CHATTERING AVOIDANCE IN HYBRID SIMULATION MODELS: A MODULAR APPROACH BASED ON THE HYFLOW FORMALISM

Fernando J. Barros
Department of Informatics Engineering
University of Coimbra
3030 Coimbra, Portugal
barros@dei.uc.pt

ABSTRACT

The possibility that a hybrid simulation model can change its state infinitely often in a finite time interval impacts negatively, or even disables, the ability to simulate some hybrid systems. This behavior also known as *chattering* occurs in models that can switch continuously between different modes of operation. In this paper we provide two solutions to different kinds of chattering behavior. One solution is based on model discretization and on sliding mode control, enabling the elimination of chattering in linear systems with relay components. Another approach is based on dynamic topology models. This solution helps eliminate chattering in models that must be prevented to change their operation mode. We provide examples for these approaches and we show that chattering is avoided enabling an efficient simulation of hybrid systems. Solutions are described in HyFlow, a formalism supporting the definition and the interoperability of modular hybrid components.

Keywords: hybrid simulation models, chattering-free simulation, dynamic topology models, HyFlow formalism, co-simulation.

1 INTRODUCTION

The simulation of some hybrid models exhibits a chattering behavior that imposes very small step-sizes, highly increasing the computational cost. Chattering can be created, for example, by relay control systems where actions are determined by the value of state variables requiring, in certain cases, the repeatedly switching between modes of operation. In the limit, a system can change mode infinitely often in a finite time interval. Chattering elimination becomes thus a necessary condition to enable the simulation of many systems. An example of chattering, is a mechanical system where a surface exerts a friction force over a moving body. The force opposes to velocity, and it changes abruptly when velocity modifies its direction. This change originates a discontinuity and it requires the creation of an event. This system if not properly modeled can cause the creation of a very large number of events when operated close to zero velocity, that will slow the simulation.

Another problem associated with chattering is the handling of discontinuities when high accurate numerical methods are required. The most efficient numerical solvers use adaptive step-size for achieving a high accuracy. They start usually with a very small step-size and they adjust it according to accuracy constraints. In these methods after a discontinuity, the smallest step-size is employed before the methods reaches the required order. Chattering, by imposing discontinuities in the system, may prevent solvers ever to reach the required order, imposing the successive reset of the step-size. Thus, chattering when represented by adaptive

SpringSim-TMS/DEVS 2017, April 23-26, Virginia Beach, VA, USA

step-size high-order methods can eventually make the simulation come to an halt by imposing the minimum step size to be used during a long series of transitions. Given a typical value of 10^{-6} for the smallest step size, chattering can make high accuracy numerical simulations virtually impossible.

In many cases chattering is not present in the real system, but it can be originated by approximations introduced in model development. Chattering in real systems, specially in mechanical devices, is also usually avoided since it may significantly reduce component lifetime.

In this paper we consider discrete sliding model control (SMC) (Su, Drakunov, and Ozguner 2000) and dynamic topologies (Barros 2003) as a construct to enable the development of chattering-free models of hybrid systems. Dynamic topologies are used here for modeling systems that cannot leave the current model of operation, and model adaptation is used to guaranteed that key constraints are obeyed. This is the case of liquid volume in a tank that must be a non-negative value at anytime (Barros 2011).

We note that chattering behavior is inherent to the some systems, like relay systems, and no *simple* model can eliminate this problem. In fact state-of-the art solutions are based on sliding-mode-control (Kamal, Moreno, Chalanga, Bandyopadhyay, and Fridman 2016), or in techniques involving the numerical computation of directional derivatives (Aljarbouh and Caillaud 2016), to maintain the system in the sliding region. The use of a model-based strategy to avoid chattering in some systems is described in Section 5, and it involves adjusting models at runtime.

The paper is organized as follows. Section 2 describes discrete Sliding Model Control (SMC). The HyFlow formalism is described in Section 3. Results for a chattering-free simulation of linear systems are given in Section 4. The use of dynamic topologies for obtaining chattering avoidance is shown in Section 5.

2 SLIDING-MODE CONTROL

A simple solution to chattering involves the use of digital controllers (Galias and Yu 2008), (Wang, Yu, and Chen 2009). In this approach, instead of letting the model undergo a chattering behavior, one can set digital controller sampling rate for enabling an effective control over the rate of switching mode and the creation of events. This approach, minimizes the effect on numerical solvers.

In this paper we consider relay linear systems described by:

$$\dot{x} = Ax(t) + Bu(t) \tag{1}$$

and

$$y(t) = Cx(t) \tag{2}$$

Sliding mode control (SMC) defines the action

$$u(t) = \operatorname{sgn}(y(t)) \tag{3}$$

with

$$\operatorname{sgn}(v) = \begin{cases} -1, & v < 0 \\ [-1,1], & v = 0 \\ 1, & v > 0 \end{cases}$$

The control (3) is a continuous function that drives the system into the sliding (manifold) surface $y = 0$. However, in the continuous form SMC may become difficult to achieve in many physical systems due to chattering. In fact, when in the vicinity of the sliding surface control may switch between the values -1 and 1, imposing very fast changes in the real system that can, in many cases, cause damage the system.

Given the difficultly brought by chattering, a discrete zero-order holder (ZOH) controller equivalent was developed in (Su, Drakunov, and Ozguner 2000). In this approach, a fixed sampling rate is used, and the control value is kept constant between samples. This value is set to drive the system into the sliding region, and it is given by:

$$u'_k = -(C\Gamma)^{-1} C \Phi x_k \tag{4}$$

where

$$\Phi = e^{Ah}$$
$$\Gamma = \Big(\int_0^h e^{At}\,\mathrm{d}t \Big) B$$

In many systems the admissible control is limited to an interval $[min, max]$ and thus the actual control is given by:

$$u_k = \mathrm{sat}(min, max, u'_k) = \begin{cases} min & u'_k < min \\ u'_k & u'_k \in [min, max] \\ max & u'_k > max \end{cases} \tag{5}$$

Hybrid models can suffer form other type of efficiency issues. This is the case of Zeno behavior that occurs, for example, in the bouncing ball system (Johansson, Egerstedt, Lygeros, and Sastry 1999). Zeno behavior, however, requires a different approach not involving SMC.

In the next section we provide a description of the discrete SMC controller in the HyFlow formalism (Barros 2003). This representation makes it possible to seamlessly integrate SMC with other HyFlow-based models, including event detectors and numerical integrators, enabling the description of complex hybrid systems.

3 THE HYFLOW FORMALISM

The Hybrid Flow System Specification (HyFlow) is a formalism aimed to represent hybrid systems with a time-variant topology (Barros 2008). HyFlow achieves the representation of continuous variables using the concept of multi-sampling, while the representation of discrete events is based on the Discrete Event System Specification (DEVS) (Zeigler, Praehofer, and Kim 2000). HyFlow has two types of models: basic and network. Basic models provide state representation and transition functions. Network models are a composition of basic models and/or other network models. Given its definition, a network provides an abstraction for representing hierarchical systems.

3.1 HyFlow Basic Model

We consider $\widehat{B}$ as the set of names corresponding to basic HyFlow models. A HyFlow basic model associated with name $B \in \widehat{B}$ is defined by

$$M_B = (X, Y, P, \rho, \omega, s_0, \delta, \bar{\Lambda}, \ddot{\Lambda})$$

where

$X = \bar{X} \times \ddot{X}$ is the set of input flow values

$\bar{X}$ is the set of continuous input flow values
$\ddot{X}$ is the set of discrete input flow values
$Y = \bar{Y} \times \ddot{Y}$ is the set of output flow values
$\bar{Y}$ is the set of continuous output flow values
$\ddot{Y}$ is the set of discrete output flow values
P is the set of partial states (p-states)
$\rho : P \longrightarrow \mathbb{H}_0^+$ is the time-to-input function
$\omega : P \longrightarrow \mathbb{H}_0^+$ is the time-to-output function
$S = \{(p, e) | p \in P, 0 \le e \le \nu(p)\}$ is the state set
with $\nu(p) = \min\{\rho(p), \omega(p)\}$, the time-to-transition function
$s_0 \in S$ is the initial state
$\delta : S \times X^\phi \longrightarrow P$ is the transition function
where $X^\phi = \bar{X} \times (\ddot{X} \cup \{\phi\})$
and ϕ is the null value (absence of value)
$\bar{\Lambda} : S \longrightarrow \bar{Y}$ is the continuous output function
$\ddot{\Lambda} : S \longrightarrow \ddot{Y}$ is the discrete output function

HyFlow time base is the set of hyperreals numbers $\mathbb{H} = \{x + z\varepsilon | x \in \mathbb{R}, z \in \mathbb{Z}\}$, where ε is an infinitesimal value. The set of positive hyperreals is defined by $\mathbb{H}_0^+ = \{h \in \mathbb{H} | h \ge 0\}$. The discrete output of a component described by a HyFlow basic model is constrained to be null (ϕ) when in a state (s, e) with $e \neq \omega(p)$.

Figure 1 depicts the typical trajectories of a HyFlow component. At time t_1 the component in p-state p_0 samples its input since its elapsed time reaches $\omega(p_0) = \rho(p_0) = e$. The component changes its p-state to $p_1 = \delta((p_0, \rho(p_0)), (x_1, \phi))$, where x_1 is the sampled value and no discrete flow is present.

At time t_2 the discrete flow x_d is received by the component that changes to p-state $p_2 = \delta((p_1, e_1), (x_2, x_d))$, where x_2 is the continuous flow at t_2. At time t_3 the component reaches the time-to-output time limit and it changes to p-state $p_3 = \delta((p_2, \omega(p_2)), (x_3, \phi))$. At this time the discrete flow $y_d = \lambda(p_2, \omega(p_2))$ is produced. Additionally, component continuous output flow is always present and given by $\bar{\Lambda}(p, e)$. The formalism has shown to be deterministic and closed under the coupling operation. A detailed description of HyFlow semantics with the corresponding simulators is given in (Barros 2008).

Example: Discrete Sliding Mode Controller

In previous work we have created a large variety of HyFlow models for representing, integrators, event detectors, digital filters and fluid stochastic Petri Nets. (Barros 2015b). We provide here the description of discrete sliding-mode controllers (SMC) in HyFlow. This representation enables the seamless integration of discrete SMC components with existing HyFlow models, offering the possibility of modeling complex systems as a combinations of components. Given a set of matrices A, B and C describing the linear system (1)-(2), a discrete SMC component with period T is defined by:

$$M_{SMC}(A, B, C, T) = (X, Y, P, \rho, \omega, \delta, \bar{\Lambda}, \lambda)_{s_0}$$

where

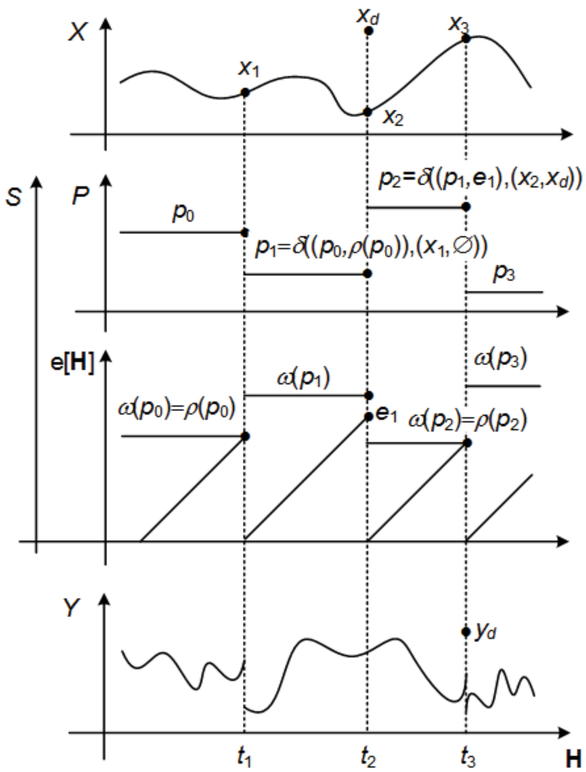


Figure 1: Basic HyFlow component trajectories.

$X = \mathbb{R} \times \{\}$
$Y = \mathbb{R} \times \mathbb{R}$
$P = \{(\alpha, \beta, u_{k-1}) | \alpha, \beta \in \mathbb{H}_0^+; u_{k-1} \in \mathbb{R}\}$
$\rho(\alpha, \beta, u_{k-1}) = \alpha$
$\omega(\alpha, \beta, u_{k-1}) = \beta$

$$\delta(((\alpha, \beta, u_{k-1}), e), (x_k, x_d)) = \begin{cases} (T, \infty, u_{k-1}) & \text{if } e = \alpha + \varepsilon \wedge |u_{k-1} - u_k| < tol \\ (\infty, 0, u_k) & \text{if } e = \alpha + \varepsilon \wedge |u_{k-1} - u_k| \geq tol \\ (T, \infty, u_k) & \text{if } e = \varepsilon \end{cases}$$

where $u_k = \mathrm{sat}(-1, 1, u'_k), u'_k = -(C\Gamma)^{-1} C \Phi x_k, \Phi = e^{AT},$ and $\Gamma = \left(\int_0^T e^{At} \, \mathrm{d}t \right) B$

$\bar{\Lambda}((\alpha, \beta, u_{k-1}), e) = u_{k-1}$
$\lambda(\alpha, \beta, u_{k-1}) = u_{k-1}$
$s_0 = ((0, \infty, \infty), 0)$

This component samples the input with period T. The control action is computed by the transition function δ. For performance reasons a discrete control action is only produced if the new value differs from the previous by tol. This is particularly important when there is a series of constant values. In this case solvers accuracy is not affected. However, performance is increased since at each discrete event value sent by the controller solvers need to be (re)initialized, increasing simulation runtime.

3.2 HyFlow Network Model

HyFlow network models are compositions of HyFlow models (basic or other HyFlow network models). Let $\widehat{N}$ be the set of names corresponding to HyFlow network models, with $\widehat{N} \cap \widehat{B} = \{\}$. Formally, a HyFlow network model associated with name $N \in \widehat{N}$ is defined by

$$M_N = (X, Y, \eta)$$

where

N is the network name
$X = \bar{X} \times \ddot{X}$ is the set of network input flows
  $\bar{X}$ is the set of network continuous input flows
  $\ddot{X}$ is the set of network discrete input flows
$Y = \bar{Y} \times \ddot{Y}$ is the set of network output flows
  $\bar{Y}$ is the set of network continuous output flows
  $\ddot{Y}$ is the set of network discrete output flows
$\eta \in \widehat{\eta}$ is the name of the dynamic topology network executive
with
  $\eta \in \widehat{\eta}$ representing the set of all names associated with HyFlow executive models, constrained to $\widehat{\eta} \cap \widehat{B} = \widehat{\eta} \cap \widehat{N} = \{\}$

Executives are uniquely assigned to network models, i.e.,

$$\forall_{i,j \in \widehat{N}, i \neq j} \eta_i \neq \eta_j \text{ with } M_k = (X_k, Y_k, \eta_k), \forall_{k \in \widehat{N}}$$

The model of the executive is a modified HyFlow basic model, defined by

$$M_\eta = (X_\eta, Y_\eta, P, \rho, \omega, s_0, \delta, \Lambda, \gamma, \widehat{\Sigma})$$

where

$\widehat{\Sigma}$ is the set of network topologies
$\gamma : P \longrightarrow \widehat{\Sigma}$ is the topology function

The network topology $\Sigma_\alpha \in \widehat{\Sigma}$, corresponding to the p-state $p_\alpha \in P$, is given by the 4-tuple

$$\Sigma_\alpha = \gamma(p_\alpha) = (C_\alpha, \{I_{i,\alpha}\} \cup \{I_{\eta,\alpha}, I_{N,\alpha}\}, \{F_{i,\alpha}\} \cup \{F_{\eta,\alpha}, F_{N,\alpha}\})$$

where

C_α is the set of names associated with the executive state p_α
for all $i \in C_\alpha \cup \{\eta\}$

$I_{i,\alpha}$ is the sequence of influencers of i
$F_{i,\alpha}$ is the input function of i
$I_{N,\alpha}$ is the sequence of network influencers
$F_{N,\alpha}$ is the network output function

For all $i \in C_\alpha$

$M_i = (X_i, Y_i, P_i, \rho_i, \omega_i, s_{0,i}, \delta_i, \bar{\Lambda}_i, \lambda_i)$ if $i \in \widehat{B}$
$M_i = (X_i, Y_i, \eta_i)$ if $i \in \widehat{N}$

Variables are subjected to the following constraints for every $p_\alpha \in P_\alpha$

$N \notin C_\alpha, N \notin I_{N,\alpha}, \eta \notin C_\alpha$
$F_{N,\alpha} : \times_{k \in I_{N,\alpha}} Y_k \longrightarrow Y^\phi$
$F_{i,\alpha} : \times_{k \in I_{i,\alpha}} V_k \longrightarrow X_i^\phi$
where

$$V_k = \begin{cases} Y_k^\phi & \text{if } k \neq N \\ X^\phi & \text{if } k = N \end{cases}$$

and
$F_{N,\alpha}((\bar{v}_{k_1}, \phi), (\bar{v}_{k_2}, \phi), ...) = (\bar{y}_N, \phi)$
$F_{i,\alpha}((\bar{v}_{k_1}, \phi), (\bar{v}_{k_2}, \phi), ...) = (\bar{x}_i, \phi)$

These two last constraints are a characteristic of discrete systems and impose that non-null values cannot be created from a sequence composed exclusively by null values.

The topology of a network is defined by its executive through the topology function γ, that maps executive p-state into network composition and coupling. Thus, topology adaption can be achieved by changing executive p-state.

4 SIMULATION RESULTS FOR SLIDING-MODE CONTROL

We consider the system of Figure 2 composed by two masses M_1 and M_2 subjected to friction forces. Friction is present between masses M_1 and M_2, while the interface between M_2 and the ground is frictionless. Masses are attached to the wall by two springs K_1 and K_2.

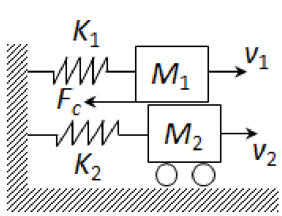


Figure 2: Two mass system with friction.

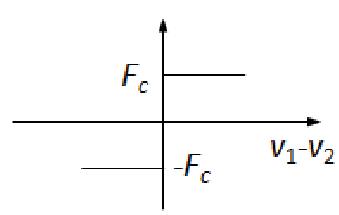


Figure 3: Friction force.

The system is described, according to (1) and (2), by:

$$\begin{bmatrix} \dot{v_1} \\ \dot{x_1} \\ \dot{v_2} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{K_1}{M_1} & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{K_2}{M_2} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ x_1 \\ v_2 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{F_C}{M_1} \\ 0 \\ -\frac{F_C}{M_2} \\ 0 \end{bmatrix} u(t)$$

and

$$y = \begin{bmatrix} 1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ x_1 \\ v_2 \\ x_2 \end{bmatrix}$$

The friction force is modeled by a discrete sliding mode controller and it is given by (4) and (5). In this system the sliding region is defined by the condition $v_1 = v_2$.

The HyFlow network model describing this system is given in Figure 4. This network has static topology associated with the executive p-state p and it is defined by:

$\gamma(p) = (C, \{I_i | i \in C\} \cup \{I_\eta, I_N\}, \{F_i | i \in C\} \cup \{F_\eta, F_N\})$, where
$C = \{v_1, x_1, v_2, x_2, U\}$
$I_\eta = I_N = \{\}, I_{x_1} = \{v_1, U\}, I_{v_1} = \{x_1, U\}, I_{x_2} = \{v_2, U\}, I_{v_2} = \{x_2, U\}, I_U = \{v_1, x_1, v_2, x_2\}$
$F_\eta() = F_N() = (\phi, \phi)$
$F_{v_1}(x_1, u) = (\frac{-x_{1c}K_1 + F_C u_c}{M_2}, \mathrm{reset}(x_1, u))$
$F_{x_1}(v_1, u) = (v_{1c}, \mathrm{reset}(v_1, u))$
$F_{v_2}(x_2, u) = (\frac{-x_{2c}K_2 - F_C u_c}{M_2}, \mathrm{reset}(x_2, u))$
$F_{x_2}(v_2, u) = (v_{2c}, \mathrm{reset}(v_2, u))$
$F_U(x_1, v_1, x_2, v_2) = ([x_{1c}, v_{1c}, x_{2c}, v_{2c}]^\intercal, \phi)$,

In the description we have assumed that a signal s can be decomposed into its continuous and discrete parts as $s = (s_c, s_d)$. Function reset checks if the controller has issued a discrete event signal, i.e., a new control action. This function is defined by:

$$\mathrm{reset}(x, u) = \begin{cases} \mathrm{reset}, & \text{if } u_d \neq \phi \\ x_d, & \text{otherwise} \end{cases}$$

The reset function signals integrators that a discontinuity has occurred and that they need to re-initialize integration related variables. The events created by other integrators do not require a re-initialization. Components v_1, x_1, v_1, x_2 are defined as 3rd degree asynchronous multistep integrators described in (Barros 2015a). These solvers adjust the stepsize in order to obtain the desired integration error.

Component U is the discrete sliding-mode controller described in Section 3.1. The control actions are computed by the input function F_U that calculates values according to (4) and (5). Since integrators v_1, x_1, v_1, x_2 are hybrid models that produce discrete event signals, the controller input function F_U ignores these values so the controller can keep a constant rate.

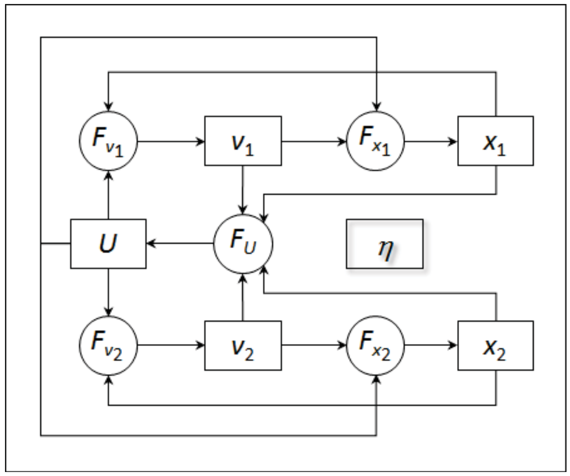
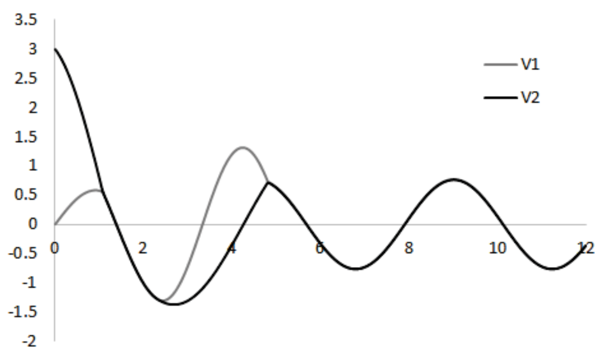
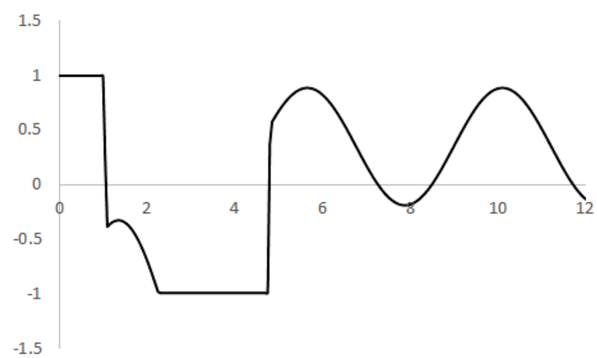


Figure 4: HyFlow component diagram of the 2-mass system.

We have simulated the system with parameters $M_1 = M_2 = 1$kg, $v_1(0) = 0, v_2(0) = 3$m/s, $F_c = 1$N, $K_1 = 3$N/m, $K_2 = 1$N/m. Controller sampling rate $T = 0.05$s. Mass velocities and control actions are represented in Figures 5(a) and 5(b), respectively.



(a) Velocities.



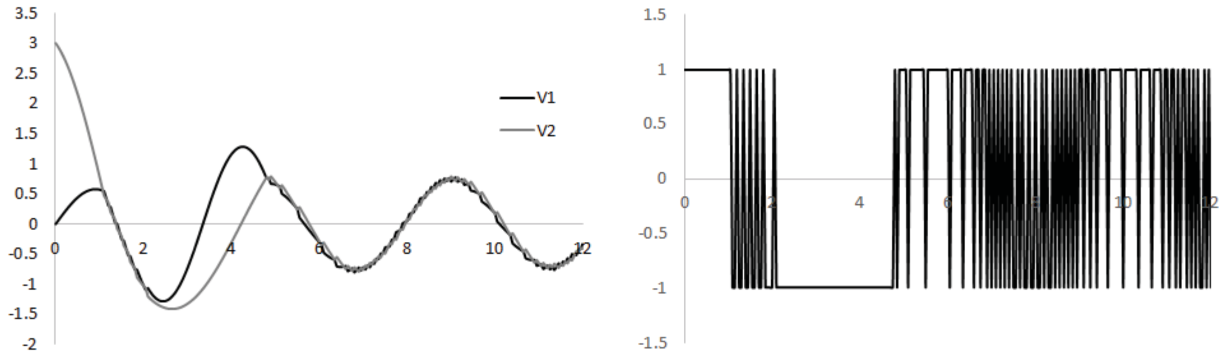
(b) Control acions.

Figure 5: Equivalent discrete control.

The system was also simulated with a chattering-based discrete controller and results are depicted in Figure 6. As expected, results are similar, supporting that both controllers converge for small sampling intervals. However, we can observe two main differences. First, velocities are not smooth as in the previous controller. Second, in this controller chattering is introduced, opposing to the previous controller, where control actions were smooth.

5 CHATTERING AVOIDANCE USING DYNAMIC TOPOLOGIES

Discrete event SMC can be used to avoid chattering in relay components, as described in the last section. However, some systems require different approaches to eliminate chattering. This can be the case when hard constrains need to be obeyed. In a tank, for example, the liquid volume can never be negative, and chattering occurs when tank volume is zero and tank input rate is smaller that the required output rate. While the discrete control can drive a system *close* to the sliding region, it cannot guarantee it will stay in that region. In the case of a tank this would mean that the volume could become negative. To avoid this



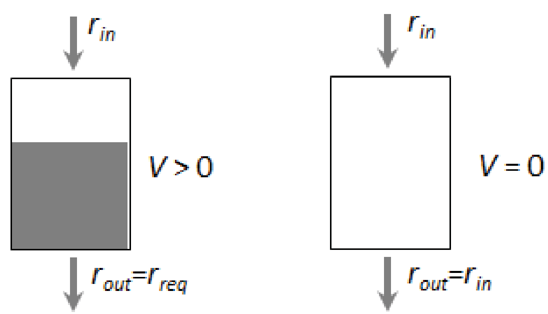
(a) Velocities.

(b) Control acions with chattering behavior.

Figure 6: Chattering-based control.

problem we will use the dynamic topology approach sketched in (Barros 2011). Instead of trying to follow the sliding surface $volume = 0$ we impose it by changing the model when the sliding surface is entered. The model is then switched back when the sliding surface is quit. In the tank system, the surface is reached when the volume is zero and it is exited when flow input rate is larger than the required output flow rate, and the liquid starts accumulating.

We consider the tank represented in Figure 7. It receives liquid with the rate $r_{in}(t)$ and it receives a request for delivering liquid with the rate $r_{req}(t)$. When the tank is not empty the output rate equals the required rate, as depicted in Figure 7.a). However, when the tank is empty, the output rate equals r_{in}, implying that the tanks remains empty while $r_{in}(t) < r_{req}(t)$.

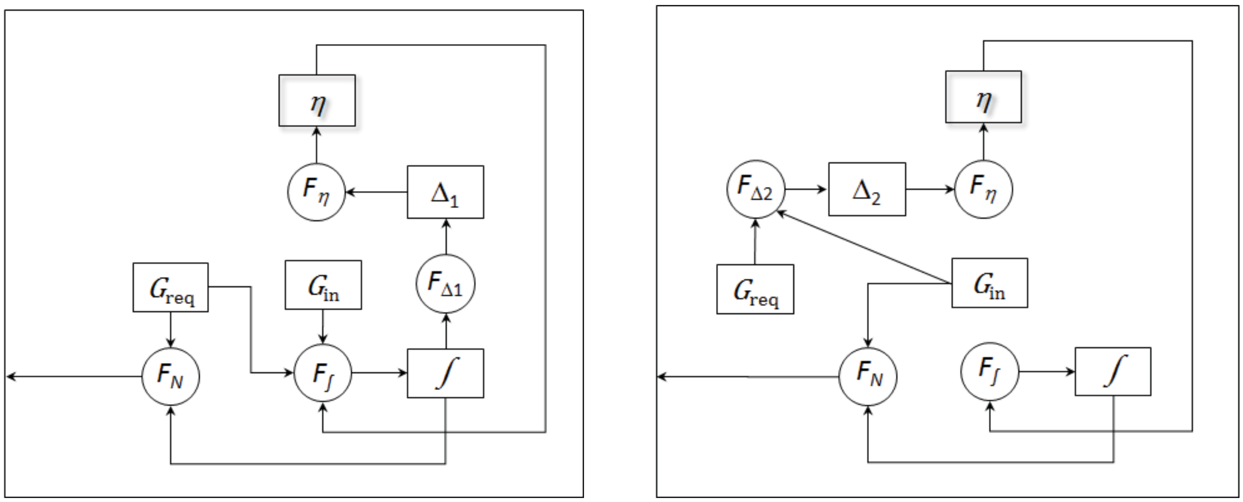


(a) Filled tank.

(b) Empty tank.

Figure 7: Tank modes and the corresponding output rates.

The tank system can be represented by the HyFlow component diagram of Figure 8.a). The output rate is $r_{req}(t)$, since tank liquid acts as a buffer, enabling an output rate larger that the input flow. Let us consider the case when $r_{in}(t) < r_{req}(t)$. After some time the tank becomes empty, as signaled by detector Δ_1, and the output rate cannot be $r_{req}(t)$ as desired, but it is reduced to $r_{in}(t)$, instead. However, if simulation model is kept, chattering will occur violating the hard constraint (tank) $volume \geq 0$. To avoid this problem we can switch the current model to the model of Figure 8.b). In this model the tank keeps a zero volume and the integrator $\int$ is stopped. When $r_{in} > r_{req}$ the executive is signaled by the detector Δ_2. The executive then switches back the model topology to Figure 8.a).



(a) Non-empty tank. (b) Empty tank.

Figure 8: Tank topologies.

We consider an input rate $r_{in}(t) = 3\cos(t) + 3$ and a required output rate $r_{req}(t) = 4\cos(0.5t) + 4$. Simulation results are depicted in Figure 9. When the volume reaches zero the output rate,$r_{out}(t)$ drops to the input flow rate $r_{in}(t)$. The output rate becomes equal to the requested rate when $r_{in}(t) > r_{req}(t)$ that originates an increase in tank volume.

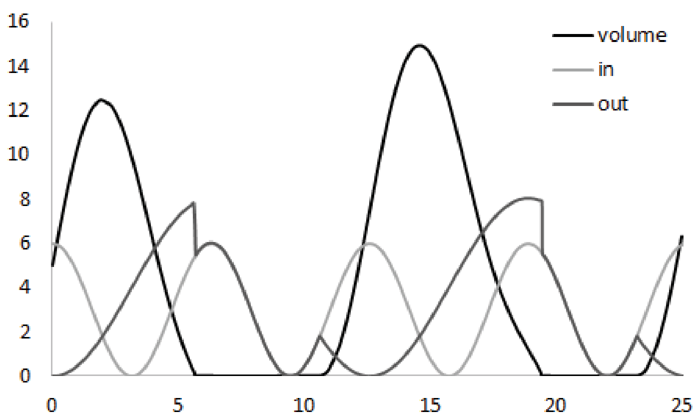


Figure 9: Tank volume, input and output rates.

The simulation becomes very efficient since the sliding region can be signaled with two events: $V = 0$ and $r_{in} > r_{req}$, entering and leaving the sliding region, respectively between these two events the numerical integration proceeds without any interruption or chattering effect.

6 CONCLUSION

Developing chattering-free simulation is a challenge for M&S methodologies. We have presented an HyFlow model of a discrete sliding model controller. This model can seamlessly be integrated with other HyFlow models expanding the set of models that can be used to describe complex systems. Some models exhibit a kind of chattering behavior that can be better handled by dynamic topology models. HyFlow provides full support for topology adaption, enabling chattering free simulation of an important class of models. In future work we plan to provide support for other kind of chattering behavior, specially those involving non-linear systems. Given recent developments in continuous SMC (Kamal, Moreno, Chalanga, Bandyopadhyay, and Fridman 2016), we also plan to develop HyFlow representation of these type of continuous controllers.

REFERENCES

Aljarbouh, A., and B. Caillaud. 2016. "Chattering-Free Simulation of Hybrid Dynamical Systems with the Functional Mock-Up Interface 2.0". In *The First Japanese Modelica Conferences*.

Barros, F. 2003. "Dynamic Structure Multiparadigm Modeling and Simulation". *ACM Transactions on Modeling and Computer Simulation* vol. 13 (3), pp. 259–275.

Barros, F. 2008. "Semantics of Dynamic Structure Event-based Systems". In *Proceedings of the Second International Conference on Distributed Event-based Systems*, pp. 245–252.

Barros, F. 2011. "On the Representation of Spatially Moving Entities Using Time-variant Topologies". In *Proceedings of the 2011 Summer Computer Simulation Conference*, pp. 145–152.

Barros, F. 2015a. "Asynchronous, Polynomial ODE Solvers Based on Error Estimation". In *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, pp. 115–121.

Barros, F. 2015b. "A modular representation of Fluid Stochastic Petri Nets". In *Symposium on Theory of Modeling and Simulation*.

Galias, Z., and X. Yu. 2008. "On Euler's Discretization of Sliding Mode Control Systems with Relative Degree Restriction". In *Modern Sliding Mode Control Theory: New Perspectives and Applications*, pp. 119–133, Springer.

Johansson, K. H., M. Egerstedt, J. Lygeros, and S. Sastry. 1999. "On the regularization of Zeno hybrid automata". *Systems & Control Letters* vol. 38, pp. 141–150.

Kamal, S., J. A. Moreno, A. Chalanga, B. Bandyopadhyay, and L. M. Fridman. 2016. "Continuous terminal sliding-mode controller". *Automatica* vol. 69, pp. 308–314.

Su, W.-C., S. V. Drakunov, and U. Ozguner. 2000. "An O(T^2) boundary layer in sliding mode for sampled-data systems". *IEEE Transactions on Automatic Control* vol. 45 (3), pp. 482–485.

Wang, B., X. Yu, and G. Chen. 2009. "ZOH Discretization Effect on Single-Input Mode Control Systems with Matched Uncertainties". *Automatica* vol. 48, pp. 118–125.

Zeigler, B., H. Praehofer, and T. Kim. 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press.

AUTHOR BIOGRAPHIES

FERNANDO J. BARROS is a Professor in the Department of Informatics Engineering at the University of Coimbra. He holds a PhD in Electrical Engineering from the University of Coimbra. His research interests include theory of modeling and simulation, hybrid systems and dynamic topology models. His email address is barros@dei.uc.pt.