\mathcal{A}_u$, by the pairs $(i, j) \in \{1, 2, 3\} \times \{1, 2\}$ where i indicates the node number and j the path number. Then, the two paths are

Path 1: $(1,1) \to (2,1) \to (3,1)$ (i.e. $\mathcal{A}_{ur} \to \mathcal{A}_r \to \mathcal{A}_u$).
Path 2: $(1,2) \to (2,2) \to (3,2)$ (i.e. $\mathcal{A}_{ru} \to \mathcal{A}_r \to \mathcal{A}_u$).

The controller has two logic states, q and p, $(q, p) \in \{-3, -2, 0, 1, 2\} \times \{1, 2\}$, and a timer state $\tau \in \mathbb{R}$. The state q indicates the mode of the controller, p indicates the current path of the trajectories, and τ keeps track of the time that the system has been in open loop. Let $\tau^*_{1,1} = \tau^*_{ur\to r}$, $\tau^*_{1,2} = \tau^*_{ru\to r}$, and $\tau^*_{2,1} = \tau^*_{2,2} = \tau^*_{r\to u}$. The control logic is as follows:

- *Catch mode* when $q < 0$. This mode indicates that the state x is steered to the $(|q|, p)$-th node of the current path p. If $q = -2$ then the control law applied is κ_r, while if $q = -3$ then the control law applied is κ_u.
- *Throw mode* when $q > 0$. This mode indicates that the trajectories are being steered from a neighborhood of the (q, p)-th node to a neighborhood of the $(q+1, p)$-th node of the current path p. If $q = 1, p = 1$, then the control law applied is $\alpha_{ur\to r}$; if $q = 1, p = 2$, then $\alpha_{ru\to r}$ is applied; and if $q = 2$ then $\alpha_{r\to u}$ is applied.
- *Recovery mode* when $q = p = 0$. This mode indicates that the trajectories are being steered to the tree with the control law κ_0.

The hybrid controller updates its state under the following events:

(C) **"Throw-to-catch" transitions**: when the state x is in some neighborhood E of an open-loop control law and in *throw mode* ($q > 0$), the controller jumps to *catch mode* (q updated to $-(|q| + 1)$). The timer state τ is reset to zero.

(T) **"Catch-to-throw" transitions**: when the state x is in some neighborhood S of an open-loop control law and in *catch mode* ($q < 0$), the controller jumps to *throw mode* (q updated to $|q|$). The timer state τ is reset to zero.

(R) **"Throw- or catch- to-recovery" transitions**: when the trajectories
  - while in *throw mode*, do not reach a neighborhood of the associated set E in the expected amount of time (that is, $q > 0$ and $\tau \geq \tau^*_{q,p}$); or
  - while in *catch mode*, leave the basin of attraction of the current local stabilizer;

  then the controller jumps to *recovery mode* (q, p updated to $q = p = 0$).

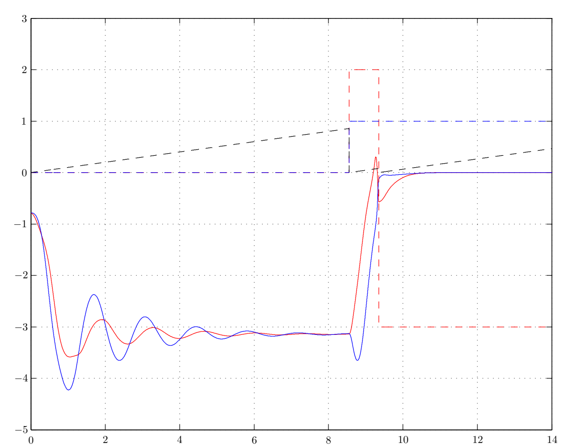

Fig. 4. *Simulation of the pendubot system with our hybrid control strategy. Initial conditions: $x^0 = [-\pi/4, 0, -\pi/4, 0]^T$, $q^0 = p^0 = 1$, $\tau^0 = 0$. The figure depicts: pendubot angles ϕ_1 (red) and ϕ_2 (blue), logic state q (dashed blue), logic state p (dashed red), and timer state τ (dashed black). After an initial switch to recovery mode, when x reaches a neighborhood of $[-\pi, 0, -\pi, 0]^T$, a "throw" is performed (at around 8.5sec.) from the resting configuration (node $(2,1)$, $\mathcal{A}_r$) to a neighborhood of the upright configuration (node $(3,1)$, $\mathcal{A}_u$). Finally, a switch to the local stabilizer κ_r ($q = -3$) (at around 9.5sec.) steers x to the origin.*

Figure 4 shows a simulation of the closed-loop system resulting from controlling the pendubot with our hybrid controller. The initial state of the pendubot is such that it is far away from the regions where the open-loop laws and the local stabilizer κ_u are applicable. Therefore, the hybrid controller initially switches to *recovery mode* ($q = p = 0$) and applies κ_0. (In this case, the controller κ_0 was designed to be given by $-L_g V := -\langle \nabla V(x), g(x) \rangle$ where V is the kinetic plus potential energy of the pendubot and g is such that $f(x, u) = \tilde{f}(x) + g(x)u$. This controller removes energy faster than $\kappa_0 \equiv 0$. With this controller, the angles of the pendulums reach a neighborhood of $-\pi$ and the angular velocities a neighborhood of 0. Then, the hybrid controller switches to *throw mode* in the first path and from node $(2, 1)$ to node $(3, 1)$ ($q = 2, p = 1$). The open-loop control law applied is $\alpha_{r \to u}$ which steers the state x to a neighborhood of the origin. In that event, a switch to the local stabilizer κ_u follows, and the state x converges to the origin asymptotically.

III. Throw-Catch Hybrid Control For General Non-linear Systems

We now generalize the control strategy described for the pendubot to general nonlinear control systems. Consider the nonlinear control system

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

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $x \in \mathbb{R}^n$ is the state, and $u \in \mathbb{R}^m$ is the control input. Let $\mathcal{A} \subset \mathbb{R}^n$ be compact; $P := \{1, 2, \ldots, p_{\max}\} \subset \mathbb{N}$, $p_{\max} \geq 1$; for each $j \in P$, $Q_j := \{1, 2, \ldots, q^j_{\max}\} \subset \mathbb{N}$, $q^j_{\max} \geq 2$; and $R := \cup_{k \in P} (Q_k \times \{k\})$. We assume the following.

Assumption 3.1: The function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is continuous. For each $(i, j) \in R$, there exist:

1. Disjoint compact sets $\mathcal{A}_{i,j} \subset \mathbb{R}^n$ satisfying $\mathcal{A}_{i,j} = \mathcal{A}$ for each $i = q^j_{\max}$, $j \in P$.
2. When $i > 1$, continuous state-feedback laws $\kappa_{i,j} : \mathbb{R}^n \to \mathbb{R}^m$ such that the compact set $\mathcal{A}_{i,j}$ is asymptotically stable with basin of attraction $\mathcal{B}_{\mathcal{A}_{i,j}} \subset \mathbb{R}^n$ for $\dot{x} = f(x, \kappa_{i,j}(x))$.
3. When $i < q^j_{\max}$, piecewise-continuous functions $\alpha_{(i,j) \to (i+1,j)} : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ that are capable of steering trajectories of (3) from a set $S_{i,j}$ to an open set $E_{i,j}$ in finite time with maximum time $\tau^*_{i,j} \geq 0$, where $S_{i,j} \subset \mathbb{R}^n$ contains an open neighborhood of $\mathcal{A}_{i,j}$, $E_{i,j}$ contains an open neighborhood of $\mathcal{A}_{i+1,j}$ and is such that an open $\delta^c_{i,j}$-neighborhood of itself, $\delta^c_{i,j} > 0$, is contained in $\mathcal{B}_{\mathcal{A}_{i+1,j}}$.

Moreover, there exists:

4. Continuous state-feedback law $\kappa_0 : \mathbb{R}^n \to \mathbb{R}^m$, such that, for each solution x to $\dot{x} = f(x, \kappa_0(x))$ there exists finite $T > 0$ such that $x(T)$ is in the union of each of the sets $E_{i,j} + \frac{\delta^c_{i,j}}{2}\mathbb{B}$ and $S_{i,j}$ above (this corresponds to a "bootstrap" feedback controller) [1].

Remark 3.2: In most applications, the compact sets $\mathcal{A}_{i,j}$, $(i, j) \in R$, are given by single points, in particular equilibrium points, for which local regulation of the trajectories of (3) is known with the state-feedback laws $\kappa_{i,j}$. The functions $\alpha_{(i,j) \to (i+1,j)}$ are functions of time that can be recorded in the memory of the digital controller. ■

The compact sets $\mathcal{A}_{i,j}$, $(i, j) \in R$, define a *directed tree* in the sense that for every compact set $\mathcal{A}_{i,j}$ with $i < q^j_{\max}, j \in P$, there exists an open-loop control law that transfers the state from nearby points of $\mathcal{A}_{i,j}$ to nearby points of $\mathcal{A}_{i+1,j}$. Every path has the last node in common and first independent nodes defining the paths which eventually merge with other paths. This connectivity between nodes is denoted in Figure 5 by a directed arc joining the node $\mathcal{A}_{i,j}$ with the node $\mathcal{A}_{i+1,j}$. Note that Assumption 3.1.4 guarantees the existence of a state-feedback law κ_0 such that, when the trajectories are away from the basin of attraction of the local stabilizers or at points where the open-loop control laws are not able to transfer the state to the next node, the trajectories are steered back to the tree.

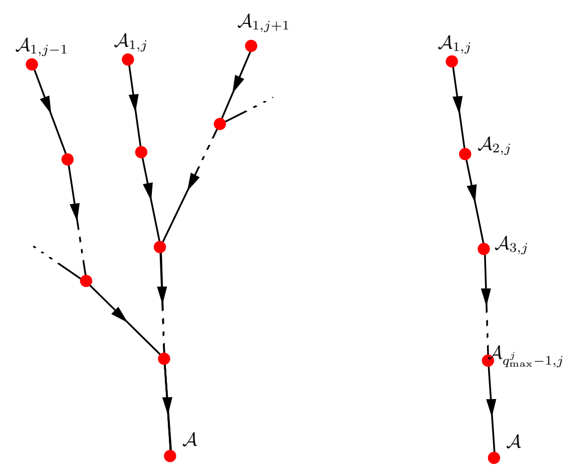

Fig. 5. *General case of directed tree (left) and j-th path (right).*

A. Control Design

We design a state-feedback hybrid controller, which we denote by $\mathcal{H}_c$, that performs the switching between the feedback control laws κ_0, $\kappa_{i,j}$, and the functions $\alpha_{(i,j) \to (i+1,j)}$ in Assumption 3.1. We follow the framework for hybrid systems in [4], [5] where solutions are given on *hybrid time domains*[2].

Let $Q^c_j := \{-q^j_{\max}, -q^j_{\max} + 1, \ldots, -2\}$ and $Q^t_j := \{1, 2, \ldots, q^j_{\max} - 1\}$, for each $j \in P$; $Q := \cup_{j \in P} (Q^c_j \cup Q^t_j)$; $L^t := \cup_{j \in P} (Q^t_j \times \{j\})$; and $L := (\cup_{j \in P} ((Q^c_j \cup Q^t_j) \times \{j\})) \cup (0, 0)$. The controller state is given by $[q\ p\ \tau]^T$, where q and p are logic states and $\tau \in \mathbb{R}$ is a timer state. The logic state p takes value in $P \cup \{0\}$ and the logic state q takes value in $Q \cup \{0\}$. They store the state of the system:

- "Catch mode" at the $|q|$-th node of the p-th path when $q \in Q^c_p$, $p \in P$.

[1] $\delta\mathbb{B}$ denotes the open ball of radius $\delta > 0$ centered at the origin in $\mathbb{R}^n$.

[2] In this framework, a solution x to a hybrid system on a hybrid time domain $\operatorname{dom} x$ is parameterized by a continuous variable t which keeps track of the continuous dynamics and a discrete variable j which keeps track of the discrete dynamics. Then, $x(t, j)$ is the value of the solution at time $(t, j) \in \operatorname{dom} x$. For more details, see [4], [5].

- "Throw mode" at the q-th node of the p-th path when $q \in Q_p^t$, $p \in P$.
- "Recovery mode" when $q = p = 0$.

Following the control logic outlined in Section II-C, the output of the hybrid controller is given by

$$\kappa_c(x,q,p,\tau) := \begin{cases} \kappa_{|q|,p}(x) & \text{if } q \in Q_p^c \\ \alpha_{(q,p)\to(q+1,p)}(\tau) & \text{if } q \in Q_p^t \\ \kappa_0(x) & \text{if } q = 0 \ . \end{cases} \quad (4)$$

We now design several sets used in the control logic.

I) **Sets for "Catch mode" update logic**

For each $(i,j) \in L^t$, let $E_{i,j}$ and $\delta_{i,j}^c$ be given as in Assumption 3.1.3, and define

$$D_{i,j}^c = E_{i,j} + \delta_{i,j}^c \overline{\mathbb{B}}, \qquad C_{i,j}^c = \overline{\mathbb{R}^n \setminus D_{i,j}^c} + \frac{\delta_{i,j}^c}{2}\overline{\mathbb{B}} \ .$$

II) **Sets for "Throw mode" update logic**

For each $(i,j) \in L^t$, let $S_{i,j}$ be given as in Assumption 3.1.3, and define $D_{i,j}^t$ to be a closed set such that for some $\delta_{i,j}^t > 0$ satisfies

$$\mathcal{A}_{i,j} + \delta_{i,j}^t \mathbb{B} \subset D_{i,j}^t, \qquad D_{i,j}^t + \frac{\delta_{i,j}^t}{2}\mathbb{B} \subset S_{i,j} \ .$$

Then, for each $(i,j) \in L^t$, let $C_{i,j}^t$ be given by

$$C_{i,j}^t := \overline{\mathbb{R}^n \setminus D_{i,j}^t} + \frac{\delta_{i,j}^t}{2}\overline{\mathbb{B}} \ .$$

III) **Sets for "Recovering mode" update logic**

For each $(i,j) \in \cup_{k\in P}(Q_k^c \times \{k\})$, for some $\delta_{i,j}^r > 0$, define

$$C_{i,j}^r := \mathcal{R}_{i,j}(D_{|i|-1,j}^c) + \delta_{i,j}^r \overline{\mathbb{B}} \quad (5)$$

where $\mathcal{R}_{i,j}(D_{|i|-1,j}^c)$ is the reachable set of $\dot{x} = f(x, \kappa_{|i|,j}(x))$ from $D_{|i|-1,j}^c$. Also, define $D_{i,j}^r$ as

$$D_{i,j}^r := \overline{\mathbb{R}^n \setminus C_{i,j}^r} + \frac{\delta_{i,j}^r}{2}\overline{\mathbb{B}} \ . \quad (6)$$

Define $C_{0,0}^r$ and $D_{0,0}^r$ as follows. For each $(i,j) \in \cup_{k\in P}(Q_k \times \{k\})$ define an auxiliary set $\tilde{D}_{i,j}^r$ to be a closed set such that for some $\delta_{i,j}^r > 0$ satisfies

$$\mathcal{A}_{i,j} + \delta_{i,j}^r \mathbb{B} \subset \tilde{D}_{i,j}^r, \qquad \tilde{D}_{i,j}^r + \frac{\delta_{i,j}^r}{2}\mathbb{B} \subset D_{i-1,j}^c \cup D_{i,j}^t \ ,$$

where $D_{0,j}^c = D_{q_{\max}^j,j}^t = \emptyset$ for all $j \in P$. Then, $C_{0,0}^r$ and $D_{0,0}^r$ are given by

$$D_{0,0}^r := \bigcup_{(i,j)\in\cup_{k\in P}(Q_k\times\{k\})} \tilde{D}_{i,j}^r, \quad C_{0,0}^r := \overline{\mathbb{R}^n \setminus D_{0,0}^r} + \frac{\delta_{0,0}^r}{2}\overline{\mathbb{B}}$$

where $\delta_{0,0}^r$ is the minimum $\delta_{i,j}^r$ over $\cup_{k\in P}(Q_k \times \{k\})$.

With these definitions, the update laws are designed as follows. If in *throw mode* and the state x is such that a "catch" is possible, i.e.

$$(q,p) \in Q_p^t \times \{p\}, \quad x \in D_{q,p}^c \ , \quad (7)$$

then jumps to catch mode are enabled with update law $q^+ = -(|q|+1)$. If in *catch mode* and the state x is such that a "throw" is possible, i.e.

$$(q,p) \in \left(Q_p^c \setminus \{-q_{\max}^p\}\right) \times \{p\}, \quad x \in D_{|q|,p}^t \ , \quad (8)$$

then jumps to throw mode are enabled with update law $q^+ = |q|$.

If in *throw mode* and the timer state τ is larger than $\tau_{q,p}^*$ or if in *catch mode* and the state x is such that $x \in D_{q,p}^r$ then jumps to *recovery mode* are enabled with update law $q^+ = 0, p^+ = 0$. While in this mode, the controller enables updates of (q,p) to a pair in $\cup_{k\in P}((Q_k^c \cup Q_k^t) \times \{k\})$ when $x \in D_{0,0}^r$.

The construction of the sets in I)-III) define the flow and jump sets of the hybrid controller (while in mode $q \in Q$ and in path $p \in P$, the sets $C_{|q|,p}^c$ and $D_{|q|,p}^c$; $C_{|q|,p}^t$ and $D_{|q|,p}^t$; and $C_{-|q|,p}^r$, $C_{0,0}^r$ and $D_{-|q|,p}^r$, $D_{0,0}^r$ define the flow and jump sets for jumps to *catch*, *throw*, and *recovery mode*, respectively). Figure 6 illustrates the sets used in the update law and a sample trajectory for the i-th compact set in the j-th path, $(i,j) \in L$, $i \in \{2,3,\ldots,q_{\max}^j - 1\}$.

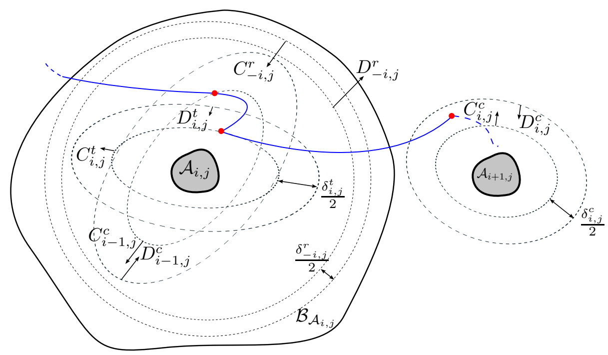

Fig. 6. The compact set $\mathcal{A}_{i,j}$, $(i,j) \in L$, $i \in \{2,3,\ldots,q_{\max}^j - 1\}$; the associated flow sets $C_{i,j}^c, C_{i,j}^t, C_{-i,j}^r$; and the jump sets $D_{i,j}^c, D_{i,j}^t, D_{-i,j}^r$ are depicted. The sets $C_{i-1,j}^c$ and $D_{i-1,j}^c$ associated with the compact set $\mathcal{A}_{i-1,j}$ are also shown for the computation of $C_{-i,j}^r$ and $D_{-i,j}^r$. Vaguely, the control strategy is such that with $q = i-1$ and $p = j$, a jump can occur as soon as the trajectory enters the set $D_{i-1,j}^c$, from where the local state feedback law $\kappa_{i,j}$ is applied. A jump that activates the control law $\alpha_{(i,j)\to(i+1,j)}$ can be triggered as soon as the trajectory hits the set $D_{i,j}^t$. The local stabilizer for $\mathcal{A}_{i+1,j}$ is enabled when the trajectory enters the set $D_{i,j}^c$. The sequence is repeated until the compact set $\mathcal{A}_{i^*,j}$, $i^* = q_{\max}^j$, is reached.

B. Hybrid controller

Let $\mathcal{X} := \mathbb{R}^n \times L \times \mathbb{R}$, $\overline{\tau} = \max \tau_{i,j}^*$ for all $(i,j) \in L^t$ where $\tau_{i,j}^*$ is given by Assumption 3.1.3, and $\xi := [x^T \ q \ p \ \tau]^T$. Our hybrid controller, denoted by $\mathcal{H}_c$, is

$$\mathcal{H}_c \begin{cases} (\dot{q},\dot{p}) & = (0,0), \quad \dot{\tau} = 1, \qquad \xi \in C_c \\ (q,p)^+ & \in g_c(\xi), \quad \tau^+ = 0, \qquad \xi \in D_c \end{cases}$$

where the sets C_c and D_c are given by

$$\begin{aligned} C_c \ := \ & \left\{ \xi \in \mathcal{X} \ \middle| \ q \in Q_p^c \setminus \{-q_{\max}^p\}, \ x \in C_{|q|,p}^t \cap C_{q,p}^r \right\} \\ \cup \ & \left\{ \xi \in \mathcal{X} \ \middle| \ q = -q_{\max}^p, \ x \in C_{q,p}^r \right\} \\ \cup \ & \left\{ \xi \in \mathcal{X} \ \middle| \ q \in Q_p^t, \ x \in C_{q,p}^c, \ \tau \le \tau_{q,p}^* \right\} \\ \cup \ & \left\{ \xi \in \mathcal{X} \ \middle| \ q = p = 0, \ x \in C_{0,0}^r \right\} \ , \end{aligned}$$

$$
\begin{aligned}
D_c &:= D_{c1} \cup D_{c2} \cup D_{c3} \ , \\
D_{c1} &:= \left\{ \xi \in \mathcal{X} \ \middle| \ q \in Q_p^t, \ x \in D_{q,p}^c, \ \tau \le \tau_{q,p}^* \right\} \cup \\
&\quad \left\{ \xi \in \mathcal{X} \ \middle| \ q = p = 0, \ (q',p') \in L^t, \ x \in D_{q',p'}^c \cap D_{0,0}^r \right\} , \\
D_{c2} &:= \left\{ \xi \in \mathcal{X} \ \middle| \ q \in Q_p^c \setminus \{-q_{\max}^p\}, \ x \in D_{|q|,p}^t \right\} \cup \\
&\quad \left\{ \xi \in \mathcal{X} \ \middle| \ q = p = 0, \ (q',p') \in L^t, \ x \in D_{q',p'}^t \cap D_{0,0}^r \right\} , \\
D_{c3} &:= \left\{ \xi \in \mathcal{X} \ \middle| \ q \in Q_p^t, \ \tau \ge \tau_{q,p}^* \right\} \cup \\
&\quad \left\{ \xi \in \mathcal{X} \ \middle| \ q \in Q_p^c, \ x \in D_{|q|,p}^r \right\} ,
\end{aligned}
$$

and the jump map g_c is given by

$$
g_c(\xi) := \begin{cases} g_{c1}(\xi) & \xi \in D_{c1} \\ g_{c2}(\xi) & \xi \in D_{c2} \\ (0,0) & \xi \in D_{c3} \end{cases}
$$

$$
g_{c1}(\xi) := \begin{cases} (-|q|-1, p) & \text{if } (q,p) \in L^t, \\ \\ \left\{ (-|q'|-1, p') \ \middle| \ (q',p') \in L^t, \ x \in D_{q',p'}^c \right\} \\ & \text{if } (q,p) = (0,0) \end{cases}
$$

$$
g_{c2}(\xi) := \begin{cases} (|q|, p) & \text{if } q \in Q_p^c \setminus \{-q_{\max}^p\}, \\ \\ \left\{ (q',p') \ \middle| \ (q',p') \in L^t, \ x \in D_{q',p'}^t \right\} \\ & \text{if } (q,p) = (0,0) \end{cases}
$$

and output κ_c given in (4).

The jump map g_{c1} and jump set D_{c1} implement the logic for *catch mode*, while g_{c2} and D_{c2} implement the logic *throw mode*. The first pieces of these sets correspond to the conditions in (7) and (8), while the second pieces allow jumps from *recovering* to *catch* or *throw mode*. Moreover, the definitions of D_{c1} and D_{c2} differ from the corresponding ones in (7)-(8) as they implement jumps that update the path state p to a correct one. Similarly for g_{c1} and g_{c2}. (This mechanism is needed when the initialization of the logic states is not correct.) The jump set D_{c3} states the conditions for jumps to *recovery mode*. The flow set C_c includes all points at which jumps are not allowed, and to guarantee robust existence of solutions (see e.g. the discussion at the end of [8, Section III]), it overlaps with the jump set D_c.

C. Stability and robustness to measurement noise

The closed-loop system resulting from controlling the nonlinear system (3) with the controller $\mathcal{H}_c$ is given by

$$
\left. \begin{array}{rcl} \dot{x} &=& f(x, \kappa_c(\xi)) \\ (\dot{q}, \dot{p}) &=& (0,0) \\ \dot{\tau} &=& 1 \end{array} \right\} \ \xi \in C_c
$$

$$
\left. \begin{array}{rcl} x^+ &=& x \\ (q,p)^+ &\in& g_c(\xi) \\ \tau^+ &=& 0 \end{array} \right\} \ \xi \in D_c \ .
$$

We denote this hybrid system by $\mathcal{H}_{cl}$. The hybrid controller $\mathcal{H}_c$ confers the following stability property.

Theorem 3.3: (nominal asymptotic stability) Let Assumption 3.1 hold. For the hybrid system $\mathcal{H}_{cl}$, the compact set $\mathcal{A} \times (\cup_{j \in P}(\{-q_{\max}^j\} \times \{j\})) \times [0, \overline{\tau}]$ is asymptotically stable with basin of attraction $\mathcal{B} := C_c \cup D_c$.

This result states that every solution to $\mathcal{H}_{cl}$ is such that the x component converges to $\mathcal{A}$ and that every solution with initial x close to $\mathcal{A}$ stays close for all time. This corresponds to global asymptotic stability of $\mathcal{A}$ for the the nonlinear system (3) controlled by $\mathcal{H}_c$. The proof of this result follows from the control logic implemented in $\mathcal{H}_c$. The open-loop schedules are used to steer trajectories from a neighborhood of one node to a neighborhood of the following node, and the state-feedback control laws steer the trajectories toward the nodes of the tree. The control logic in $\mathcal{H}_c$ is such that for every point in the state space, by measuring the state, a sequence of switches between the control laws takes the state of the system to $\mathcal{A}$.

The hybrid controller $\mathcal{H}_c$ confers a margin of robustness to measurement noise e on the state x. This is stated in the following result. Below, $|x|_{\mathcal{A}} = \inf_{y \in \mathcal{A}} |x - y|$. Also, recall that $\mathrm{dom}(x,q,p,\tau)$ denotes the domain of the solution (x,q,p,τ) to $\mathcal{H}_{cl}$.

Theorem 3.4: (robustness to measurement noise) Let Assumption 3.1 hold. Then, there exists $\beta \in \mathcal{KL}$, for each $\varepsilon > 0$ and each compact set $K \subset \mathcal{B}$ there exists $\delta^* > 0$, such that for each measurement noise $e : \mathbb{R}_{\ge 0} \to \delta^* \mathbb{B}$, solutions (x,q,p,τ) to $\mathcal{H}_{cl}$ exist, are complete, and for initial conditions $(x^0, q^0, p^0, \tau^0) \in K$ the x component of the solutions satisfies

$$
|x(t,j)|_{\mathcal{A}} \le \beta(|x^0|_{\mathcal{A}}), t + j) + \varepsilon \quad \forall (t,j) \in \mathrm{dom}(x,q,p,\tau).
$$

The proof of this result follows by the regularity properties of the data of $\mathcal{H}_{cl}$ and the results for perturbed hybrid systems in [5]. Due to space limitations, we do not discuss the concept and the issues on existence of solutions to hybrid systems with measurement noise here. See [8] for more details.

In addition to the property in Theorem 3.4, the hybrid controller $\mathcal{H}_c$ confers an additional robustness property to the closed-loop system when the open-loop schedules are in the loop. When a disturbance or failure prevents a “throw” from being successful, the recovery logic implemented in the hybrid controller steers the state of the system back to the tree and retries the “throw-catch” sequence.

REFERENCES

[1] P.-A. Absil and R. Sepulchre. A hybrid control scheme for swing up acrobatics. In *European Conference on Control ECC*, 2001.

[2] I. Fantoni, R. Lozano, and M. W. Spong. Energy based control of the pendubot. *IEEE Trans. Aut. Cont.*, 45:725–729, 2000.

[3] E. Frazzoli, M. A. Dahleh, and E. Feron. Maneuver-based motion planning for nonlinear systems with symmetries. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 21:1077–1091, 2005.

[4] R. Goebel, J.P. Hespanha, A.R. Teel, C. Cai, and R.G. Sanfelice. Hybrid systems: Generalized solutions and robust stability. In *Proc. 6th IFAC Symposium in Nonlinear Control Systems*, pages 1–12, 2004.

[5] R. Goebel and A.R. Teel. Solutions to hybrid inclusions via set and graphical convergence with stability theory applications. *Automatica*, 42(4):573–587, 2006.

[6] J.Rubi, A.Rubio, and A.Avello. Swing-up control problem for a self-erecting double inverted pendulum. *IEE Proc.- Control Theory Applications*, 149:169–175, 2002.

[7] C. Prieur. Uniting local and global controllers with robustness to vanishing noise. *Math. Control Signals Systems*, 14:143–172, 2001.

[8] R. G. Sanfelice, A. R. Teel, R. Goebel, and C. Prieur. On the robustness to measurement noise and unmodeled dynamics of stability in hybrid systems. In *Proc. 25th American Control Conference*, pages 4061 – 4066, 2006.