Pre-orders for Reasoning about Stability Properties with respect to Input of Hybrid Systems

Pavithra Prabhakar  
IMDEA Software Institute  
Madrid, Spain  
pavithra.prabhakar@imdea.org

Jun Liu  
University of Sheffield  
Sheffield, UK  
j.liu@sheffield.ac.uk

Richard M. Murray  
California Inst. of Tech.  
Pasadena, CA, USA  
murray@caltech.edu

ABSTRACT

Pre-orders on systems are the basis for abstraction based verification of systems. In this paper, we investigate pre-orders for reasoning about stability with respect to inputs of hybrid systems. First, we present a superposition type theorem which gives a characterization of the classical incremental input-to-state stability of continuous systems in terms of the traditional $c$-$\delta$ definition of stability. We use this as the basis for defining a notion of incremental input-to-state stability of hybrid systems. Next, we present a pre-order on hybrid systems which preserves incremental input-to-state stability, by extending the classical definitions of bisimulation relations on systems with input, with uniform continuity constraints. We show that the uniform continuity is a necessary requirement by exhibiting counter-examples to show that weaker notions of input bisimulation with just continuity requirements do not suffice to preserve stability. Finally, we demonstrate that the definitions are useful, by exhibiting concrete abstraction functions which satisfy the definitions of pre-orders.

Categories and Subject Descriptors

D.2.4 [Software Engineering]: Software/Program Verification—Formal Methods

General Terms

Design, Verification

Keywords

Stability, (Bi)-Simulations, Abstraction, Verification

1. INTRODUCTION

The ubiquitous use of embedded processors to control safety critical systems such as aeronautics, automotive and medical devices, has pressurized the need for scalable methods for reliable development of embedded control systems. A unique feature of such systems is the interaction between the discrete components - a network of embedded processors, and the continuous components - the physical system that the processors control. Such systems with mixed discrete continuous behaviors have been popularly termed hybrid systems.

In this paper, we develop the foundations for scalable verification of a robustness property of hybrid systems, namely, stability with respect to inputs. Intuitively, we expect small perturbations in the system input to lead to only small changes in the behavior of the system. The small perturbations in input capture disturbances such as quantization errors in actuators and sensors, which are often ignored during control design, but are, nevertheless, present in any digital implementation of the control law. Hence, stability of systems with respect to input is an important property required of any control design.

Input-to-state stability [22, 21] is the classical notion of stability with respect to inputs for purely continuous systems. Incremental input-to-state stability [2] generalizes input-to-state stability with respect to equilibrium points to that with respect to reference trajectories. The notion of input-to-state stability has been defined for hybrid systems and has been well-investigated [4, 18, 13]. More recently, input-to-output stability notions have been explored for discrete systems [23]. However, the notion of incremental input-to-state stability has not been investigated in the literature for hybrid systems. We present a characterization of the classical definition of incremental input-to-state stability for continuous systems as a super-position theorem “separating” the stability with respect to states and with respect to inputs. The characterization gives an $c$-$\delta$ definition similar to that for Lyapunov and asymptotic stability. We generalize this definition to the setting of hybrid systems.

Current methods for verifying stability are deductive. They rely on exhibiting a certificate of stability in the form of a Lyapunov function (see, for example, [16]) - a continuously differentiable function which is positive definite and whose value decreases along any trajectory of the system. Automation of the search for a Lyapunov function essentially relies on searching for the coefficients of a template, say, a polynomial, such that the requirements of the Lyapunov function are satisfied [19]. However, there is little support for systematically iterating over templates so as to prune the search space.

The work in the paper is motivated by an alternate approach, namely, algorithmic verification [6]. It is a completely automatic approach in which an exhaustive state
space exploration is performed to deduce the property. However, it suffers from the state space explosion problem, and the crucial element in achieving algorithmic verification methods is to devise efficient abstraction mechanisms [1, 24]. An abstraction reduces the state space of the system such that the satisfaction of the property by the original system can be deduced by the property about the reduced system. However, if the reduced system does not satisfy the property, one cannot infer useful information about the original system. Hence, abstraction techniques are often accompanied by an abstraction refinement loop [7, 1], where a refinement is an abstraction of the system which is more precise than the current abstraction.

The basis for developing an abstraction refinement technique lies in understanding the relations between systems (the concrete and the abstract) which preserve the properties of interest. In this paper, we investigate such relations which preserve stability properties of hybrid systems with respect to input. More precisely, we define a pre-order - a reflexive, transitive relation - on the class of hybrid systems, such that if a system is stable with respect to inputs, then all system below it in the order are stable as well.

Simulations and bisimulations [17] are the canonical notions of pre-order and equivalence on systems with respect to several discrete-time properties such as LTL, CTL and μ-calculus. Alternate weaker notions such as approximate simulations and bisimulations have been investigated to achieve state-space reduction [10, 11]. It has been shown in [8, 20] that simulations and bisimulations do not preserve Lyapunov and asymptotic stability and hence, additional constraints such as continuity and uniform continuity are imposed on the simulation and bisimulation relations to force stability preservation. In this paper, we investigate incremental input-to-state stability. We show that the classical definitions of input simulations and bisimulations do not preserve incremental input-to-state stability, even with continuity constraints imposed on the input and state spaces. Hence, uniform continuity constraints are added to the relations corresponding to both the state space and the input space. Finally, we demonstrate that the definitions are reasonable, by exhibiting concrete abstraction functions which satisfy the definition of the pre-order. To this end, we cast the well-known Lyapunov function based analysis as a concrete abstraction function which reduces a system to a simple one-dimensional system, for which incremental input-to-state stability can be easily inferred. The future work will focus on developing new abstraction-refinement mechanism.

To summarize, the main contributions of the paper are:

1. A superposition type theorem for incremental input-to-state stability of continuous dynamical systems.

2. Using the alternate characterization of incremental input-to-state stability of continuous dynamical systems given by the superposition theorem to define a notion of incremental input-to-state stability for hybrid systems.

3. A definition of pre-order on the class of hybrid systems which respects input-to-state stability and thus defines the basis for an abstraction refinement framework.

4. Examples demonstrating concrete abstraction functions which fall under the definition of the pre-order.

**Organization of the paper.**

In Section 2, we define the necessary definitions. In Section 3, we define hybrid input transition systems and related concepts. In Section 4, we present a superposition type characterization of the notion of incremental input-to-state stability of continuous dynamical systems, which we use in Section 5 to present a definition of incremental input-to-state stability for hybrid systems. In Section 6, we define the pre-orders on hybrid systems, namely, uniformly continuous input simulations and bisimulations, and show the inadequacy of weaker notions for incremental input-to-state stability preservation. In Section 7, we show that the new pre-order introduced in Section 6 preserves incremental input-to-state stability. Finally, in Section 8, we present concrete abstraction techniques based on Lyapunov function based incremental input-to-state stability analysis, and conclude with Section 9.

**2. PRELIMINARIES**

**Notation.**

Let \( \mathbb{R} \) and \( \mathbb{R}^+ \) denote the set of reals and non-negative reals, respectively. Let \( \mathbb{R}_\infty \) denote the set \( \mathbb{R}^+ \cup \{ \infty \} \), where \( \infty \) denotes the largest element of \( \mathbb{R}_\infty \), that is, \( x < \infty \) for all \( x \in \mathbb{R}^+ \). Also, for all \( x \in \mathbb{R}_\infty \), \( x + \infty = \infty \). Let \( \mathbb{N} \) denote the set of all natural numbers \( \{ 0, 1, 2, \ldots \} \), and let \( [n] \) denote the first \( n \) natural numbers, that is, \( [n] = \{ 0, 1, 2, \ldots, n-1 \} \). Let \( \text{PreInt} \) denote the set of all closed intervals of the form \( [0, T] \), where \( T \in \mathbb{R}^+ \), and the infinite interval \( [0, \infty) \). Given an \( x \in \mathbb{R}^+ \), we use \( |x| \) to denote the Euclidean norm of \( x \). And, given a function \( f : A \to \mathbb{R}^m \), we use \( \|f\|_\infty \) to denote sup\( a \in A \) \( |f(a)| \).

**Functions and Relations.**

Given a function \( F \), let \( \text{Dom}(F) \) denote the domain of \( F \). Given a function \( F : A \to B \) and a set \( A' \subseteq A \), \( F(A') \) denotes the set \( \{ F(a) \mid a \in A' \} \). Given a binary relation \( R \subseteq A \times B \), \( R^{-1} \) denotes the set \( \{ (x, y) \mid (y, x) \in R \} \). For a binary relation \( R \), we will interchangeably use \( (x, y) \in R \) and \( “(x, y) \in R” \) to denote that \( (x, y) \in R \).

**Sequences.**

A sequence \( \sigma \) is a function whose domain is either \( [n] \) for some \( n \in \mathbb{N} \) or the set of natural numbers \( \mathbb{N} \). We denote the set of all domains of sequences as \( \text{SeqDom} \). Length of a sequence \( \sigma \), denoted \( |\sigma| \), is \( n \) if \( \text{Dom}(\sigma) = [n] \) or \( \infty \) otherwise. Given a sequence \( \sigma : \mathbb{N} \to \mathbb{R} \) and an element \( r \in \mathbb{R}_\infty \), we use \( \sum_{i=0}^{\infty} |\sigma(i)| = r \) to denote the standard limit condition \( \lim_{N \to \infty} \sum_{i=0}^{N} |\sigma(i)| = r \).

**Extended Metric Space.**

An extended metric space is a pair \( (M, d) \) where \( M \) is a set and \( d : M \times M \to \mathbb{R}_\infty \) is a distance function such that for all \( m_1, m_2, m_3 \), the following hold: (Identity of indiscernibles) \( d(m_1, m_2) = 0 \) if and only if \( m_1 = m_2 \), (Symmetry) \( d(m_1, m_2) = d(m_2, m_1) \), and (Triangle inequality) \( d(m_1, m_3) \leq d(m_1, m_2) + d(m_2, m_3) \). When the metric on \( M \) is clear we will simply refer to \( M \) as a metric space.

We define an open ball of radius \( r \) around a point \( x \) to be the set of all points which are within a distance \( r \) from \( x \). Formally, an open ball is a set of the form \( B_r(x) = \{ y \in M \mid d(x, y) < r \} \). An open set is a subset of \( M \) which is a
union of open balls. Given a set $X \subseteq M$, a neighborhood of $X$ is an open set in $M$ which contains $X$. Given a subset $X$ of $M$, an $\epsilon$-neighborhood of $X$ is the set $B_\epsilon(X) = \bigcup_{x \in X} B_\epsilon(x)$. A subset $X$ of $M$ is compact if for every collection of open sets $\{U_\alpha\}_{\alpha \in \Delta}$ such that $X \subseteq \bigcup_{\alpha \in \Delta} U_\alpha$, there is a finite subset $J$ of $\Delta$ such that $X \subseteq \bigcup_{\alpha \in J} U_\alpha$.

Set Valued Functions.
We consider set valued functions and define continuity of these functions. We choose not to treat set valued functions as single valued functions whose co-domain is a power set, since as argued in [3], it leads to strong notions of continuity, which are not satisfied by many functions. A set valued function $F : A \rightrightarrows B$ is a function which maps every element of $A$ to a set of elements in $B$. Given a set $A' \subseteq A$, $F(A')$ will denote the set $\bigcup_{a \in A'} F(a)$. Given a binary relation $R \subseteq A \times B$, we use $R$ also to denote the set valued function $R : A \rightrightarrows B$ given by $R(x) = \{ y \mid (x, y) \in R \}$. Further, $F^{-1} : B \rightrightarrows A$ will denote the set valued function which maps $b \in B$ to the set $\{ a \in A \mid b \in F(a) \}$.

Continuity of Set Valued Functions.
Let $F : A \rightrightarrows B$ be a set valued function, where $A$ and $B$ are extended metric spaces. We define upper semi-continuity of $F$ which is a generalization of the “$\epsilon, \delta$ - definition” of continuity for single valued functions [3]. The function $F : A \rightrightarrows B$ is said to be upper semi-continuous at $a \in Dom(F)$ if and only if

$$\forall \epsilon > 0, \exists \delta > 0 \text{ such that } F(B_\delta(a)) \subseteq B_\epsilon(F(a)).$$

If $F$ is upper semi-continuous at every $a \in Dom(F)$ we simply say that $F$ is upper semi-continuous. Next we define a “uniform” version of the above definition, where, analogous to the case of single valued functions, corresponding to an $\epsilon$, there exists a $\delta$ which works for every point in the domain.

Definition. A function $F : A \rightrightarrows B$ is said to be uniformly continuous if and only if

$$\forall a \in Dom(A), F(B_\delta(a)) \subseteq B_\epsilon(F(a)).$$

Given an $\epsilon > 0$, we call a $\delta > 0$ satisfying the above condition, a uniformity constant of $F$ corresponding to $\epsilon$. We refer to uniform upper semi-continuity as just uniform continuity, because it turns out that the two notions of upper and lower semi-continuity coincide with the addition of uniformity condition, i.e., uniform upper semi-continuity is equivalent to uniform lower semi-continuity.

Class $K_\epsilon$, $L$, $K_\infty$ and $K_L$ functions.
A continuous function $\alpha : [0, a) \to [0, \infty)$ is said to belong to class $K$ if it is strictly increasing and $\alpha(0) = 0$. A continuous function $\alpha : [0, \infty) \to [0, \infty)$ is said to belong to class $K_\infty$ if $\alpha$ is a class $K$ function and $\alpha(r) \to \infty$ as $r \to \infty$. A continuous function $\varphi : [0, \infty) \to [0, \infty)$ is said to be of class $L$ if it is monotonically decreasing and $\lim_{r \to \infty} \varphi(r) = 0$. A continuous function $\beta : [0, a) \times [0, \infty) \to [0, \infty)$ is a class $K_L$ function if it is a class $K$ function with respect to the first argument and class $L$ with respect to the second argument, that is, for a fixed $s$, $\beta(r, s)$ is a class $K$ function and for a fixed $r$, $\beta(r, s)$ is a class $L$ function.

3. HYBRID SYSTEMS WITH INPUT

In this section, we present a general formalism for representing hybrid systems with inputs, called hybrid input transition systems. Hybrid systems are systems exhibiting mixed discrete-continuous behaviors. We represent the continuous behavior using a pair of input and state trajectories which capture the values of input and state over an interval of time; and represent the discrete behavior using transitions which capture instantaneous changes to the state due to impulse inputs. We will not concern ourselves with the exact representation of the models, see, for example, the hybrid automaton model [14]. However, our abstract model captures the behaviors arising from a hybrid automaton model.

3.1 Trajectories
A trajectory $\tau$ over a set $A$ is a function $\tau : I \to A$, where $I \in \text{PreInt}$. We denote the set of all trajectories over $A$ as $\text{Traj}(A)$. Let us define a function $\text{Size} : \text{Traj}(A) \to \mathbb{R}_\infty$ which assigns a size to the trajectories. For $\tau \in \text{Traj}(A)$, $\text{Size}(\tau) = T$ if $\text{Dom}(\tau) = [0, T]$ and $\text{Size}(\tau) = \infty$ if $\text{Dom}(\tau) = [0, \infty)$.

Relating trajectories.
Given a relation $R \subseteq A_1 \times A_2$ and trajectories $a_1 \in \text{Traj}(A_1)$ and $a_2 \in \text{Traj}(A_2)$, we say that $a_1$ and $a_2$ are related by $R$, denoted $R(a_1, a_2)$ if $\text{Dom}(a_1) = \text{Dom}(a_2)$ and for every $t \in \text{Dom}(a_1)$, $R(a_1(t), a_2(t))$. We use $R(a_1)$ to denote the set $\{ a_2 \mid R(a_1, a_2) \}$.

Input-State Trajectories.
An input-state trajectory specifies the state evolution on an input signal. Let us fix an input space $U$ and a state space $S$. An input-state trajectory over a pair $(U, S)$ is a pair of trajectories $(u, s)$ from $\text{Traj}(U) \times \text{Traj}(S)$ such that $\text{Dom}(u) = \text{Dom}(s)$. We call $u$ an input trajectory and $s$ a state trajectory. We will use $\text{ISTraj}(U, S)$ to denote the set of all input-state trajectories over $(U, S)$.

Size, First, Last, States, Inputs of Input-State Trajectories.
We extend $\text{Size}$ to input-state trajectories in the natural way, namely, $\text{Size}(u, s) = \text{Size}(u) = \text{Size}(s)$. We use $\text{First}(u, s)$ to denote the initial state, that is, $s(0)$, and $\text{Last}(u, s)$ to denote the last state, that is, $s(\text{Size}(s))$, if $\text{Size}(s)$ is not $\infty$, and is not defined otherwise. Given a state trajectory $s$, we use $\text{States}(s)$ to denote the set of states occurring in $s$, namely, $\{ s(t) \mid t \in \text{Dom}(s) \}$. Also, for an input-state trajectory we use $\text{States}(u, s)$ to denote $\text{States}(s)$.

Similarly, for an input trajectory $u$, we use $\text{Inputs}(u)$ to denote the set of inputs occurring in $u$, namely, $\{ u(t) \mid t \in \text{Dom}(u) \}$.

3.2 Transitions
A transition specifies the instantaneous change in a state resulting from an input impulse. A transition over a pair $(U, S)$ is an element of $U \times (S \times S)$. A transition $(u, (s_1, s_2))$ denotes the fact that if an input impulse $u$ is applied to the system in state $s_1$, then the system state changes to $s_2$. We will represent a transition $(u, (s_1, s_2))$ as $s_1 \rightarrow u \rightarrow s_2$. We denote the set of all transition over a pair $(U, S)$ as $\text{Trans}(U, S)$.
Size, First, Last, States, Inputs of Transitions.

We define Size of a transition \((u, (s_1, s_2))\) to be 0. As before, given \(\tau = (u, (s_1, s_2))\), we use First(\(\tau\)) and Last(\(\tau\)) to denote the state of the system before and after the transition, namely, \(\text{First}(\tau) = s_1\) and \(\text{Last}(\tau) = s_2\). Also, \(\text{First}(s_1, s_2) = s_1\) and \(\text{Last}(s_1, s_2) = s_2\). Similarly, \(\text{States}(s_1, s_2) = \text{States}(u, (s_1, s_2)) = \{s_1, s_2\}\). And, \(\text{Inputs}(u) = \{u\}\), for an input \(u\).

### 3.3 Hybrid Input Transition Systems

We can now define a hybrid input transition system as consisting of sets of input-state trajectories and transitions.

**Definition.** A hybrid input transition system (HITS) \(\mathcal{H}\) is a tuple \((S, U, \Sigma, \Delta)\), where \(S\) is a set of states, \(U\) is a set of inputs, \(\Sigma \subseteq \text{Trans}(U, S)\) is a set of transitions and \(\Delta \subseteq \text{Int}_\text{Traj}(U, S)\) is a set of input-state trajectories.

We will just use hybrid system or hybrid transition system to refer to the above entity. Next, we define an execution of a hybrid transition system, which is a behavior of the system.

An execution is a finite or infinite sequence of trajectories and transitions which have matching end-points.

**Definition.** An execution of a hybrid input transition system \(\mathcal{H}\) is a sequence \(\sigma : M \rightarrow \Sigma \cup \Delta\), where \(M \in \text{SeqDom}\) such that for each \(0 \leq i < |\sigma| - 1\), \(\text{Last}(\sigma(i)) = \text{First}(\sigma(i+1))\). Let \(\text{Exec}(\mathcal{H})\) denote the set of all executions of \(\mathcal{H}\).

We can view an execution as a pair consisting of an input signal and state signal. Let \(\sigma \in \text{Exec}(\mathcal{H})\). Then for each \(i \in \text{Dom}(\sigma)\), \(\sigma(i) = (u_i, s_i)\), where either \((u_i, s_i)\) is an input-state trajectory or a transition. Let \(\sigma^u\) and \(\sigma^s\) be sequences whose domain is the same as \(\sigma\) such that \(\sigma^s(i) = u_i\) and \(\sigma^u(i) = s_i\). Then we also use \(\{\sigma^u, \sigma^s\}\) to denote the execution \(\sigma\).

Given a set of executions \(T\) and an input signal \(\sigma^u\), we use \(T_{\sigma^u}\) to denote the set of all state signals of executions in \(T\) which result from application of the input signal \(\sigma^u\). Formally, \(T_{\sigma^u} = \{\sigma^s | (\sigma^u, \sigma^s) \in T\}\).

**First, Last, States, Inputs of Executions.**

We extend first and last to executions and state signals in the natural way, that is, the first of the first element in the sequence and the last of the last element if the sequence is finite. Formally, for an execution or a state signal \(\sigma\), \(\text{First}(\sigma) = \text{First}(\sigma(0))\) and \(\text{Last}(\sigma)\) is defined only if \(\text{Dom}(\sigma) = [n]\) for some \(n \in \mathbb{N}\) and is equal to \(\text{Last}(\sigma(n))\). Similarly, \(\text{States}(\sigma) = \bigcup_{i \in \text{Dom}(\sigma)} \text{States}(\sigma(i))\). Also, for an input signal \(\sigma^u\), \(\text{Inputs}(\sigma^u) = \bigcup_{i \in \text{Dom}(\sigma^u)} \text{Inputs}(\sigma^u(i))\). The functions are extended to sets of trajectories, state signals and executions in a natural manner. Let \(\text{States}(\mathcal{H})\) denote \(\text{States}(\Sigma) \cup \text{States}(\Delta)\) and \(\text{Inputs}(\mathcal{H})\) denote \(\text{Inputs}(\Sigma) \cup \text{Inputs}(\Delta)\).

**Graph of an execution.**

In order to define distance between executions, we interpret the input and state signals as sets called the graphs which have information about the linear ordering between the states and inputs at various times. The set corresponding to a state signal \(\sigma^s\) consists of triples \((t, i, s)\) such that \(s\) is a state that is reached after time \(t\) has elapsed along the execution, and \(i\) is the number of discrete transitions that have taken place before time \(t\). Similarly, the set corresponding to an input signal \(\sigma^u\) consists of triples \((t, i, u)\) such that the input \(u\) was applied at time \(t\), and the number of impulse inputs applied before time \(t\) is \(i\).

**Definition.** For an input or state signal \(\sigma^u\) and \(j \in \text{Dom}(\sigma^u)\), let \(T_j = \sum_{k=0}^{j-1} \text{Size}(\sigma^s(k))\) and \(K_j = \{k | k < j, \sigma^u(k)\text{ is not a trajectory}\}\). The graph of the signal \(\sigma^u\), denoted \(gr(\sigma^u)\), is the set of all triples \((t, i, x)\) such that there exists \(j \in \text{Dom}(\sigma^u)\) satisfying the following:

- \(t \in [T_j, T_j + \text{Size}(\sigma^u(j))]\).
- If \(\sigma^u(j)\) is a trajectory, then \(i = K_j\) and \(x = \sigma^u(j)(t - T_j)\).
- If \(\sigma^u(j)\) is not a trajectory, then
  - if \(\sigma^u\) is a state signal and \(\sigma^u(j) = (s_1, s_2)\), then either \(i = K_j\) and \(x = s_1\), or \(i = K_j + 1\) and \(x = s_2\).
  - if \(\sigma^u\) is an input signal and \(\sigma^u(i) = u\), then \(i = K_j\) and \(x = u\).

### 3.4 Metric Hybrid Input Transition System

In order to reason about stability of a system, one needs a notion of distance between behaviors of the system. Hence, we extend the definition of the hybrid system with a metric on the states and inputs which can then be extended to distance between signals and executions.

A metric hybrid input transition system is a hybrid input transition system whose state and input spaces are equipped with a metric. A metric hybrid input transition system (MHTS) is a pair \((\mathcal{H}, d^s, d^u)\) where \(\mathcal{H} = (S, U, \Sigma, \Delta)\) is a hybrid input transition system, and \((S, d^s)\) and \((U, d^u)\) are extended metric spaces. The metric \(d^s\) on the state space can be lifted to state signals and \(d^u\) to input signals, which will then be used to define input-to-state stability notions.

Before defining this extension, recall that given an extended metric space \((M, d)\), the Hausdorff distance between \(A, B \subseteq M\), also denoted \(d(A, B)\), is given by the maximum of \(\{\sup_{p \in A} \inf_{q \in B} d(p, q), \sup_{q \in B} \inf_{p \in A} d(p, q)\}\).

We extend \(d\) to triples used in the definition of graphs.

**Definition.** For \((t_1, t_1, x_1), (t_2, t_2, x_2) \in \mathbb{R}^+ \times \mathbb{N} \times M\), let \(d((t_1, t_1, x_1), (t_2, t_2, x_2)) = \max\{|t_1 - t_2|, |t_1 - t_2|, d(x_1, x_2)|\}\).

Now we can define the distance between state signals and input signals.

**Definition.** Let \((\mathcal{H}, d^s, d^u)\) be a metric hybrid input transition system with \(\mathcal{H} = (S, U, \Sigma, \Delta)\). The distance between state signals \(\sigma_1^s, \sigma_2^s\), denoted as \(d^s(\sigma_1^s, \sigma_2^s)\), is defined to be the distance between their graphs, that is, \(d^s(\text{gr}(\sigma_1^s), \text{gr}(\sigma_2^s))\), and the distance between input signals \(\sigma_1^u, \sigma_2^u\), denoted as \(d^u(\sigma_1^u, \sigma_2^u)\), is defined as \(d^u(\text{gr}(\sigma_1^u), \text{gr}(\sigma_2^u))\).

Distance between executions as defined above, called graphical distance, captures the notion that two executions are close if their states are close at approximately same times. The notion of graphical distance is borrowed from [12], where it has been argued that allowing a wiggle time is necessary when one considers hybrid executions. Graphical distance between two executions is illustrated in Figure 1. Note that the two executions \(\sigma\) and \(\sigma'\) are not close at all times, for example, at a time \(t \in (t_1, t_2)\), the states are very far. However, for every time \(t\) and corresponding state \(s\) of \(\sigma\), there exists a time \(t' \in [t - c, t + c]\) such that \(s\) is close to the state of \(\sigma'\) at time \(t'\). For example, \(s_2\) is close to \(s'_2\) and times \(t_1\) and \(t_2\) are close.
In order to define convergence, we need the distance between suffixes of signals starting from some time $T$. Given a subset $G$ of $\mathbb{R}^+ \times \mathbb{N} \times A$ and a $T \in \mathbb{R}^+$, let us denote by $G_T$ the set $\{(t, i, x) \in G | t \geq T\}$. Given two signals $\sigma_1, \sigma_2$ and a $T \in \mathbb{R}^+$, we define $d(\sigma_1 | T, \sigma_2 | T)$ to be $d(\text{gr}(\sigma_1) | T, \text{gr}(\sigma_2) | T)$.

4. AN ALTERNATE CHARACTERIZATION OF INCREMENTAL INPUT-TO-STATE STABILITY OF CONTINUOUS DYNAMICAL SYSTEMS

Our definition of input-to-state stability is motivated by the following definition of incremental input-to-state stability of [2]. Let $T$ be a set of input-state trajectories over $(\mathbb{R}^m, \mathbb{R}^n)$ such that for each $\zeta \in \mathbb{R}^m$ and input trajectory $u$, there exists a unique element $(s, u) \in T$ with First$(s) = \zeta$. Given $\zeta$ and $u$, let us denote the unique trajectory $s$ by $x(\zeta, u)$. Then the definition of incremental input-to-state stability from [2] is as follows:

Definition (\delta ISS for input-state trajectories) The set of input-state trajectories $T$ is said to be incrementally input-to-state stable if there exists a $KL$ function $\beta$ and a $K_\infty$ function $\gamma$ such that for any $t \geq 0$, any $\zeta_1, \zeta_2$ and any pair of input trajectories $u_1, u_2$, the following is true:

$$|x(\zeta_1, u_1)(t) - x(\zeta_2, u_2)(t)| \leq \beta(|\zeta_1 - \zeta_2|, t) + \gamma(|u_1 - u_2|_\infty).$$

(1)

The above definition forces the following properties of the system $T$:

(C1) The system is Lyapunov stable “uniformly” in the input. For every $\epsilon > 0$, there exists a $\delta > 0$, such that for every input trajectory $u$, and for all initial states $\zeta_1, \zeta_2$, the following holds for every $t \geq 0$:

$$|\zeta_1 - \zeta_2| < \delta \Rightarrow |x(\zeta_1, u)(t) - x(\zeta_2, u)(t)| < \epsilon.$$

Note that $\delta$ depends only on $\epsilon$, in particular, it is independent of the input trajectory $u$.

(C2) The system converges “uniformly” in the input. For any $\epsilon > 0$ and initial states $\zeta_1, \zeta_2$, there exists a $T \geq 0$, such that for every input signal $u$,

$$|x(\zeta_1, u)(t) - x(\zeta_2, u)(t)| < \epsilon, \forall t \geq T.$$

Note that $T$ depends only on $\epsilon$ and is independent of $u$.

(C3) The system is input-to-state stable “uniformly” in the initial state. For any $\epsilon > 0$ and input signals $u_1, u_2$, there exists a $\delta > 0$ such that for every initial state $\zeta$, the following holds for every $t \geq 0$:

$$|u_1 - u_2|_\infty < \delta \Rightarrow |x(\zeta, u_1)(t) - x(\zeta, u_2)(t)| < \epsilon.$$

Note the independence of $\delta$ with respect to $\zeta$.

4.1 Super-position type theorem for incremental input-to-state stability

We show that the conditions (C1) – (C3) characterize incremental input-to-state stability as given in Definition 4. This is summarized in the following theorem:

Theorem 1. A set of input-state trajectories $T$ is $\delta$ ISS if it satisfies Conditions (C1) – (C3).

Proof. $\delta$ ISS $\Rightarrow$ (C1) – (C3): It is straightforward to check that $\delta$ ISS implies conditions (C1) – (C3). In fact, choosing $u_1 = u_2 = u$ and $\delta$ such that $\beta(\delta, 0) < \epsilon$ in (1) implies

$$|x(\zeta_1, u)(t) - x(\zeta_2, u)(t)| \leq \beta(|\zeta_1 - \zeta_2|, t) < \beta(\delta, 0) < \epsilon,$$

provided that $|\zeta_1 - \zeta_2| < \delta$. This shows (C1) is true. Moreover, since

$$|x(\zeta_1, u)(t) - x(\zeta_2, u)(t)| \leq \beta(|\zeta_1 - \zeta_2|, t) \to 0$$

as $t \to \infty$, for any given $\epsilon$ and $\zeta_1, \zeta_2$, we can choose $T$ independent of $u$ such that

$$|x(\zeta_1, u)(t) - x(\zeta_2, u)(t)| < \epsilon$$

for all $t > T$. This shows (C2) is true. Finally, choosing $\zeta_1 = \zeta_2 = \zeta$ and $\delta$ such that $\gamma(\delta) < \epsilon$ in (1) implies

$$|x(\zeta, u)(t) - x(\zeta, u)(t)| \leq \gamma(|u_1 - u_2|_\infty) < \gamma(\delta) < \epsilon,$$

provided that $|u_1 - u_2|_\infty < \delta$. This shows (C3) is true. 

(C1) – (C3) $\Rightarrow$ $\delta$ ISS: The proof for the opposite implication essentially follows from the proof of Lemma 4.5 in [16]. Therefore, the detailed argument is omitted and the following is an outline of the proof. First, by (C3), we can prove there exists a $K_\infty$ function $\gamma$ such that

$$|x(\zeta_1, u_1)(t) - x(\zeta_2, u_2)(t)| \leq \gamma(|u_1 - u_2|_\infty),$$

(2)

holds for all initial states $\zeta_1, \zeta_2$. Second, conditions (C1) and (C2) imply that there exist a $KL$ function $\beta$ such that

$$|x(\zeta_1, u_1)(t) - x(\zeta_2, u_2)(t)| \leq \beta(|\zeta_1 - \zeta_2|, t),$$

(3)

holds for all input trajectory $u$. Now given any pair of initial states $\zeta_1, \zeta_2$ and any pair of input trajectories $u_1, u_2$, it follows from (2) and (3) that

$$|x(\zeta_1, u_1)(t) - x(\zeta_2, u_2)(t)| \leq |x(\zeta_1, u_1)(t) - x(\zeta_2, u_1)(t) + x(\zeta_2, u_1)(t) - x(\zeta_2, u_2)(t)| \leq \beta(|\zeta_1 - \zeta_2|, t) + \gamma(|u_1 - u_2|_\infty),$$

(4)

holds for all initial states $\zeta_1, \zeta_2$ and any pair of input trajectories $u_1, u_2$. This completes the proof.

5. INCREMENTAL INPUT-TO-STATE STABILITY OF HYBRID INPUT TRANSITION SYSTEMS

In this section, we define a notion of incremental input-to-state stability of hybrid input transition systems by generalizing the characterization of incremental input-to-state
stability for continuous dynamical systems given by Theorem 1. To define distances between executions, we borrow the notion of graphical distance introduced in [12] for hybrid trajectories. However, the results in the paper are not sensitive to the particular definition of distance.

We need a few definitions. We denote by \( \text{Valid}(T) \), the set of all pairs of input signals and initial states such that there exists a state signal corresponding to it in \( T \). More precisely, \( \text{Valid}(T) = \{ (\sigma^a, \zeta) \mid \exists \sigma^a, \text{First}(\sigma^a) = \zeta, (\sigma^a, \sigma^\ell) \in \mathcal{T} \} \). We denote by \( \text{InSig}(T) \), the set of all input signals such that there exists a state signal corresponding to it in \( T \). That is, \( \text{InSig}(T) = \{ \sigma^a \mid \exists \sigma^a, (\sigma^a, \sigma^\ell) \in \mathcal{T} \} \).

Next, we define incremental input-to-state stability (\( \delta\text{ISS} \)) for hybrid input transition systems.

**Definition.** \( \delta\text{ISS} \) for Hybrid Systems Given a hybrid input transition system \( H \) and a set of executions \( T \subseteq \text{Exec}(H) \), we say that \( H \) is **incrementally input-to-state stable (\( \delta\text{ISS} \))** with respect to the set of executions \( T \), if the following hold:

\[
\quad (D1) \quad \text{for every } \epsilon > 0, \text{ there exists a } \delta > 0, \text{ such that the following holds for every input signal } \sigma^a:\n\]

\[
\forall (\sigma^a, \sigma^\ell) \in \text{Exec}(H), \; d^a(\text{First}(\sigma^a), \text{First}(T|_{\sigma^a})) < \delta \Rightarrow \exists (\sigma^a, \hat{\sigma}^a) \in \mathcal{T}, \; d^a(\sigma^a, \hat{\sigma}^a) < \epsilon
\]

\[
(D2) \text{there exists a } \delta > 0 \text{ and a function } \delta : \mathbb{R} \to \mathbb{R} \text{ such that the following holds for every input signal } \sigma^u:\n\]

\[
\forall (\sigma^u, \sigma^\ell) \in \text{Exec}(H), \; d^u(\text{First}(\sigma^u), \text{First}(T|_{\sigma^u})) < \delta \Rightarrow \exists (\sigma^u, \hat{\sigma}^u) \in \mathcal{T}, \forall \epsilon > 0, \forall t > T(\epsilon), \; d^u(\sigma^u|_{t}, \hat{\sigma}^u|_{t}) < \epsilon.
\]

\[
(D3) \text{for every } \epsilon > 0, \text{ there exists a } \delta > 0 \text{ such that for every input signal } \sigma^u \text{ and state } \zeta \text{ with } (\sigma^u, \zeta) \in \text{Valid}(T), \text{ the following holds:}\n\]

\[
\forall \hat{\sigma}^u, [d^u(\sigma^u, \hat{\sigma}^u) < \delta \Rightarrow \forall (\sigma^u, \sigma^\ell) \in \text{Exec}(H),\n\]

\[
\quad \text{First}(\hat{\sigma}^u) = \zeta \Rightarrow \exists (\sigma^u, \sigma^\ell) \in \mathcal{T}, \quad d^u(\sigma^u|_{t}, \sigma^\ell|_{t}) < \epsilon\]

Remark There have been several proposals for defining metrics on the set of executions, such as the Skorokhod metric (see [9, 5] for more details), wherein, two executions are considered close if there exists a bijective, strictly order-preserving function between the time domains of the executions, such that the distance between a time point and its image under the function is small and the values of the corresponding states are small. However, the results in the paper are not sensitive to the particular choice of the distance metric.

6. PRE-ORDERS

In this section, we define a pre-order - a reflexive, transitive relation - on the class of hybrid input transition systems, such that if a system is incrementally input-to-state stable, then all systems below it in the ordering are incrementally input-to-state stable. Defining such an ordering is the first step towards developing an abstraction refinement framework for verifying incremental input-to-state stability. We begin by defining the canonical notion of equivalence between systems with input, namely, input bisimulations. We show that this canonical notion does not suffice to preserve \( \delta\text{ISS} \) and hence strengthen it with additional conditions. In the next section, we show that the new notion preserves \( \delta\text{ISS} \).

6.1 Input (Bi)-Simulations

The notion of input (bi)-simulation is an extension of the classical notion of (bi)-simulation with inputs for hybrid input transition systems. Our definition is closely related to the definition of (bi)-simulation defined in [15].

**Definition.** Given two hybrid input transition systems \( H_1 = (S_1, U_1, \Sigma_1, \Delta_1) \) and \( H_2 = (S_2, U_2, \Sigma_2, \Delta_2) \), a pair of binary relations \( (R_1, R_2) \), where \( R_1 \subseteq S_1 \times S_2 \) and \( R_2 \subseteq U_1 \times U_2 \), is called an *input simulation relation* from \( H_1 \) to \( H_2 \) if, for every \( (s_1, s_2) \in R_1 \), the following hold:

- For every state \( s_1 \) and input \( u_1 \) such that \( (u_1, (s_1, s_1)) \in \Sigma_1 \), there exist a state \( s_2 \) and an input \( u_2 \) such that \( R_1(s_1, s_2), R_2(u_1, u_2) \) and \( (u_2, (s_2, s_2)) \in \Sigma_2 \).

- For every input-state trajectory \( (u_1, s_1) \in \Delta_1 \) such that \( \text{First}(s_1) = s_1 \), there exists an input-state trajectory \( (u_2, s_2) \in \Delta_2 \) such that \( \text{First}(s_2) = s_2 \) and \( s_2 \in R_1(s_1) \) and \( u_2 \in R_2(u_1) \).

We denote the fact that \( (R_1, R_2) \) is an input simulation relation from \( H_1 \) to \( H_2 \) by \( H_1 \preceq_{\text{ISS}} (R_1, R_2) H_2 \). Further, \( (R_1, R_2) \) is an *input bisimulation relation* between \( H_1 \) and \( H_2 \) if both \( (R_1, R_2) \) and \( (R_2^{-1}, R_2^{-1}) \) are input simulation relations, that is, \( H_1 \preceq_{\text{ISS}} (R_1, R_2) H_2 \) and \( H_2 \preceq_{\text{ISS}} (R_2^{-1}, R_2^{-1}) H_1 \).

**Image of \( H \) under \( (R_1, R_2) \).**

We define the image of a hybrid input transition system on a pair of relations \( (R_1, R_2) \) as \( (S, \Sigma, \Delta) \). Given a hybrid input transition system \( H = (S, U, \Sigma, \Delta) \), and a pair \( (R_1, R_2) \), where \( R_1 \subseteq S \times S' \) and \( R_2 \subseteq U \times U' \), for some \( S' \) and \( U' \), we define \( (R_1, R_2)(H) \) to be the hybrid input transition system \( (S', U', \Sigma', \Delta') \), where:

- \( \Sigma' = \{ (u', (s_1', s_2')) | \exists (u, (s_1, s_2)) \in \Sigma, u' \in R_2(u), s_1' \in R_1(s_1), s_2' \in R_1(s_2) \} \),

- \( \Delta' = \{ (u', s') | \exists (u, s) \in \Delta, u' \in R_2(u), s' \in R_1(s) \} \).

**Proposition 1.** Let \( H = (S, U, \Sigma, \Delta) \) be a hybrid input transition system, and \( R_1 \subseteq S \times S' \) and \( R_2 \subseteq U \times U' \), for some \( S' \) and \( U' \) such that \( R_1(s) \) and \( R_2(u) \) is not empty for any \( s \in S \) and \( u \in U \). Then \( (R_1, R_2) \) is an input simulation from \( H \) to \( (R_1, R_2)(H) \).

6.2 Uniformly Continuous Input (Bi)-Simulation

The notion of input bisimulation does not preserve incremental input-to-state stability. Hence, we strengthen it with uniformity conditions.

Let \( (H_1, d^1_1, d^1_2) \) and \( (H_2, d^2_1, d^2_2) \) be two metric input hybrid transition systems.

**Definition.** A pair \( (R_1, R_2) \) is a **uniformly continuous input simulation** from \( H_1 \) to \( H_2 \) if \( (R_1, R_2) \) is an input simulation from \( H_1 \) to \( H_2 \) and \( R_1, R_1^{-1}, R_2, R_2^{-1} \) are uniformly continuous.

We denote the fact that \( (R_1, R_2) \) is a uniformly continuous input simulation from \( H_1 \) to \( H_2 \) by \( H_1 \preceq^{\text{ISS}}_{(R_1, R_2)} H_2 \), and
\( \mathcal{H}_1 \preceq^C \mathcal{H}_2 \) to denote that there exists \((R_1, R_2)\) such that \( \mathcal{H}_1 \preceq_{(R_1, R_2)}^C \mathcal{H}_2 \).

Definition. A pair \((R_1, R_2)\) is a uniformly continuous input bisimulation from \( \mathcal{H}_1 \) to \( \mathcal{H}_2 \) if \((R_1, R_2)\) is a uniformly continuous input simulation from \( \mathcal{H}_1 \) to \( \mathcal{H}_2 \), and \((R_1^{-1}, R_2^{-1})\) is a uniformly continuous input simulation from \( \mathcal{H}_2 \) to \( \mathcal{H}_1 \).

Next, we show that uniformly continuous input simulations define a pre-order on hybrid input transition systems.

Theorem 2. Let \((\mathcal{H}_i, d_i^+ \bigcup d_i^-)\), for \( 1 \leq i \leq 3 \), where \( \mathcal{H}_i = (S_i, U_i, \Sigma_i, \Delta_i) \), be three metric hybrid transition systems. Then we have the following properties about \( \preceq^C \):

- (Reflexivity) \( \mathcal{H}_1 \preceq^C \mathcal{H}_1 \).
- (Transitivity) If \( \mathcal{H}_1 \preceq^C \mathcal{H}_2 \) and \( \mathcal{H}_2 \preceq^C \mathcal{H}_3 \), then \( \mathcal{H}_1 \preceq^C \mathcal{H}_3 \).

Proof. (Reflexivity) Reflexivity follows from the fact that \( \mathcal{H}_1 \preceq_{(Id_1, Id_2)} (\mathcal{H}_1, d_1^+ \bigcup d_1^-) \mathcal{H}_2 \), where \( Id_1 = \{(s, s) \mid s \in S\} \) and \( Id_2 = \{(u, u) \mid u \in U\} \).

(Transitivity) Transitivity follows from the fact that \( \mathcal{H}_1 \preceq_{(R_1, R_2)} (\mathcal{H}_1, d_1^+ \bigcup d_1^-) \mathcal{H}_2 \) and \( \mathcal{H}_2 \preceq_{(R_2', R_2)} (\mathcal{H}_2, d_2^+ \bigcup d_2^-) \mathcal{H}_3 \), then \( \mathcal{H}_1 \preceq_{(R_1 \circ R_2, R_2 \circ R_2')} (\mathcal{H}_1, d_1^+ \bigcup d_1^-) \mathcal{H}_3 \), where \( A \circ B = \{(x, z) \mid \exists (x, y) \in A, (y, z) \in B\} \), since composition of input bisimulations is an input bisimulation, and composition of uniformly continuous relations is a uniformly continuous relation.

6.3 Inadequacy of Weaker Notions of Input Bisimulations

We show that weaker extensions to the definition of input bisimulation which require only continuity instead of uniform continuity on either the input space or the state space, do not suffice to preserve incremental input-to-state stability.

6.3.1 Necessity of Uniform Continuity on the Input Space

We consider two systems \( \mathcal{H}_1 \) and \( \mathcal{H}_2 \) consisting only of trajectories such that \( \mathcal{H}_1 \) is ISS, whereas \( \mathcal{H}_2 \) is not. We then show that there exists a input bisimulation, which is uniformly continuous on the state space, but is only continuous on the input space.

The state-space of both the systems is \( \mathbb{R}^2 \), and the input space is \( \mathbb{R} \). All the input signals are constant signals with values in \( \mathbb{R} \). The set of initial states is \( \{0\} \times [-1, 1] \). We will use variable \( x \) and \( y \) to denote the two dimensions. Hence, \( \Phi((x, y), t, u) \) denotes the values of the trajectory starting from \((x_0, y_0)\) with input \( u \) at time \( t \). The reference trajectories, \( T_1 \) and \( T_2 \), are those corresponding to the initial state \((0, 0)\). Since our input signals are constant signals, we abuse notation and just use the constant value instead of the signal in \( \Phi \).

The dynamics of the first system \( \mathcal{H}_1 \) is given as follows:

\[
\Phi((x_0, y_0), t, u) = (t, e^{-t}y_0 + u),
\]

where \( \bar{t} = t \) if \( t \leq 1 \), and 1 otherwise.

The dynamics of the second system \( \mathcal{H}_2 \), is similar to \( \mathcal{H}_1 \), except that the affect of the input at time \( t = 1 \) is \( e^u \) instead of \( u \) as in the previous system.

\[
\Phi((x_0, y_0), t, u) = (t, e^{-t}y_0 + e^u),
\]

where \( \bar{t} = t \) if \( t \leq 1 \), and 1 otherwise.

Note that both the systems satisfy Conditions C1 and C2, where as, only the first system satisfies C3. Hence, the system \( \mathcal{H}_1 \) is ISS, whereas as \( \mathcal{H}_2 \) is not.

We can define an input bisimulation between the two systems given by:

\[
R_1 = \{(x, y, (x, y)) \mid x = x_1, y_1 = e^{-x_1}y_2\}, R_2 = \{(u, u)\}.
\]

This is an input bisimulation, since the trajectories starting from the initial state \((x_0, y_0)\) and input signal \( e^u \) in \( \mathcal{H}_1 \) is related by \( R_1 \) to the trajectory starting from the initial state \((x_0, y_0)\) and the input signal \( u \) in \( \mathcal{H}_2 \). Further, \( R_1 \) and \( R_2^{-1} \) are uniformly continuous, whereas as, \( R_2^{-1} \) is only continuous.

6.3.2 Necessity of Uniform Continuity on the State Space

We extend the counter-example in [20], which shows the necessity of uniform continuity on the state space for asymptotic stability preservation (without inputs), for showing the necessity of uniform continuity on the state-space for ISS preservation.

The first system \( \mathcal{H}_3 \) is the same as \( \mathcal{H}_1 \) above, except that the input space is restricted to be \( \{0\} \). We define \( \mathcal{H}_4 \) which is the same as \( \mathcal{H}_1 \) except that the trajectories do not converge to the reference trajectory. The dynamics of the system \( \mathcal{H}_4 \) is given as follows:

\[
\Phi((x_0, y_0), t, u) = (t, y_0).
\]

Note that both \( \mathcal{H}_3 \) and \( \mathcal{H}_4 \) satisfy Conditions C1 and C3, where as, only \( \mathcal{H}_3 \) satisfies C2. Hence, \( \mathcal{H}_3 \) is ISS, whereas as, \( \mathcal{H}_4 \) is not.

We can define an input bisimulation between the two systems given by:

\[
R_1 = \{(x_1, y_1, (x_2, y_2)) \mid x_1 = x_2, y_1 = e^{-x_1}y_2\}, R_2 = \{(u, u)\}.
\]

This is an input bisimulation, since the trajectories starting from the initial state \((x_0, y_0)\) and input signal \( u \) in \( \mathcal{H}_2 \) is related by \( R_1 \) to the trajectory starting from the initial state \((x_0, y_0)\) and the input signal \( u \) in \( \mathcal{H}_2 \). Further, \( R_2 \) and \( R_2^{-1} \) are uniformly continuous, whereas as, \( R_1 \) is only continuous.

7. Incremental Input-to-State Stability Preservation

In this section, we present the main result of the paper, namely, that incremental input-to-state stability is invariant under uniformly continuous input bisimulations.

We need a technical consistency condition between the input bisimulation relations and the reference executions.

Definition. A pair of relations \((T_1, T_2)\), where \( T_1 \subseteq S_1 \times S_2 \) and \( T_2 \subseteq U_1 \times U_2 \), is said to be semi-consistent with respect to the sets of executions \( T_1 \) and \( T_2 \) over \((S_1, U_1)\) and \((S_2, U_2)\), respectively, if the following hold:

(A1) For every \((\sigma_1, \zeta_1) \in Valid(T_1)\), there exists \((\sigma_2, \zeta_2) \in Valid(T_2)\) such that \( R_2(\sigma_1, \sigma_2^*) \) and \( R_1(\zeta_1, \zeta_2) \).

(A2) For every \((\sigma_2^*, \sigma_2^*) \in T_2\), for every \( \sigma_1^* \in R_2^{-1}(\sigma_2^*) \) and \( \zeta_1 \in R_1^{-1}(\text{First}(\sigma_2^*)) \) such that \( (\sigma_1^*, \zeta_1) \in Valid(T_1) \), there exists \( \sigma_1^* \) with \( \text{First}(\sigma_1^*) = \zeta_1 \), \( R_1(\sigma_1^*, \sigma_1^*) \), and \( (\sigma_1^*, \sigma_1^*) \in T_1 \).

(A3) \( R_2(u) \) is a singleton for every \( u \in Inputs(T_1) \).

(A4) \( R_1^{-1}(s) \) is singleton for every \( s \in States(T_2) \).
(A5) For every $\sigma^u_1$, $R_1(\text{First}(T_1|_{x^u_1})) = \text{First}(T_2|_{x^u_2})$.

(A6) There exists $\delta > 0$ such that for every $x \in B(\text{First}(T_1))$, there exists a $y$ such that $R_1(x, y)$.

$(R_1, R_2)$ is said to be consistent with respect to $T_1$ and $T_2$ if both $(R_1, R_2)$ and $(R_1 \uparrow, R_2^{\uparrow^{-2}})$ are semi-consistent with respect to $T_1$ and $T_2$.

**Theorem 3.** Let $(H_1, d_1^y, d_1^u)$ and $(H_2, d_2^y, d_2^u)$, where $H_1 = (S_1, U_1, \Sigma_1, \Delta_1)$ and $H_2 = (S_2, U_2, \Sigma_2, \Delta_2)$, be two metric hybrid input transition systems, and let $T_1 \subseteq \text{Exec}(H_1)$ and $T_2 \subseteq \text{Exec}(H_2)$ be two sets of executions. Let $(R_1, R_2)$ be a uniformly continuous input simulation from $H_1$ to $H_2$, and let $(R_1, R_2)$ be semi-consistent with respect to $T_1$ and $T_2$. Then the following holds:

If $H_2$ is $\delta$ISS with respect to $T_2$, then $H_1$ is $\delta$ISS with respect to $T_1$.

**Proof.** Let us assume $H_2$ is $\delta$ISS with respect to $T_2$. We need to show that $H_1$ is $\delta$ISS with respect to $T_1$. We will show that $H_1$ satisfies conditions (D1) – (D3).

**Proof of satisfaction of Condition (D1)** Let us fix an $\epsilon_1 > 0$. We need to find a $\delta_1 > 0$ such that Condition (D1) holds in $H_1$ and $T_1$. Let $\epsilon_2$ be the uniformity constant of $R_1 \uparrow$ corresponding to $\epsilon_1$. Let $\delta_2$ be the constant satisfying Condition (D1) for $H_2$ corresponding to $\epsilon_2$. Set $\delta_1$ to be the uniformity constant of $R_2$ corresponding to $\delta_2$.

![Figure 2: Illustration for Proof of Condition (D1)](image)

Let us fix an input signal $\sigma^u_1$. Let $(\sigma_1^u, \sigma_1^s) \in \text{Exec}(H_1)$ such that $d_1^y(\text{First}(\sigma_1^u), \text{First}(T_1|_{x_1^u})) < \delta_1$ (see Figure 2). We need to show that there exists a $\sigma_1^s$, such that $(\sigma_1^u, \sigma_1^s) \in T_1$ and $d_1^u(\sigma_1^u, \sigma_2^u) < \epsilon_1$.

Note that Condition (A1) also implies that there exists $\sigma_2^u \in \text{InSig}(T_2)$ such that $R_2(\sigma_2^u, \sigma_2^s)$. Further, $\sigma_2^u$ is unique because of Condition (A3) on $R_2$. From Condition (A6), there exists a $\zeta_1$ such that $(\text{First}(\sigma_1^u), \zeta_1) \in R_1$. Therefore, from input simulation relation, there exists $\sigma_2^u$ such that $(\sigma_2^u, \sigma_2^s) \in \text{Exec}(H_2)$ and $R_2(\sigma_2^u, \sigma_2^s)$ (note that $\sigma_2^u$ is the same as before, this follows from the uniqueness of $\sigma_2^u$).

![Figure 3: Illustration for Proof of Condition (D3)](image)

Theorem 2: For every $s$, $\exists R_1(\text{First}(T_1|_{x_1^u})) = \text{First}(T_2|_{x_2^u})$.

Let us assume $H_1$ is $\delta$ISS with respect to $T_1$. Then the following holds:

If $H_2$ is $\delta$ISS with respect to $T_2$, then $H_1$ is $\delta$ISS with respect to $T_1$.

Let us assume $H_2$ is $\delta$ISS with respect to $T_2$. We need to show that $H_1$ is $\delta$ISS with respect to $T_1$. We will show that $H_1$ satisfies conditions (D1) – (D3).

**Proof of satisfaction of Condition (D1)** Let us fix an $\epsilon_1 > 0$. We need to find a $\delta_1 > 0$ such that Condition (D1) holds in $H_1$ and $T_1$. Let $\epsilon_2$ be the uniformity constant of $R_1 \uparrow$ corresponding to $\epsilon_1$. Let $\delta_2$ be the constant satisfying Condition (D1) for $H_2$ corresponding to $\epsilon_2$. Set $\delta_1$ to be the uniformity constant of $R_2$ corresponding to $\delta_2$.
hybrid input transition systems, and \( T_1 \subseteq \text{Exec}(H_1) \) and \( T_2 \subseteq \text{Exec}(H_2) \) be two sets of executions. Let \( (R_1, R_2) \) be a uniformly continuous input bisimulation from \( H_1 \) to \( H_2 \), and let \( (R_1, R_2) \) be consistent with respect to \( T_1 \) and \( T_2 \). Then the following holds:

\[ H_2 \text{ is } \delta\text{ISS with respect to } T_2 \text{ if and only if } H_1 \text{ is } \delta\text{ISS with respect to } T_1. \]

Theorem 3 states that uniformly continuous input simulations serve as a notion of abstraction with respect to \( \delta\text{ISS and Theorem 4 states that uniformly continuous input bisimulations are a notion of equivalence between systems with respect to incremental input-to-state-stability.} \]

8. EXAMPLES OF CONCRETE ABSTRACTION FUNCTIONS

In this section, we argue that the notion of abstraction introduced in the paper is not too stringent by exhibiting concrete abstraction functions which satisfy the constraints imposed by the uniformly continuous input simulations/bisimulations. We show that the proofs of stability based on Lyapunov functions can be interpreted as constructing simpler hybrid input transition systems where the Lyapunov function serves as a uniformly continuous input simulation relation. We further illustrate this through a concrete example of a linear system with inputs. The main purpose of this section is to demonstrate that the definitions of uniformly continuous input simulations and bisimulations are reasonable.

8.1 Lyapunov Functions for Input-to-State Stability

In this section, we focus on systems in which the reference executions consist of the unique trajectory which always remains at the equilibrium point \( \theta \). In this case, the definition of incremental input-to-state stability coincides with the notion of input-to-state stability. First, we define a Lyapunov’s theorem for analyzing input-to-state stability.

Consider a continuous dynamical system

\[ \dot{x} = f(x, u), \]

where \( f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \) is locally Lipschitz in \( x \) and \( u \), and \( X_0 \) and \( U \) are compact sets. We will assume that the input signal space \( D_u \) consists of functions \( u : [0, \infty) \to U \) that are piecewise continuous, bounded functions of \( t \) for all \( t \geq 0 \).

We define the hybrid system corresponding to the System (4) to be the following: \( H_{f,x_0,u} = (X, U, \emptyset, \Delta) \), where \( \Delta \) is the set of pairs \((u, x)\), where \( u \) is in \( D_u \), \( x(0) \in X_0 \) and \( x \) is the solution to the System (4) starting from \( x(0) \), that is, \( u, x \) satisfy \( \dot{x}(t) = f(x(t), u(t)) \) for every \( t \geq 0 \). Let \( d^u \) and \( d^n \) be the standard Euclidean norms on \( \mathbb{R}^n \) and \( \mathbb{R}^m \), respectively.

For System (4), assume that the system \( \dot{y} = f(y, 0) \) has a uniformly asymptotically stable equilibrium point at the origin.

Definition. A continuously differentiable function \( V : X \to \mathbb{R}^+ \) is said to be an ISS Lyapunov function for the System (4) if there exist class \( K_\infty \) functions \( \alpha_1, \alpha_2, \alpha_3 \) and \( \chi \) such that:

\[ \alpha_1(|x|) \leq V(x(t)) \leq \alpha_2(|x|), \forall x \in X, t > 0 \]

\[ \frac{\partial V(x)}{\partial x} f(x, u) \leq \alpha_3(|x|), \forall u \in D_u : |x| \geq \chi(|u|). \]

Theorem 5. [22] (ISS Theorem) Let \( V : X \to \mathbb{R}^+ \) be an ISS Lyapunov function for the System (4). Then System (4) is input-to-state stable.

Theorem 6. Let \( V \) be an ISS Lyapunov function for System (4), and let \( N : \mathbb{R}^n \to \mathbb{R}^+ \) be the function \( u \mapsto |u| \). Let \( V \) have non-zero differential if there exists a neighborhood \( Y \) containing 0 such that the gradient of \( F \) at any point \( y \in Y \) other than 0, \( \nabla F(y) \), is non-zero. Following theorem formulates Lyapunov analysis in our framework:

8.2 Illustration on linear system

We illustrate Lyapunov function based analysis as an abstraction based analysis on a linear system example. Consider a linear system with inputs, that is,

\[ \dot{x} = f(x, u) = Ax + Bu, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, \]

where, \( A \) is a Hurwitz matrix, and \( X_0 \) and \( U \) are compact sets.

Let \( P \) be a positive definite symmetric matrix satisfying \( A^TP + PA = -Q \) for some positive definite matrix \( Q \). Consider a function \( R_1 : \mathbb{R}^n \to \mathbb{R}^+ \) given by \( R_1(x) = x^TPx \) and a function \( R_2 : \mathbb{R}^m \to \mathbb{R}^+ \) given by \( R_2(u) = |u| \). Then,

\[ R_1(x) = \dot{x}^TPx + x^TPx = x^T(A^TP + PA)x + \dot{x}^TPx + x^TPBu \leq -\lambda R_1(x) + \mu |u| \infty, \]

where \( \lambda \) and \( \mu \) are positive constants depending on \( P, Q \) and \( B \).
Consider the one-dimensional system:

$$\dot{y} \leq -\lambda y + \mu \|v\|_\infty, \ y \geq 0.$$  \hfill (8)

Note that the solutions to the system satisfy

$$y(t) \leq e^{-\lambda t} y(0) + \frac{\mu}{\lambda} \|v\|_\infty.$$  

It is easy to check from the solutions of this system, that it is trivially incrementally input-to-state stable.

We will show that $(R_1, R_2)$ is a uniformly continuous input simulation from System (7) to System (8). Input simulation follows from the fact that if $(x, u)$ satisfies $\dot{x}(t) = Ax(t) + Bu(t)$ for all $t \geq 0$, then by construction, $(R_1(x), R_2(u))$ satisfies $R_1(x) \leq -\lambda R_1(x) + \mu R_2(u)$ for all $t \geq 0$. Also, when $R_1$ and $R_2$ are interpreted as relations or set valued functions, then $R_1$, $R_2$ are continuous over $\text{States}(\mathcal{H}, x_0, x, u)$ and $\text{Inputs}(\mathcal{H}, x_0, x, u)$. Note that the set of reference executions in both the systems is $\{0, 0\}$, where $0$ is of appropriate dimension. It is easy to see that semi-consistency is trivially satisfied. Hence, from Theorem 3 System (7) is incremental input-to-state stable.

9. CONCLUSIONS

In this paper, we investigated pre-orders for reasoning about incremental input-to-state stability properties. We introduced the notion of uniformly continuous input simulations and bisimulations as pre-orders which preserve incremental input-to-state stability of systems. We showed that the notion is a reasonable pre-order to consider by establishing Lyapunov function based analysis of incremental input-to-state stability as a special case of our analysis framework.

In the future, we intend to develop concrete techniques for constructing abstractions based on uniformly continuous input simulations and bisimulations. The notion of refinement as given by the pre-order is a new concept which does not arise in the Lyapunov function based analysis. The notion of refinements (an abstraction ordered between the concrete and the abstract system) will be the guiding principle for developing concrete refinement techniques. Our broad goal is to develop an abstraction refinement technique for analysis of stability properties.

10. ACKNOWLEDGEMENTS

The authors would like to thank the anonymous referees for their feedback in improving the earlier versions of the paper. This work was partially supported by a postdoctoral fellowship from the Center for Mathematics of Information at the California Institute of Technology.

11. REFERENCES


