Approximated Stability Analysis of Bi-Modal Hybrid Co-simulation Scenarios*

Cláudio Gomes[1], Paschalis Karalis[2], Eva M. Navarro-López[2], and Hans Vangheluwe[134]

[1] Department of Computer Science and Mathematics, University of Antwerp
{claudio.gomes}{hans.vangheluwe}@uantwerp.be
[2] School of Computer Science, The University of Manchester
{paschalis.karalis}{eva.navarro}@manchester.ac.uk
[3] McGill University
[4] Flanders Make

Abstract. Co-simulation is a technique to orchestrate multiple simulators in order to approximate the behavior of a coupled system as a whole. Simulators execute in a lockstep fashion, each exchanging inputs and output data points with the other simulators at pre-accorded times. In the context of systems with a physical and a cyber part, the communication frequency with which the simulators of each part communicate can have a negative impact in the accuracy of the global simulation results. In fact, the computed behavior can be qualitatively different, compared to the actual behavior of the original system, laying waste to potentially many hours of computation. It is therefore important to develop methods that answer whether a given communication frequency guarantees trustworthy co-simulation results.
In this paper, we take a small step in that direction. We develop a technique to approximate the lowest frequency for which a particular set of simulation tools can exchange values in a co-simulation and obtain results that can be trusted.

Keywords: hybrid co-simulation, hybrid systems, Lyapunov stability analysis, coupled simulation, hybrid automata

1 Introduction

As complexity in systems grows, and market pressure increases, system development is made by increasingly specialized teams, with tools tailored for each domain. Each team develops a part of the system, which is integrated with the remaining parts. Paradoxically, attaining innovative and multidisciplinary solutions requires that the development process is more integrated [24,22].

* This work has been done under the framework of the COST Action IC1404 – Multi-Paradigm Modelling for Cyber-Physical Systems (MPM4CPS), and partially supported by Flanders Make vzw, the strategic research centre for the manufacturing industry, and PhD fellowship grants from the Agency for Innovation by Science and Technology in Flanders (IWT, dossier 151067).

Modeling and simulation techniques improve the development of each part of the system (see, e.g., [8]), but face challenges when applied in holistic development processes. Co-simulation is a technique to overcome some of those challenges [25].

Co-simulation — the integration of multiple simulation tools, each specialized in the simulation of a particular kind of models, for the purpose of computing the global time behavior of the system — can be used to study system as a whole, when a single tool cannot. It is often required for systems whose models are best expressed in different formalisms [26].

During a co-simulation, each tool simulates the sub-model that pertains to its domain, and assumes that other tools will simulate the environment of the sub-model [10,21]. Therefore, underlying a particular co-simulation scenario – an assignment from outputs to inputs, of a set of simulation tools – there is an heterogeneous coupled model, which we denote as the original system.

To facilitate tool interaction, the Functional Mockup Interface (FMI) standard [3] was created, that defines an interface based on input and outputs.

An extra advantage of a standardized, modular coupling of tools, is that, in advanced stages of the development process, sub-models can be swapped by real-physical prototypes, interacting seamlessly in a (real-time) co-simulation with other tools – hardware-in-the-loop co-simulation.

Theoretically, any trajectory computed in a co-simulation should be the same as a solution to the original system [10]. In practice, just as with the simulation of systems described by differential equations, this is not the case. For example, it can happen that, while the behavior of the original system remains bounded over time, the trajectories computed by the co-simulation do not. This may be due to a delayed reaction of simulation tools to certain events, caused in turn by the frequency with which tools exchange inputs and outputs. It is therefore of utmost importance that co-simulation orchestration algorithms can tweak the communication frequency of the tools to ensure that system developers can trust the co-simulation results. However, more frequent communication entails a performance toll. Hence, a valid *research question* is: for a particular co-simulation scenario, what is the lowest frequency for which tools can exchange values, that still ensures that the computed trajectories are bounded? The question is not new: it has been studied for traditional simulation.

The *novelty of this paper* is that we reformulate the numerical stability of co-simulation scenarios as the stability of equilibrium points where the underlying original system is a hybrid system of a particular family.

Stability of hybrid systems has been studied extensively (see, for example, [9,14,12,20]). Most of the results define stability in the Lyapunov sense [13] (bounded trajectories, adapted from the continuous smooth case), and can be classified as: (i) the study of stability by using a common energy function for all the subsystems [14], or (ii) the use of multiple Lyapunov functions, one for each subsystem [4,12,19]. The consideration of multiple equilibria is not common in the hybrid systems literature, being typically focused on the study of systems

with a unique equilibrium point for all the subsystems. Among very few results considering multiple equilibria are [19,20].

To the best of our knowledge, there is no work that applies these stability analysis techniques to study the effects of co-simulation in hybrid systems. Stability for co-simulation of original systems described by differential equations has been studied in [5,11,23,1] though. The dwell time approaches (e.g., [17]) can potentially be used, in the sense that they restrict the time that the system spends in each mode, as we do. However, our approach, spanning from the necessity to analyze co-simulation scenarios, does so by controlling the co-simulation step.

We contribute with a method for analyzing the stability of hybrid co-simulation scenarios and an algorithm to approximate the safe range of communication frequency between tools. To this end, we apply the stability results in [4] to a hybrid automaton representation of the co-simulation with multiple equilibria. The modeling of the co-simulation algorithm as a hybrid automaton – in a deterministic and non-deterministic versions – will be also one of the main contributions of our work. We use the hybrid-automaton modeling framework of hybrid systems (see, e.g., [18,16]).

The next section gives a brief introduction to hybrid automata, and describes the family of hybrid systems that our contribution applies to and how their stability can be studied. Sections 3 and 4 describe our contribution. Finally, in Section 5 we discuss the limitations of our approach and opportunities for future work.

2 Hybrid Systems

2.1 Hybrid Automaton Representation

Definition 1. *A hybrid automaton $\mathscr{H}$ is a collection*

$$\mathscr{H} = (Q, E, \mathscr{X}, Dom, \mathscr{F}, Init, G, R)$$

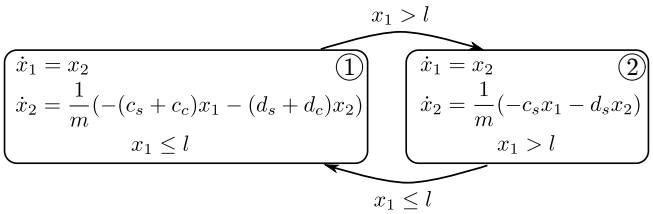
where: • $Q = \{q_1, q_2, \ldots\}$ *is a finite set of modes.* • $E \subseteq Q \times Q$ *is a finite set of edges called transitions.* • $\mathscr{X} \subseteq \mathbb{R}^n$ *is the continuous state space, for some natural* n. • $Dom \subseteq Q \to 2^{\mathscr{X}}$ *is the mode domain.* • $\mathscr{F} = \left\{ f_{q_i}(x) : q_i \in Q \right\}$ *is a collection of time-invariant vector fields such that each* $f_{q_i}(x)$ *is Lipschitz continuous on* $Dom(q_i)$ *in order to ensure that the solution exists in* q_i *and is unique for a given initial state.* • $Init \subseteq Q \times \mathscr{X}$ *is a set of initial states.* • $G : E \to 2^{\mathscr{X}}$ *defines a guard set for each transition.* • $R : E \times \mathscr{X} \to 2^{\mathscr{X}}$ *specifies how the continuous state is reset at each transition.*

Intuitively, at any point in time t, $\mathscr{H}$ is in a mode $q_i \in Q$, with a continuous state $x(t)$. The continuous state evolves according to the Ordinary Differential Equation (ODE) $\dot{x}(t) = f_{q_i}(x(t))$ associated with mode q_i. $\mathscr{H}$ is allowed to stay in mode q_i as long as $x(t) \in Dom(q_i)$ holds. $\mathscr{H}$ *may* switch to mode q_j if $(q_i, q_j) \in E$ and $x(t) \in G((q_i, q_j))$. When such a mode switch happens at time t_s, the continuous state is reset to a new continuous state given by $R((q_i, q_j), x(t_s))$. The new

state will be the initial state for the new mode ODE $\dot{x}(t) = f_{q_j}(x(t))$. Note that, for a given unique initial state, the behavior can still be non-deterministic.

It is common to represent a hybrid automaton as a directed graph with nodes depicting each mode, and edges depicting the transitions. The dynamics associated with each mode are represented inside the respective node, and the guards and reset map of each transition are represented near the edge corresponding to that transition. The guards are represented with conditions and the resets with assignments of the form $x := \ldots$. When the state is not changed at the transition, that is $x := x$, we omit the assignment. We will often eliminate the time when writing the continuous state x for the sake of simplicity in the notation.

Example 1. Consider a system where a cart is connected to a spring/damper and a cord, illustrated in Figure 1. The cord is connected to an actuator that stretches it whenever the cart crosses is to the left of the sensor, and loosens it when the cart is to the right of the sensor. The system is modeled as:



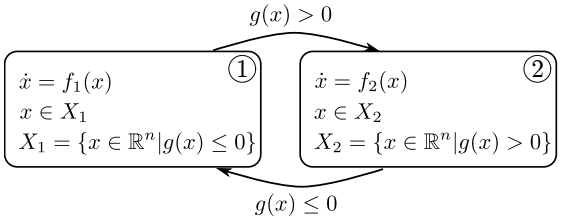
(1)

Mode 1 refers to the cord stretched and mode 2 to the cord loose, and: $l < 0$ is the sensor position, m is the mass of the cart, $x \in \mathbb{R}^2$ is the continuous state (position and velocity) of the cart, $c_s > 0$ is the stiffness coefficient of the spring, $d_s > 0$ the damping coefficient of the damper, and $c_c, d_c > 0$ the analogous coefficients of the cord. Figure 2 shows a trajectory of the cart position, for parameters $l = 10^{-4}, m = 1, c_s = 1, d_s = 0.5, d_c = 10^{-4}, c_c = 10^3$, and initial mode 2 and state $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$.

2.2 Bi-modal Hybrid Automaton

We restrict our study to bi-modal hybrid automata defined as follows.

Definition 2. *A bi-modal hybrid automaton is represented as:*



(2)

with a given initial state $x(t_0) = x_0$. The initial mode is inferred from x_0.

The sets X_1 and X_2 define the invariant set of each mode and the dot denotes the time derivative. Throughout this paper, we make the following assumptions:

1. the switching surface $\mathscr{S} = \{x \in \mathbb{R}^n : \ g(x) = 0\}$ is a smooth hyper surface in $\mathbb{R}^n$;
2. the system has a single equilibrium point $x_{\text{eq}} = \bar{0} \in \mathbb{R}^n$ common to all modes, i.e., $f_{q_i}(x_{\text{eq}}) = \bar{0}$ for all $q_i \in \{1,2\}$, not at the switching surface ($g(x_{\text{eq}}) \neq 0$);
3. the system trajectory does not enter a sliding motion on $\mathscr{S}$;
4. the continuous state is kept the same across mode transitions ($x := x$).

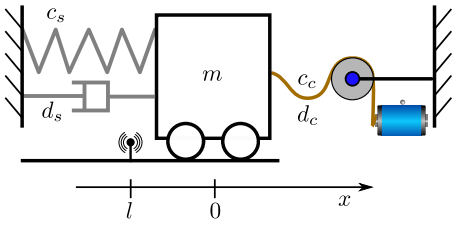


Fig. 1. Cart example.

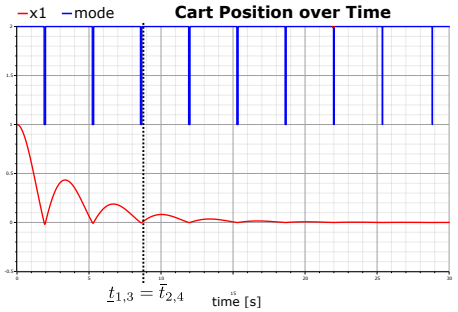


Fig. 2. Cart position over time.

The cart system, defined in 1, is an example of a bi-modal hybrid automaton. It has one equilibrium point, at the origin.

In general, a solution of System (2) has two components: a continuous state evolution $x(t)$ in $\mathbb{R}^n$, and a piecewise constant function of time $\sigma(t) \in \{1,2\}$, called switching sequence. At any time t, $\dot{x}(t)$ is given by $f_{\sigma(t)}(x(t))$.

For mode $q_i \in \{1,2\}$, we denote the sequence of times at which mode q_i is switched on as $\{\bar{t}_{q_i,k}\}$ with $\bar{t}_{q_i,k} \leq \bar{t}_{q_i,k+1}$, and the set at which q_i is switched off as $\{\underline{t}_{q_i,k}\}$ with $\underline{t}_{q_i,k} \leq \underline{t}_{q_i,k+1}$. For example, if the system starts in mode $q_i = 1$, then $\bar{t}_{1,1} = 0 \in \{\bar{t}_{1,k}\}$. Note that $\bar{t}_{q_i,k} \leq \underline{t}_{q_i,k}$ is always the case, so $[\bar{t}_{q_i,k}, \underline{t}_{q_i,k}]$ represents the interval during which the system is in mode q_i for the k-th time. Figure 2 shows an example. To link these definitions with System (2), note that $\forall t \in [\bar{t}_{1,k}, \underline{t}_{1,k}] \implies g(x(t)) \leq 0$.

2.3 Stability

We use the term stability to refer to the fact that any solution to System (2), starting arbitrarily close to $x_{\text{eq}} = \bar{0}$, will remain close to it as time advances [13].

Definition 3. *Given any $\varepsilon > 0$, the equilibrium point $\bar{0}$ is* stable *if and only if one can always find a $\delta(\varepsilon) > 0$, such that,*

$$\|x(0)\| < \delta(\varepsilon) \implies \|x(t)\| < \varepsilon, \forall t \geq 0$$

The equilibrium point is asymptotically stable *if it is stable and* $\lim_{t\to\infty} x(t) = \bar{0}$.

We will often assume that the initial conditions of System (2) belong to a known domain X_0, thereby relaxing the condition $\varepsilon > 0$ to have an upper bound. It is only within this domain of initial conditions that we are interested in studying the stability of x_{eq}.

The trajectory of System (1), plotted in Figure 2, suggests that the equilibrium point $\bar{0}$ is stable. To formally prove this, we can formulate the stability property in terms of the energy of the system in each mode, and consider each mode separately.

Any trajectory starting in mode 2 (with the cord loose) has a certain level of energy given by the kinetic and elastic potential energy of the cart. This energy dictates how far to the right the cart can go, say b. Within that mode, there is no external source of energy and, if the damping coefficient d_s is positive, the cart looses energy (kinetic) over time. Two cases are possible: the cart stops, or moves to the left of the sensor. The first case means that the energy level reached 0. For the second case, ignore what happens to the cart beyond the sensor position. If it comes back to the right of the sensor position with the same or less energy than what it had initially, then it is not possible that it will move beyond b. In fact, if $d_s > 0$ then it will move less and less to the right, as it re-enters mode 2, eventually coming to a rest.

If the above argument applies to the system starting in mode 1 as well, then $\bar{0}$ is stable. And if one of the damping coefficients is positive, then it is asymptotically stable. This is essentially the result described in [4], applied to the cart example.

Without loss of generality, we will always assume that the stability analysis is made for the equilibrium point in mode 1. We now enumerate the formal conditions that need to be met to show the stability of the equilibrium point of System (2).

Suppose we have two continuous differentiable energy functions $V_1(x)$ and $V_2(x)$, and let Σ be a given set of possible switching sequences for solutions of System (2). If for each switching sequence $\sigma(t) \in \Sigma$, the following conditions are satisfied,

Condition 1. $V_{q_i}(x) > 0$, for all $x \in X_{q_i} \setminus \{\bar{0}\}$;

Condition 2. $V_1(\bar{0}) = 0$;

Condition 3. $\dot{V}_{q_i}(x) \leq 0$, for all $x \in X_{q_i} \setminus \{\bar{0}\}$;

Condition 4. $V_{q_i}(x(\bar{t}_{q_i,k+1})) \leq V_{q_i}(x(\bar{t}_{q_i,k})), \forall k$;

where $\bar{t}_{1,k}, \bar{t}_{1,k+1}$ are two consecutive switch-on instants and $q_i \in \{1,2\}$, then the equilibrium point at the origin of System (2) is stable.

In addition, if the following condition is satisfied,

Condition 5. $\dot{V}_q(x) < 0$, for all $x \in X_q \setminus \{\bar{0}\}$;

then the equilibrium point at the origin is asymptotically stable.

It is important to note that, in proving Condition 4, only the trajectories that re-enter mode q_i are of interest. For example, in System (1), not all trajectories

starting in mode 1, will re-enter mode 1 after being in mode 2. Below a given level of energy, the cart cannot reach the sensor, and will thus stay in mode 2.

To relate these formal conditions with the intuitive argument given above, note that $V_{q_i}(x)$ is the energy level when the cart is in state x, and that $\dot{V}_{q_i}(x) \leq 0$ means that the energy is decreasing along the state trajectory, because $\dot{V}_{q_i}(x) = \frac{\partial V_{q_i}}{\partial x}\dot{x}$. Hence, as long as the cart is obeying the dynamics associated with mode q_i, its energy does not increase.

To exemplify the application of these conditions, we show the stability of the zero equilibrium point of the cart, in mode 2, using the energy functions:

$$V_1\left(\left[x_1\ x_2\right]^T\right) = \frac{1}{2}mx_2^2 + \frac{1}{2}(c_s + c_c)x_1^2; \qquad V_2\left(\left[x_1\ x_2\right]^T\right) = \frac{1}{2}mx_2^2 + \frac{1}{2}c_s x_1^2$$

Conditions 1–3 are easy to satisfy, so we prove Condition 4 for mode $q_i = 1$ only. For mode 2 the proof is analogous.

Let $x(\bar{t}_{1,k})$ be the state at the k-th switch on instant of mode 1 (stretched cord). Then we know that $x(\bar{t}_{1,k}) = \left[x_1(\bar{t}_{1,k})\ x_2(\bar{t}_{1,k})\right]^T = \left[l\ v\right]^T$, for some $v < 0$. Over the interval $\bar{t}_{1,k} \leq t \leq \underline{t}_{1,k}$, $x_1(t) \leq l$ and $\dot{V}_1(x) < 0$, hence $V_1(x(\underline{t}_{1,k})) \leq V_1(x(\bar{t}_{1,k}))$. Similarly, over the next interval $\bar{t}_{2,k} \leq t \leq \underline{t}_{2,k}$, $x_1(t) \leq l$ while in mode 2, we have $\dot{V}_2(x) < 0$, hence $V_2(x(\underline{t}_{2,k})) \leq V_2(x(\bar{t}_{2,k}))$. Now, note that since $V_1\left(\left[x_1\ x_2\right]^T\right) = V_2\left(\left[x_1\ x_2\right]^T\right) + \frac{1}{2}c_c x_1^2$, we have:

$$V_2(x(\bar{t}_{1,k+1})) = V_1(x(\bar{t}_{1,k+1})) - \frac{1}{2}c_c l^2 \leq V_2(x(\underline{t}_{1,k})) = V_1(x(\underline{t}_{1,k})) - \frac{1}{2}c_c l^2 \Leftrightarrow$$
$$V_1(x(\bar{t}_{1,k+1})) \leq V_1(x(\underline{t}_{1,k})) \leq V_1(x(\bar{t}_{1,k}))$$

The next section shows how the co-simulation of a stable System (2) can yield unstable trajectories, and the steps that can be taken to prevent so.

3 Hybrid Co-simulation

Co-simulation can be seen as a relaxation of the coupling constraints of a decoupled hybrid system, introduced by the need for a finite frequency of communication between simulators. A particular realization of System (2) can be made with two coupled sub-systems: a software controller, and a plant. During a simulation, the controller reads the state $x(t_i)$ of the plant at a designated communication time t_i, and outputs $\sigma(t_i) \in \{1,2\}$ which decides the mode that the plant should be in, until the next communication time $t_i + H$, with $H > 0$ denoting the communication time step. Figure 3 shows an example where the controller decides to change mode at time t_i because the plant state is above the $g(x)$ surface. This co-simulation approach, often denoted as Jacobi [10], fits most co-simulation scenarios that include software controllers and continuous sub-systems, with a fixed communication step size H.

Due to the fact that the two simulators do not communicate in between communication points, there is a variable delay in the reaction of the controller. As

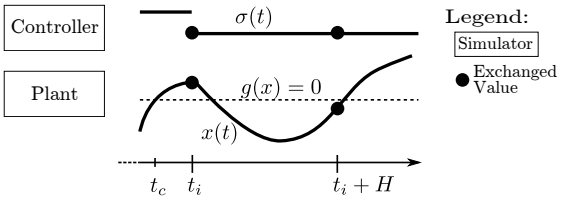


Fig. 3. Co-simulation approach under study.

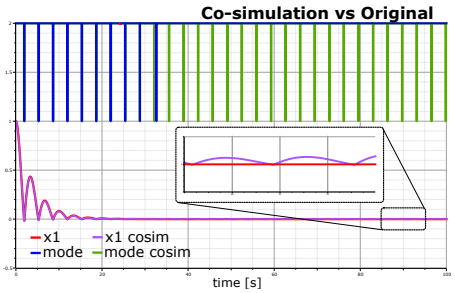


Fig. 4. Co-simulation with $H = 0.05$.

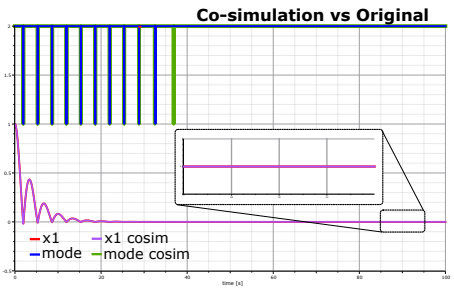


Fig. 5. Co-simulation with $H = 0.001$.

illustrated in Figure 3, the plant has crossed the switching surface at time t_c, before the controller detects that change, at time t_i. This is known as the state event location, or zero crossing detection, problem [27]. We assume that the simulators do not know the exact moment that the plant crosses the surface. This is a reasonable assumption since employing state event location techniques in co-simulation is technically demanding (e.g., it requires the modification of existing simulation tools), has a performance penalty, and there is no standardized interface to communicating switching surfaces between simulators yet (e.g., see [2,7] for proposals).

Fortunately, as we show here, it is possible to select an appropriate communication step size H, that ensures that the co-simulation preserves the stability properties of the original system. To achieve this, we follow an approach that has also been followed for the study of numerical techniques in general simulation (see, e.g., [6]): we model the co-simulation of the original system as a dynamical system.

Figure 4 compares a trajectory approximated by the co-simulation algorithm illustrated in Figure 3, with the analytical solution of System (1) (position and mode). The communication step size is $H = 0.05s$ and all other parameters are as in Figure 2. While the original trajectory is asymptotically stable, the co-simulation is not. The co-simulation keeps alternating between mode 1 and mode 2 while the original trajectory settles in mode 2 after about 33 seconds. For a $H = 0.001s$, the co-simulation at the origin seems to be asymptotically stable, as Figure 5 suggests.

These experimental results hint that H plays an important role in making sure that the co-simulation has the same qualitative behavior as the original system it is intended to represent. Our research questions follows: for a given original system, with a given range of valid initial conditions, what is the *safe* range of communication step sizes that ensures that the co-simulation preserves the stability properties of the original system?

4 Stability Analysis

In order to study the stability of the co-simulation of an original system, we want to apply Conditions 1–4. For that, we need a hybrid automaton that represents the co-simulation algorithm applied to the original system.

Consider the co-simulation algorithm in Figure 3. When there is no mode switch, the reaction delay does not affect the trajectory of the plant, since the plant will anyhow assume the most recently communicated mode. It is only when the plant crosses the switching surface, that the controller reaction delay can affect the co-simulation trajectory. For example, in the figure, the reaction delay at t_i is $t_i - t_c$. Obviously, the reaction delay is bounded by H, as shown at $t_i + H$ in the figure. The distance travelled by the plant before the controller reacts is also finite and depends on the reaction delay.

Therefore, in any co-simulation of a bi-modal system, a mode switch can happen anywhere in time between the moment that the plant crosses the switching surface (zero reaction delay), and *at most* H units of time after the plant crossed the surface. We can build a non-deterministic automaton that captures this behavior.

4.1 Non-deterministic hybrid model of co-simulation

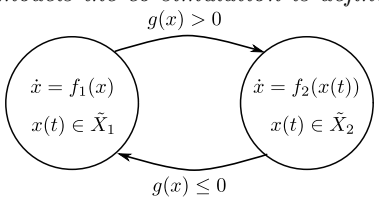
To build such hybrid automaton, we take the original system mode invariants ($x \in X_1$ and $x \in X_2$) in Equation (2), and define new *relaxed* mode invariants, as functions of H, that capture the worst case reaction delay in each mode.

Definition 4. *For a given original invariant set X_{q_i} with $q_i \in \{1,2\}$, the co-simulation and the communication step size H induce a relaxed mode invariant set $\tilde{X}_{q_i}$, defined as the reachable set [15] in H units, starting in X_{q_i}:*

$$\tilde{X}_{q_i}(H) = \left\{\tilde{x}(t) | 0 \le t \le H \text{ and } \tilde{x}(t) \text{ satisfies the ODE } \dot{\tilde{x}} = f_{q_i}(\tilde{x});\ \tilde{x}(0) \in X_{q_i}\right\}$$

Note that $\lim_{H\to 0} \tilde{X}(H)_{q_i} = X_{q_i}$, as expected. We will drop (H) from the notation and just write $\tilde{X}_{q_i}$ from now on.

Definition 5. *Using the relaxed invariant sets of Definition 4, the non-deterministic hybrid automaton that models the co-simulation is defined as:*



(3)

Due to the non-determinism of hybrid automata, the state $x(\bar{t}_{q_i,k})$ at switch-on instant $\bar{t}_{q_i,k}$ of mode q_i can be anywhere in the region $\tilde{X}_{q_j} \setminus X_{q_j}$, with $q_j \neq q_i$ being the other state. The region $D_{q_j} = \tilde{X}_{q_j} \setminus X_{q_j}$ will be called the danger zone of q_j.

Definition 6. *The danger zone of each mode is defined as:*

$$\begin{aligned} D_1(H) &= \left\{\tilde{x}(t) | 0 \leq t \leq H \text{ and } \tilde{x}(t) \text{ is a solution to } \dot{\tilde{x}} = f_1(\tilde{x});\ g(\tilde{x}(t)) \geq 0\right\} \\ D_2(H) &= \left\{\tilde{x}(t) | 0 \leq t \leq H \text{ and } \tilde{x}(t) \text{ is a solution to } \dot{\tilde{x}} = f_2(\tilde{x});\ g(\tilde{x}(t)) \leq 0\right\} \end{aligned} \tag{4}$$

For $H = 0$, the danger zone D_{q_i} coincides with the switching surface of the original system. Furthermore, by definition, $0 < H_r \leq H_s$, implies that $D_{q_i}(H_r) \subseteq D_{q_i}(H_s)$.

Definition 6 allows us to approximate D_{q_i}, for a given H, by solving simultaneously a set of ordinary differential equations, whose initial value is a point in the switching surface. Continuity ensures that this set can be approximated with arbitrary accuracy. However, D_{q_i} is unbounded if the set of points in the surface (solutions to $g(\tilde{x}) = 0$) is unbounded. In that case, we construct D_{q_i} for a bounded set of points, that make sense in the physics of the original system. For example, in the cart system, the set of solutions to the surface equation $x_1 - l = 0$ is unbounded (all speeds at the sensor position are possible), but it is reasonable to assume that $|x_2| < 100$.

4.2 Stability analysis

We are interested in studying the stability of System (3), assuming that the original system is stable (or asymptotically stable). We therefore assume that the energy functions V_1 and V_2 are given, so we do not need to build new ones for the co-simulation hybrid automaton. With these, we essentially restrict H until any solution to System (3) satisfies each of the Conditions 1–4.

Condition 1 needs to be satisfied for all $x \in \tilde{X}_{q_i} \setminus \{\bar{0}\}$. However, it suffices to show it for all $x \in D_{q_i} \setminus \{\bar{0}\}$. This can be done operationally by iteratively approximating the danger zone $D_{q_i}(H)$ for increasingly large values of H, as long as:

$$D_{q_i} \subseteq \left\{x \in \mathbb{R}^n : V_{q_i}(x) > 0\right\} \Leftrightarrow D_{q_i} \setminus \left\{x \in \mathbb{R}^n : V_{q_i}(x) > 0\right\} = \emptyset$$

Condition 2 is always satisfied by the co-simulation.

Condition 3 is checked by the same procedure as Condition 1: approximate the danger zone D_{q_i} for increasingly large values of H, as long as:

$$D_{q_i} \subseteq \left\{x \in \mathbb{R}^n : \dot{V}_{q_i}(x) \leq 0\right\} \Leftrightarrow D_{q_i} \setminus \left\{x \in \mathbb{R}^n : \dot{V}_{q_i}(x) \leq 0\right\} = \emptyset$$

To find the largest value H that proves Condition 4, we must understand how the co-simulation influences the energy of the system. Without loss of generality, we focus on studying the condition for mode $q_i = 1$. The analysis for mode 2 follows the same steps.

Figure 6 sketches the case where $H = 0$, that is, where the co-simulation behaves exactly as the original system, which satisfies the condition by assumption. In the figure, the notation $V_{q_i,k} = V_{q_i}(x(\bar{t}_{q_i,k}))$. The trajectory shown in grey

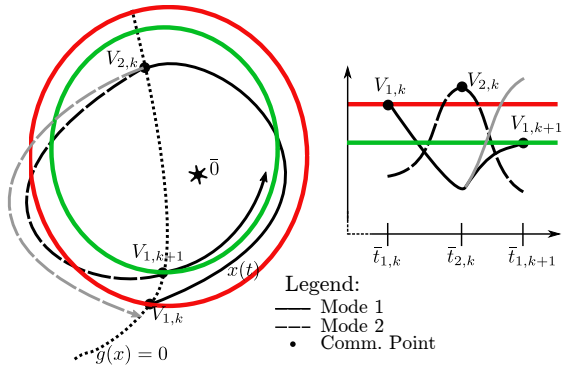


Fig. 6. Co-simulation trajectory for $H = 0$.

violates the condition because it re-enters mode 1 with a level of energy higher than $V_{1,k}$.

For $H = \varepsilon > 0$, the point $x(\bar{t}_{1,k}) \in D_2$, and $x(t)$ crosses the switching surface at some time $t_{c_1} \leq \bar{t}_{1,k}$, before switching from mode 2 to mode 1. Following the definition of System (3), $x(t)$ obeys:

$$\dot{x}(t) = f_2(x(t)) \text{ with } t_{c_1} \leq t \leq \bar{t}_{1,k} \text{ for some } \bar{t}_{1,k} \leq t_{c_1} + H \tag{5}$$

where $x(t_{c_1})$ is any given value that satisfies $g(x(t_{c_1})) = 0$.

The value $\bar{t}_{1,k}$ is uncertain, but bounded by $t_{c_1} + H$. It can be the case that $\bar{t}_{1,k} > t_{c_1}$ and $V_1(x(t_{c_1})) < V_1(x(\bar{t}_{1,k}))$, which means that the co-simulation, due to a switch that is delayed by $\bar{t}_{1,k} - t_{c_1}$ units of time, can introduce extra energy into the system, compared to the original system (this is essentially what happened in Figure 4, where the co-simulation introduces just enough energy for the cart to always leave mode 2 again). However, the energy change caused by the co-simulation is always finite, as the functions f_1, f_2 are known to be continuous; further, it is known that, as H approaches zero, and the co-simulation approaches the behaviour of the original system, the change in energy also approaches zero.

In the following paragraphs, to facilitate the explanation, we assume that

$$\begin{aligned} \bar{t}_{1,k} = t_{c_1}, \quad & \min_{t_{c_1} \leq t \leq t_{c_1}+H} V_1(x(t)) = V_1(x(t_{c_1})), \\ & \max_{t_{c_1} \leq t \leq t_{c_1}+H} V_1(x(t)) = V_1(x(t_{c_1} + H)) \end{aligned} \tag{6}$$

In other words, V_1 is minimal under the dynamics of mode 2 at the switching surface, maximal away from the surface after H units of time, and the co-simulation has made the switch to mode 1 at the surface, introducing at that moment, a minimal amount of energy (compared to the original system).

After the switch to mode 1 is made, the co-simulation trajectory $x(t)$ satisfies the dynamics of mode 1 in the interval $\left[\bar{t}_{1,k}, \underline{t}_{1,k}\right]$ Furthermore, by assumption, there exists a t_{c_2} such that $\bar{t}_{1,k} < t_{c_2} \leq \underline{t}_{1,k}$ and $g(x(t_{c_2})) = 0$.

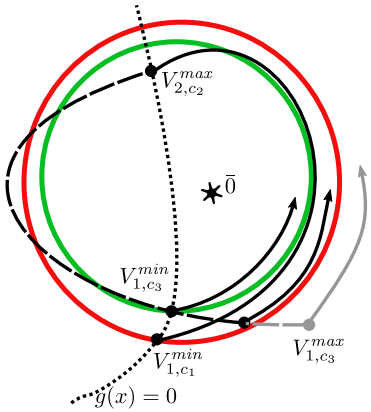


Fig. 7. Co-simulaton effect in trajectories with assumptions 6 and 7.

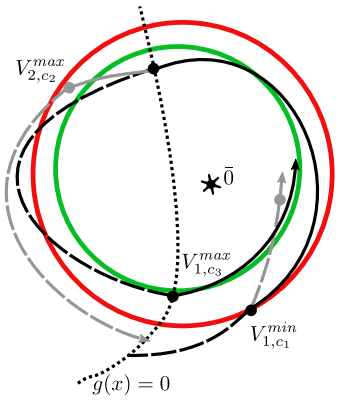


Fig. 8. Co-simulaton effect in general trajectories.

For the sake of the argument, assume that the co-simulation switches right at the switching surface and that it introduces *maximal* energy, that is:

$$\underline{t}_{1,k} = \bar{t}_{2,k} = t_{c_2}, \qquad \max_{t_{c_2} \le t \le t_{c_2}+H} V_2(x(t)) = V_2(x(t_{c_2})) \tag{7}$$

After the switch to mode 2 is made, $x(t)$ satisfies the dynamics of mode 2 in the interval $\left[\underline{t}_{1,k}, \bar{t}_{1,k+1}\right]$. By assumption, there exists a $\underline{t}_{1,k} < t_{c_3} \le \bar{t}_{1,k+1}$ such that $g(x(t_{c_3})) = 0$.

Assumptions 6 and 7 where made to make sure that any co-simulation trajectory $x(t)$ under study is exactly the same the original system trajectory (with the same initial value), up to time t_{c_3}. Under these assumptions, $\bar{t}_{1,k} = t_{c_1}$ and $\underline{t}_{1,k} = t_{c_2}$ are uniquely defined.

For the re-entry in mode 1, the actual time of the mode switch $\bar{t}_{1,k+1}$ is uncertain but bounded by $t_{c_3} + H$. Hence, there are infinitely many trajectories that re-enter mode 1 at each time between t_{c_3} and $t_{c_3} + H$, starting from the same initial value $x(t_{c_1})$. Figure 7 shows three of these possible trajectories satisfying the above equations. Two of those trajectories satisfy Condition 4 but the third one, in gray, does not. The notation $V^{max}_{q_i,c_i} = \max_{t_{c_i} \le t \le t_{c_i}+H} V_{q_i}(x(t))$. It stays in mode 2 for too long, increasing the energy level of the system beyond the limit $V_1(x(\bar{t}_{1,k}))$.

As Figure 7 suggests, for a given initial value $x(t_{c_1})$, we do not need to consider all possible trajectories. It suffices to make sure H is small enough so that the condition is satisfied for the $x(t)$ where $V_1(x(\bar{t}_{1,k+1}))$ is maximal, so that the grey trajectory can never happen. Any other trajectory will then satisfy the condition as well. Formally, we consider $x(t)$ such that $V_1(x(\bar{t}_{1,k+1})) = \max_{t_{c_1} \le t \le t_{c_1}+H} V_1(x(t))$.

The above result is incomplete because we made assumptions 6 and 7, to give the intuition. Relaxing these assumptions, we have the following general result. It suffices to check the trajectories $x(t)$ for which:

1. The initial entry in mode 1 *minimizes* the added energy to the system. That is, $\bar{t}_{1,k}$ is such that
$$V_1(x(\bar{t}_{1,k})) = \min_{t_{c_1} \le t \le t_{c_1}+H} V_1(x(t)) \tag{8}$$

2. The intermediate entry in mode 2 *maximizes* the energy added to the system. That is, $\underline{t}_{1,k}$ is such that
$$V_2(\underline{t}_{1,k}) = \max_{t_{c_2} \le t \le t_{c_2}+H} V_2(x(t)) \tag{9}$$

3. The re-entry in mode 1 *maximizes* the energy added to the system. Formally, $\bar{t}_{1,k+1}$ satisfies
$$V_1(x(\bar{t}_{1,k+1})) = \max_{t_{c_1} \le t \le t_{c_1}+H} V_1(x(t)) \tag{10}$$

These trajectories represent essentially the worst case scenario in terms of energy distortion, caused by $H > 0$. Note that maximizing the energy at the intermediate entry in mode 2 increases the maximum energy when re-entering mode 1 later. Figure 8 shows a sketch of three trajectories, only one of which satisfies the stability condition. The grey ones do not.

Based on the above result, we can compute a safe H as follows:

1. Start with an initial $H \leftarrow H_0$.
2. Pick a range of values $x^{[s]} \in \mathbb{R}^n$ that are in the switching surface, that is, $g(x^{[s]}) = 0$.
3. For each $x^{[s]}$ in the switching surface:
   (a) Solve Equation (5), with $x(t_{c_1}) = x^{[s]}, t_{c_1} = 0$ and find the $\bar{t}_{1,k}$ that satisfies Equation (8).
   (b) Compute the trajectory in mode 1, find t_{c_2} and $\underline{t}_{1,k}$ such that Equation (9) is satisfied.
   (c) If t_{c_2} does not exist, the trajectory has not enough energy to change mode, so the initial value $x^{[s]}$ and any other initial value $x^{[r]}$ that satisfies $V_1(x^{[r]}) < V_1(x^{[s]})$ can be safely ignored.
   (d) Compute the trajectory in mode 2, find t_{c_3}, and $\bar{t}_{1,k+1}$ such that Equation (10) is satisfied.
   (e) If t_{c_3} does not exist, end the current iteration.
   (f) If Condition 4 is not satisfied, decrease H and go to step 3a.

The above algorithm ignores trajectories that do not meet the pre-requisites to be considered for Condition 4. After it is applied to both modes, the smallest H is the largest safest H that can be used to ensure that the Condition 1 is met. Adaptations for checking asymptotic stability are straightforward.

The algorithm terminates because, even when the range of initial values in the switching surface is infinite: 1) there is a lower limit to the level of energy of the points in the surface that need to be considered; 2) not all initial values are physically meaningful.

An application of this algorithm to the cart example is available for download[5]. The result is $H < 0.039$ for mode 2, and $H < 0.0027$ for 1. Therefore the overall safest $H < 0.0027$.

[5] http://msdl.cs.mcgill.ca/people/claudio/projs/AnalysisCart.zip

5 Conclusion

This work presents an analysis procedure, and a conservative algorithm, to compute a safe range of communication step sizes which ensures that the co-simulation preserves the stability properties of the original system. As part of the analysis, we show how to model a co-simulation as a non-deterministic hybrid automaton, and how to satisfy the stability conditions presented in [4].

This is only a baby step in the analysis of co-simulation applied to hybrid systems, and the following are some of the important limitations: (i) it applies only to bi-modal hybrid automata that have at most one equilibrium in each mode, at the origin; (ii) only the Jacobi orchestration approach is considered. (iii) the current implementation of the algorithm is slow and requires insight about the physics of the original system; (iv) we require access to the equations of the original system, or at least, we need to be able to simulate them.

Ongoing and future work aims at addressing these limitations. For example, we are researching into how to generalize the analysis for multi-modal systems, with multiple equilibria, such as the ones studied in [19,20]. Furthermore, we are exploring the possibility of using the FMI standard to simulate the original system in multiple modes, thus avoiding the need to disclose important information (such as intellectual property) about its dynamics.

With this and future work, we wish that system integrators running co-simulations can trust their results, thus enhancing the development of complex systems.

References

1. Arnold, M.: Stability of Sequential Modular Time Integration Methods for Coupled Multibody System Models. Journal of Computational and Nonlinear Dynamics 5(3), 031003 (may 2010)
2. Bertsch, C., Ahle, E., Schulmeister, U.: The Functional Mockup Interface-seen from an industrial perspective. In: 10th International Modelica Conference (2014)
3. Blockwitz, T., Otter, M., Akesson, J., Arnold, M., Clauss, C., Elmqvist, H., Friedrich, M., Junghanns, A., Mauss, J., Neumerkel, D., Olsson, H., Viel, A.: Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. In: 9th International MODELICA Conference. pp. 173–184. Linköping University Electronic Press, Munich, Germany (nov 2012)
4. Branicky, M.: Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. IEEE Transactions on Automatic Control 43(4), 475–482 (apr 1998)
5. Busch, M.: Continuous approximation techniques for co-simulation methods: Analysis of numerical stability and local error. ZAMM - Journal of Applied Mathematics and Mechanics 96(9), 1061–1081 (sep 2016)
6. Cellier, F.E., Kofman, E.: Continuous System Simulation. Springer Science & Business Media (2006)
7. Cremona, F., Lohstroh, M., Broman, D., Di Natale, M., Lee, E.A., Tripakis, S.: Step Revision in Hybrid Co-simulation with FMI. In: 14th ACM-IEEE International Conference on formal Methods and Models for System Design. Kanpur, India (2016)

8. Friedman, J., Ghidella, J.: Using Model-Based Design for Automotive Systems Engineering - Requirements Analysis of the Power Window Example. SAE Technical Paper (apr 2006)
9. Goebel, R., Sanfelice, R.G., Teel, A.R.: Hybrid Dynamical Systems: Modeling, Stability and Robustness. Princeton University Press (2012)
10. Gomes, C., Thule, C., Broman, D., Larsen, P.G., Vangheluwe, H.: Co-simulation: State of the art. Tech. rep. (feb 2017), http://arxiv.org/abs/1702.00686
11. Kalmar-Nagy, T., Stanciulescu, I.: Can complex systems really be simulated? Applied Mathematics and Computation 227, 199–211 (jan 2014)
12. Karalis, P., Navarro-López, E.M.: Feedback stability for dissipative switched systems. In: Proceedings of the 20th IFAC World Congress. p. to appear. IFAC, Toulouse, France (2017)
13. Khalil, H.K.: Nonlinear Systems. Prentice-Hall, New Jersey (1996)
14. Liberzon, D.: Switching in Systems and Control. Springer Science & Business Media (2012)
15. Lygeros, J., Johansson, K., Simic, S., Jun Zhang, Sastry, S.: Dynamical properties of hybrid automata. IEEE Transactions on Automatic Control 48(1), 2–17 (jan 2003)
16. Lygeros, J.: Lecture notes on hybrid systems (2004)
17. Mitra, S., Liberzon, D., Lynch, N.: Verifying average dwell time of hybrid systems. ACM Transactions on Embedded Computing Systems 8(1), 1–37 (dec 2008)
18. Navarro-López, E.M., Carter, R.: Hybrid automata: an insight into the discrete abstraction of discontinuous systems. International Journal of Systems Science 42(11), 1883–1898 (nov 2011)
19. Navarro-López, E.M., Carter, R.: Deadness and how to disprove liveness in hybrid dynamical systems. Theoretical Computer Science 642, 1–23 (aug 2016)
20. Navarro-López, E.M., Laila, D.S.: Group and Total Dissipativity and Stability of Multi-Equilibria Hybrid Automata. IEEE Transactions on Automatic Control 58(12), 3196–3202 (dec 2013)
21. Ni, Y., Broenink, J.F.: Hybrid systems modelling and simulation in DESTECS: a co-simulation approach. In: Klumpp, M. (ed.) 26th European Simulation and Modelling Conference. pp. 32–36. EUROSIS-ETI, Ghent, Belgium (2012)
22. Nielsen, C.B., Larsen, P.G., Fitzgerald, J., Woodcock, J., Peleska, J.: Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions. ACM Comput. Surv. 48(2), 18:1—-18:41 (sep 2015)
23. Schweizer, B., Li, P., Lu, D.: Explicit and Implicit Cosimulation Methods: Stability and Convergence Analysis for Different Solver Coupling Approaches. Journal of Computational and Nonlinear Dynamics 10(5), 051007 (sep 2015)
24. Tomiyama, T., D'Amelio, V., Urbanic, J., ElMaraghy, W.: Complexity of Multi-Disciplinary Design. CIRP Annals - Manufacturing Technology 56(1), 185–188 (2007)
25. Van der Auweraer, H., Anthonis, J., De Bruyne, S., Leuridan, J.: Virtual engineering at work: the challenges for designing mechatronic products. Engineering with Computers 29(3), 389–408 (2013)
26. Vangheluwe, H., De Lara, J., Mosterman, P.J.: An introduction to multi-paradigm modelling and simulation. In: Proceedings of AIS2002 (AI, Simulation & Planning). pp. 9–20. SCS (2002)
27. Zhang, F., Yeddanapudi, M., Mosterman, P.: Zero-crossing location and detection algorithms for hybrid system simulation. In: IFAC World Congress. pp. 7967–7972 (2008)