Modeling and Simulation of Filippov System Models with Sliding Motions using Modelica

Mohammed Ahsan Adib Murad[1]   Luigi Vanfretti[2]   Federico Milano[1]

[1]School of Electrical Engineering, University College Dublin, Ireland, `mohammed.murad@ucdconnect.ie`, `federico.milano@ucd.ie`
[2]ECSE, Rensselaer Polytechnic Institute, Troy, NY, USA, `vanfrl@rpi.edu`

Abstract

This paper presents a generalized modeling formulation for implementation of dynamical system models that exhibit sliding behavior. The proposed formulation is based on Filippov theory, and is implemented using Modelica. The main advantage of the developed framework is that it effectively removes numerical chattering and trajectory deadlock. The robustness of the formulation is assessed considering three example system models: a stick-slip system, a relay feedback system and an anti-windup Proportional Integral (PI) controller in a power system application, i.e. an automatic voltage controller.

Keywords: Discontinuity, non-smooth, Filippov, hybrid dynamics, chattering.

1 Introduction

The Modelica Language is becoming an industry standard to model dynamical systems described by a set of Ordinary Differential Equations (ODEs) or Differential Algebraic Equations (DAEs) with mixed continuous and discrete variables, known as hybrid systems (Fritzson, 2014). A subclass of hybrid systems referred to as Filippov systems (Filippov, 1988) are those where discontinuities appear in vector fields, i.e. in the right hand side of the model's equations. If the solution of Filippov systems enters into a constrained subset of the state space, known as *sliding*, the formalism given by Filippov (Filippov, 1988) allows defining a vector field on the sliding surface to properly handle discontinuities.

A Modelica implementation of the Filippov systems without considering Filippov formalism leads to chattering-Zeno-type deadlocks, which consists of infinitely many instantaneous switches of the discrete variables during time domain simulation (Aljarbouh and Caillaud, 2016). This significantly restrains the performance of the solvers of Modelica simulation tools and can lead to a simulation halt. More importantly, such chattering does not represent the complicated real physical system chattering (Levant, 2010). Therefore a generalized formulation is required for smooth continuation of trajectories. This paper fills this gap.

The authors in (Aljarbouh and Caillaud, 2016) and (Aljarbouh et al., 2016) discussed chattering problem of Fil-

ippov systems in Modelica, using OpenModelica. However, to solve this problem, a framework based on the Functional Mock-up Interface standard (FMI) and Acumen (Taha et al., 2015) is proposed. The authors in (Suski and Pytlak, 2017) presented a race car model based on the Filippov formalism implemented in OpenModelica, but did not provide implementation details and a generalized formulation.

It is worth mentioning that there exist heuristic methods to solve the chattering issues (Bonilla et al., 2011). However, such methods lack a systematic generalization. Considering this, we propose a general purpose hybrid design for implementing Filippov systems using Modelica that allows smooth integration along chattering regions.

The main contributions of this paper are as follows:

- A generic formulation based on Filippov Theory (FT) for the implementation and direct numerical simulation of Filippov systems with one sliding surface using Modelica is proposed.

- A validation of the proposed formulation is performed comparing the results with a Matlab implementation and via simulation in two Modelica tools, namely OpenModelica and Dymola.

The remainder of the paper is organized as follows. Section 2 provides a background on FT. Section 3 presents the proposed generalized formulation for the implementation of Filippov system models using Modelica. Case studies are discussed in Section 4 where three examples are presented: a stick-slip system, a relay feedback system and an anti-windup PI controller in a power system voltage control application. Conclusions and future work directions are drawn in Section 5.

2 Filippov Systems

Filippov systems are dynamical system models with discontinuous right-hand side first-order ordinary differential equations (Filippov, 1988). Consider the following switched dynamical system of equations:

$$\dot{x} = f(x) = \begin{cases} f_1(x) & \text{when } h(x) < 0 \\ f_2(x) & \text{when } h(x) > 0 \end{cases} \quad (1)$$

where, the event function $h : \mathbb{R}^n \to \mathbb{R}$ and an initial condition $x(t_0) = x_0$ are known. The state space $\mathbb{R}^n$ is separated by a hyper-surface Σ into two regions R_1 and R_2 as follows:

$$
\begin{aligned}
R_1 &= \{x \in \mathbb{R}^n \mid h(x) < 0\}, \\
R_2 &= \{x \in \mathbb{R}^n \mid h(x) > 0\}, \\
\Sigma &= \{x \in \mathbb{R}^n \mid h(x) = 0\},
\end{aligned}
\tag{2}
$$

such that $\mathbb{R}^n = R_1 \cup \Sigma \cup R_2$, assuming at $x \in \Sigma$ the gradient of h never vanishes, i.e. $h_x(x) \neq 0$ for all $x \in \Sigma$.

The *Filippov convex method* (Filippov, 1988) states that the vector field on the surface of discontinuity is a convex combination of the two vector fields in the different regions of the state-space:

$$
\dot{x} = f(x) = \begin{cases} f_1(x), & x \in R_1 \\ \overline{co}\{f_1(x), f_2(x)\}, & x \in \Sigma \\ f_2(x), & x \in R_2 \end{cases}
\tag{3}
$$

where, $\overline{co}(f_1, f_2)$ is the minimal closed convex set containing f_1 and f_2, i.e.

$$
\overline{co}\{f_1, f_2\} = \{f_F : x \in \mathbb{R}^n \to \mathbb{R}^n : f_F = (1 - \alpha)f_1 + \alpha f_2\},
\tag{4}
$$

where $\alpha \in [0, 1]$.

Consider the trajectory starting at t_0 with $\dot{x} = f_1(x)$, with $x(t_0) = x_0$ reaches at Σ in finite time (t_k). Then at t_k the trajectory can cross, slide or exit Σ. In such situation, the first order theory given by Filippov explains how to solve these equations as summarized in the following section.

2.1 Filippov first order theory

Filippov first order theory defines the vector field if the solution approaches the discontinuous surface. Let $x \in \Sigma$ and $n(x)$ is the unit normal to Σ at x i.e. $n(x) = \frac{h_x(x)}{\|h_x(x)\|}$ where, $h_x(x) = \nabla h(x)$ and $\nabla = \frac{\partial}{\partial x}$; the components of $f_1(x)$ and $f_2(x)$ onto the normal to the Σ are $n^T(x)f_1(x)$ and $n^T(x)f_2(x)$ respectively.

2.1.1 Transversal crossing

If $x \in \Sigma$, then

$$
(n^T(x)f_1(x)).(n^T(x)f_2(x)) > 0,
\tag{5}
$$

i.e. the trajectory leaves Σ. The system will return to R_1 with $f = f_1$, if $n^T(x)f_1(x) < 0$ or it will proceed to R_2 with $f = f_2$ (see Fig. 1[I]), if $n^T(x)f_1(x) > 0$.
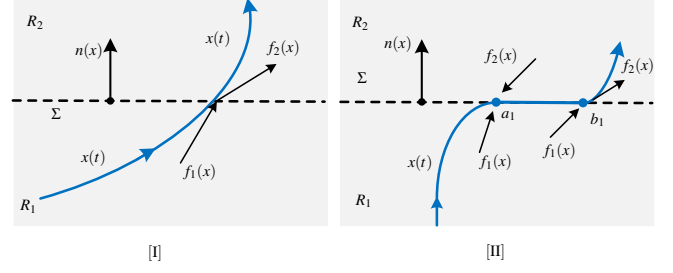
2.1.2 Sliding mode

Sliding occurs, at $x \in \Sigma$ if,

$$
(n^T(x)f_1(x)).(n^T(x)f_2(x)) < 0 .
\tag{6}
$$

An unique attracting sliding mode will occur if,

$$
(n^T(x)f_1(x)) > 0 \quad \text{and} \quad (n^T(x)f_2(x)) < 0, \; x \in \Sigma ,
\tag{7}
$$

Figure 1. Different regions of the state space with [I] transversal and [II] sliding trajectory.

and the solution does not leave Σ (see a_1 in Fig. 1[II]). During sliding the time derivative f_F is given by:

$$
f_F(x) = (1 - \alpha(x))f_1(x) + \alpha(x)f_2(x) ,
\tag{8}
$$

where, $\alpha(x)$ is given by [proof, see (Filippov, 1988)]:

$$
\alpha(x) = \frac{n^T(x)f_1(x)}{n^T(x)(f_1(x) - f_2(x))} .
\tag{9}
$$

If the signs are opposite in (7) a *repulsive sliding mode* will occur. In such a case, the solution is not unique, and thus, is not considered in this work.

2.1.3 Exit conditions

During sliding mode if one of the vector fields starts to point away, the solution continues above or below the sliding surface (see b_1 in Fig. 1[II]). The exit point is calculated by finding either the root $\alpha(x) = 0$ or $\alpha(x) = 1$ as appropriate. The following remarks are relevant:

- If $f_F(x) \neq f_1(x)$, $f_F(x) \neq f_2(x)$ such a solution is often called a *sliding motion*.

- If at the point of discontinuity, condition (6) becomes ≤ 0 and $f_1(x) \neq f_2(x)$ then a continuous vector-valued function $f_F(x)$ is given which determines the velocity of motion $\dot{x} = f_F(x)$ along the discontinuity line. If $n^T(x)f_1(x) = 0$ then $f_F(x) = f_1(x)$; if $n^T(x)f_2(x) = 0$ then $f_F(x) = f_2(x)$.

3 Filippov Theory Based Formulation

Filippov systems can be implemented in a computer language considering *event driven* or *time stepping* approaches (Dieci and Lopez, 2012). The former method simulates the system model by detecting the actual event time whereas the latter method without event detection (Piiroinen and Kuznetsov, 2008).

Modelica allows the conditional equations become activated or deactivated at the event instant (Fritzson, 2014) and the exact event instant can be detected during time domain simulation. Utilizing this Modelica language feature, in the following, we propose a formulation considering an accurate event detection method for the implementation of Filippov system models in Modelica.
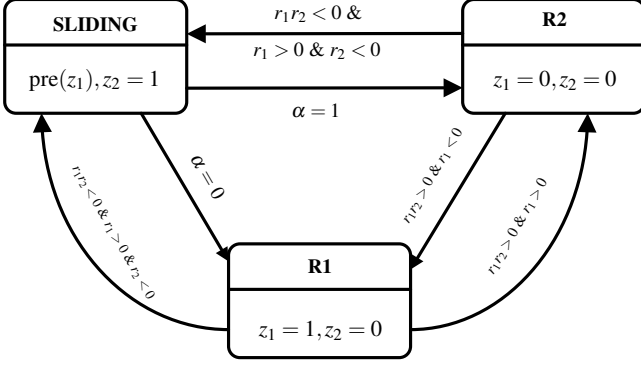
Figure 2. Generalized state transitions of Filippov systems.

Figure 3. Time derivative of state variables $(\dot{x}_1, \dot{x}_2)$ of stick-slip system without Filippov sliding simulated in Dymola.

3.1 General purpose design

According to FT, a system can have three states, the two states for $h(x) < 0$ (R1) and $h(x) > 0$ (R2), and a new state called SLIDING, characterized by $h(x) = 0$. For implementation using Modelica, it is convenient to introduce two discrete variables, say z_1 and z_2, into the differential equations, as follows:

$$\dot{x} = f_1(x)z_1(1-z_2) + f_2(x)(1-z_1)(1-z_2) + f_F(x)z_2 .$$

Observe that in the above equation, depending on the values of z_1 and z_2 (e.g. 1 or 0), a proper vector field needs to be activated during time domain simulation. Lets define $r_1 = h_x^T(x)f_1(x)$ and $r_2 = h_x^T(x)f_2(x)$ and Fig. 2 shows the changes in (z_1, z_2) for the three states and the conditions to move from one state to another. All these conditions are based on FT and are evaluated the moment at which the event function $(h(x))$ crosses zero. The sliding vector field $f_F(x)$ is derived explicitly according to (8). The exit conditions are defined based on (9). In particular, $\alpha(x) = 0$ and $\alpha(x) = 1$ are the conditions that indicate to move to the R1 and R2 regions, respectively.

The case studies below show the steps of the proposed approach followed during time domain simulation in Modelica tools.

4 Case Studies

In this section we discuss the implementation and validation of the Filippov systems in Modelica considering our generalized formulation. The case studies are posted online: `https://github.com/ALSETLab/Modelica_Fillipov_Sliding_Models`.

4.1 Example 1: Stick-slip system

Consider the two-dimensional system (so-called stick-slip system) (Dieci and Lopez, 2009, 2012)

$$\dot{x} = f(x) = \begin{cases} f_1(x) & \text{when } h(x) < 0 \\ f_2(x) & \text{when } h(x) > 0 \end{cases}$$
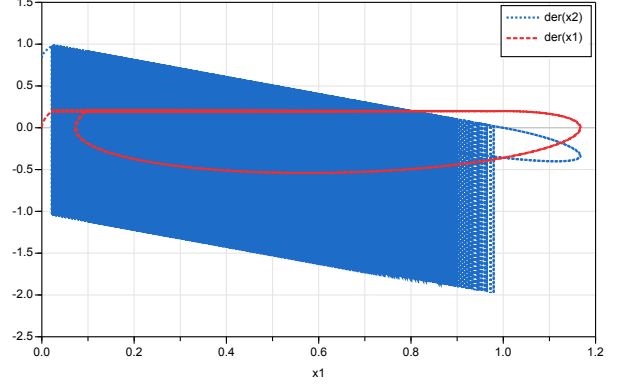
with

$$f_1(x) = \begin{pmatrix} x_2 \\ -x_1 + \frac{1}{1.2 - x_2} \end{pmatrix}, \qquad f_2(x) = \begin{pmatrix} x_2 \\ -x_1 - \frac{1}{0.8 + x_2} \end{pmatrix},$$

and $h(x) = x_2 - 0.2$. This system has a single switching manifold with two dynamic states.

4.2 Direct implementation

A direct implementation of this system using Modelica is as follows:

```modelica
model Stick_slip "Stick−Slip System"
  Real x1 (start=0);
  Real x2 (start=0);
  Integer z1 (start=1);
  Real h;
equation
  der(x1) = x2*z1 + x2*(1−z1);
  der(x2) = (−x1+(1/(1.2−x2)))*z1+(−x1−
    (1/(0.8+x2)))*(1−z1);
  h = x2 − 0.2;
  when h < 0 then
    z1 = 1;
  elsewhen h > 0 then
    z1 = 0;
  end when;
end Stick_slip;
```

OpenModelica and Dymola were used to simulate this example, however these tools halt when simulating this simple model. In OpenModelica, all solvers fail to simulate and report an error message: *Chattering detected around time 0.221654558425..0.221654756475 (100 state events in a row with a total time delta less than the step size 0.001).* On the other hand, Dymola's solver DASSL fails to continue the simulation. However some solvers for example: RkFix2 and Euler allows to continue simulation exposing chattering as shown in Fig. 3. Because of unnecessary chattering during the simulation, the results are not mathematically correct and it is not possible to understand the dynamic behavior of the real physical system.

4.2.1 Implementation using Filippov theory

The surface Σ is defined by zero of $h(x) = x_2 - 0.2$. Here, $h_x(x) = [\frac{\delta h(x)}{\delta x_1} \quad \frac{\delta h(x)}{\delta x_2}]^T = [0 \quad 1]^T$, thus on Σ (i.e. $x_2 =$

0.2), calculating,

$$r_1 = (0 \quad 1) \begin{pmatrix} x_2 \\ -x_1 + \frac{1}{1.2 - x_2} \end{pmatrix} = -x_1 + 1 \,,$$

$$r_2 = (0 \quad 1) \begin{pmatrix} x_2 \\ -x_1 - \frac{1}{0.8 + x_2} \end{pmatrix} = -x_1 - 1 \,.$$

Therefore according to (6), $x_1 \in (-1, 1)$ there will be an attractive sliding mode on Σ. This sliding vector field on Σ is calculated using equations (8,9):

$$\alpha(x) = \frac{n^T(x) f_1(x)}{n^T(x)(f_1(x) - f_2(x))} = \frac{-x_1 + 1}{2} \,,$$

$$f_F(x) = \begin{pmatrix} x_2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0 \end{pmatrix} \,,$$

which means that on the sliding surface Σ the x_1 state will grow linearly until reaching the value $x_1 = 1$, at which the point the trajectory will leave Σ with vector field f_1 as for $x_1 = 1$ the $\alpha(x) = 0$.

Using the proposed FT based approach, the expressions derived above can be used to implement the model as follows (equation part of the model is given):

```
der(x1) = x2*z1*(1-z2) + x2*(1-z1)*(1-z2)
    + x2*z2;
der(x2) = (-x1+(1/(1.2-x2)))*z1*(1-z2) +
    (-x1-(1/(0.8+x2)))*(1-z1)*(1-z2);
h = x2 - 0.2;
r1 = -x1+1;
r2 = -x1-1;
a = (-x1+1)/2;
zeroCrossing.u = h;
zeroCrossing1.u = (-x1+1)/2;
zeroCrossing2.u = 1-a;
when zeroCrossing.y then
    if r1*r2 < 0 then
      if r1 > 0 and r2 < 0 then
      z1 = pre(z1);
      z2 = 1;
      else
      z1 = pre(z1);
      z2 = 0;
      end if;
    elseif r1*r2 > 0 then
      if r1 < 0 then
      z1 = 1;
      z2 = 0;
      elseif r1 > 0 then
      z1 = 0;
      z2 = 0;
      else
      z1 = pre(z1);
      z2 = 0;
      end if;
    else
      z1 = pre(z1);
      z2 = 0;
    end if;
elsewhen zeroCrossing1.y and pre(z2) == 1
    then
        z1 = 1;
        z2 = 0;
```
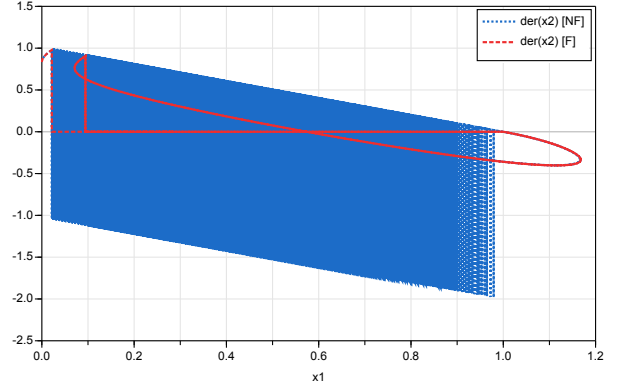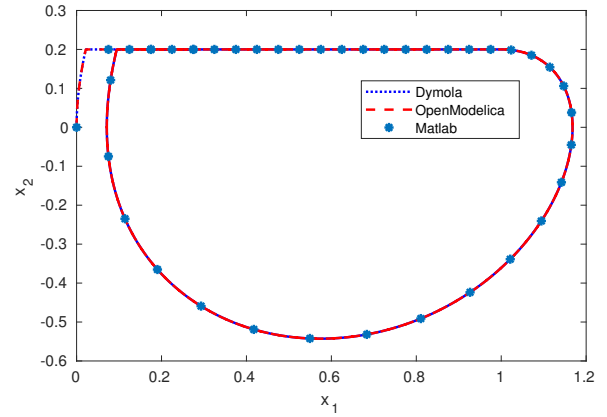

Figure 4. Time derivative of state variable ($\dot{x}_1$) of stick-slip system without (NF) and with (F) Filippov theory simulated in Dymola.

Figure 5. Periodic trajectories of the stick-slip system obtained in different simulation software tools.

```
elsewhen zeroCrossing2.y and pre(z2) == 1
    then
        z1 = 0;
        z2 = 0;
  end when;
```

Using this implementation the simulation of this system can be successfully carried out in both OpenModelica and Dymola without numerical issues. Results are compared in Fig. 4. Observe that considering the Filippov sliding condition the simulation continues without any chattering. For further validation of the proposed generalized formulation and its implementation, the trajectories in Fig. 5 obtained using Modelica tools (OpenModelica and Dymola) and compared with the results obtained using Matlab utilizing the method in (Piiroinen and Kuznetsov, 2008).

4.3 Example 2: A relay feedback system

A relay feedback system with single-input and single-output is as follows (Piiroinen and Kuznetsov, 2008):

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \\ u &= -\text{sgn}(y) \end{aligned} \quad (10)$$

or

$$\dot{x} = \begin{cases} Ax + B, & \text{when } Cx < 0 \\ Ax - B, & \text{when } Cx > 0 \end{cases} \tag{11}$$

where,

$$A = \begin{pmatrix} -(2\zeta\omega+1) & 1 & 0 \\ -(2\zeta\omega+\omega^2) & 0 & 1 \\ -\omega^2 & 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 1 \\ -2\sigma \\ 1 \end{pmatrix}, C = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

The state vector of this model is $x = [x_1, x_2, x_3]^T$ and the discontinuity surface Σ is defined by $h(x) = x_1$. Re-writing the dynamical system according to FT,

$$\dot{x} = f(x) = \begin{cases} f_1(x) & \text{when } h(x) < 0 \\ f_2(x) & \text{when } h(x) > 0 \end{cases} \tag{12}$$

with

$$f_1(x) = \begin{pmatrix} -(2\zeta\omega+1)x_1 + x_2 + 1 \\ -(2\zeta\omega+\omega^2)x_1 + x_3 - 2\sigma \\ -\omega^2 x_1 + 1 \end{pmatrix},$$

$$f_2(x) = \begin{pmatrix} -(2\zeta\omega+1)x_1 + x_2 - 1 \\ -(2\zeta\omega+\omega^2)x_1 + x_3 + 2\sigma \\ -\omega^2 x_1 - 1 \end{pmatrix}.$$

Here, $h_x(x) = [1 \quad 0 \quad 0]^T$, thus on Σ (i.e. $x_1 = 0$), calculating,

$$r_1 = -(2\zeta\omega+1)x_1 + x_2 + 1,$$

$$r_2 = -(2\zeta\omega+1)x_1 + x_2 - 1.$$

The sliding vector field on Σ obtained using equations (8,9):

$$\alpha(x) = (-(2\zeta\omega+1)x_1 + x_2 + 1)/2,$$

$$f_F(x) = \begin{pmatrix} 0 \\ b + 4(-(2\zeta\omega+1)x_1 + x_2 + 1)/2 \\ c - 2(-(2\zeta\omega+1)x_1 + x_2 + 1)/2 \end{pmatrix},$$

where $b = -(2\zeta\omega+\omega^2)x_1 + x_3 - 2\sigma$ and $c = -\omega^2 x_1 + 1$.

Similarly to the previous example, we have implemented both in direct form and using the proposed formulation. The parameters considered are: $\zeta = 0.05, \omega = 25$ and $\sigma = -1$. Using the Modelica language similar issues (chattering and trajectory deadlock) as in the previous example arise in OpenModelica and Dymola during time domain simulation without considering FT and the simulation results obtained using Dymola are shown in Figs. 6-7. Observe that chattering does not occur for the model implemented following the proposed approach based on FT. Finally the validation of the results of Modelica tools against the implementation in Matlab (Piiroinen and Kuznetsov, 2008) is shown in Fig. 8.
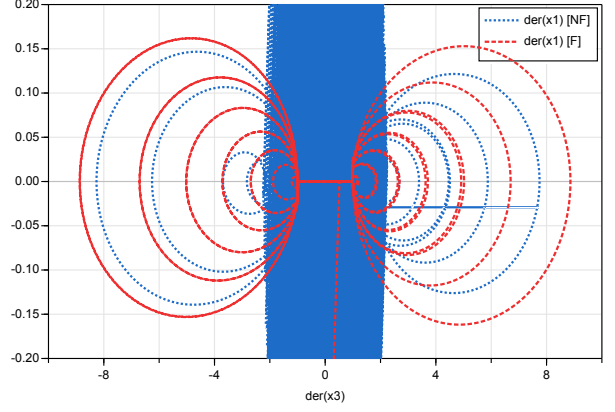
Figure 6. Time derivative of state variable ($\dot{x}_1$) of the relay feedback system model without (NF) and with (F) Filippov theory simulated in Dymola.
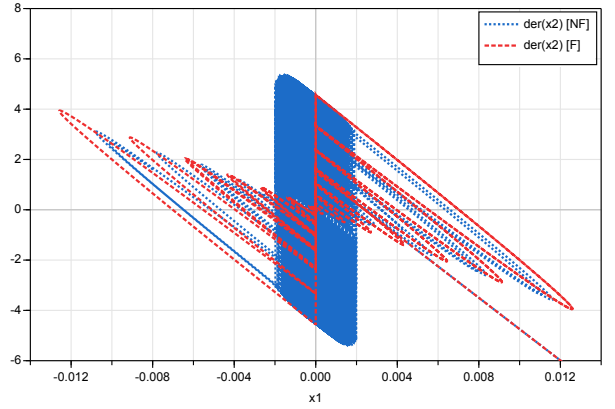


Figure 7. Time derivative of state variable ($\dot{x}_2$) of the relay feedback system model without (NF) and with (F) Filippov theory simulated in Dymola.
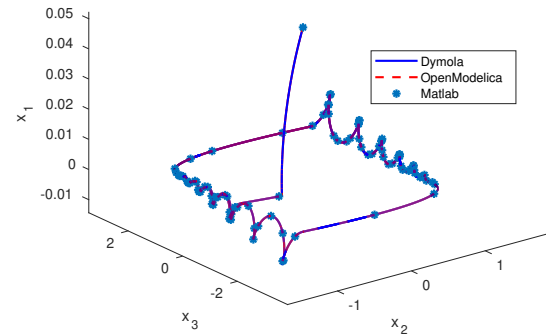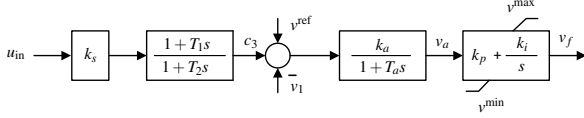

Figure 8. State space response for the relay feedback system obtained in different simulation software tools.

4.4 Example 3: Anti-windup PI controller

The IEEE Standard 421.5-2016 recommends an anti-windup (AW) or non-windup PI controller (IEEE, 2016) model for dynamic analysis of power systems. Mathemat-
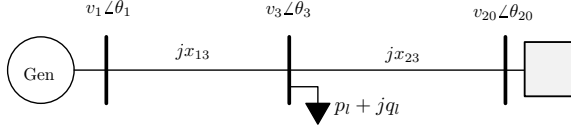
Figure 9. A single generator connected to an infinite bus.

Figure 10. Scheme of AVR and power system stabilizer.

ically, the model is:

If $y \geq w^{\text{max}}$: $w = w^{\text{max}}$ and $\dot{x} = 0$,

If $y \leq w^{\text{min}}$: $w = w^{\text{min}}$ and $\dot{x} = 0$, $\qquad$ (13)

Otherwise : $w = y = k_p u + x$ and $\dot{x} = k_i u$.

where k_p, k_i, are the proportional, integral gains of the controller, respectively; x is the controller's state variable; and u, y and w are the control input, unconstrained output and constrained output respectively.

The discontinuous nature of the integral state variable of this model can lead to numerical issues such as trajectory deadlock and chattering. Only *ad hoc* approaches have been proposed to handle such issues (Hiskens, 2012; Fabozzi et al., 2017). In the authors preliminary results, it was shown that the formalism given by Filippov can effectively remove those numerical issues (Murad et al., 2019). In this work, the AW PI controller is utilized in a power system voltage control application namely an Automatic Voltage Controller (AVR), which is implemented using the proposed formulation in Modelica. In addition, the dynamic response is compared to a deadband (DB) based technique proposed in (Hiskens, 2012) and it is shown that the proposed approach achieves a smooth transient response.

4.4.1 Single machine infinite bus system

Consider a simple three bus power system network with a single machine shown in Fig. 9. The generator in bus 1 is equipped with an AW PI controller based AVR and a power system stabilizer as depicted in Fig. 10. The dynamics of this Single Machine Infinite Bus (SMIB) system is described by a set of DAEs in the following form (Milano, 2010),

$$\dot{x} = f(x,y) \ , \\ 0 = g(x,y) \ , \qquad (14)$$

where x and y are the vector of state and algebraic variables respectively.

Then generator in Fig. 9 (Gen) is modelled using a third order model (Milano, 2010), and the switching manifold for the maximum of the AVR, $h(x) = k_p v_a + x_i - v^{\text{max}}$.

When $h(x) < 0$, the differential equations of the SMIB system are:

$$\dot{\delta} = \omega \qquad (15)$$

$$\dot{\omega} = \frac{1}{M}(p_m - p_e - D\omega) \qquad (16)$$

$$\dot{e}'_q = \frac{1}{T'_{d0}}(v_f - \frac{x_d}{x'_d}e'_q + \frac{x_d - x'_d}{x'_d}v_1\cos(\delta - \theta_1)) \qquad (17)$$

$$\dot{v}_a = (k_a(v^{\text{ref}} + c_3 - v_1) - v_a)/T_a \qquad (18)$$

$$\dot{x}_i = k_i v_a \qquad (19)$$

$$\dot{s}_1 = \frac{1}{T_2}(c_2 - s_1) \ . \qquad (20)$$

The notations and algebraic equations are given in (Murad et al., 2019). When the v_f reaches to its max (v^{max}) then (17) and (19) will be switched and all other state variables will remain same, as follows:

$$\dot{e}'_q = \frac{1}{T'_{d0}}(v^{\text{max}} - \frac{x_d}{x'_d}e'_q + \frac{x_d - x'_d}{x'_d}v_1\cos(\delta - \theta_1)) \qquad (21)$$

$$\dot{x}_i = 0 \qquad (22)$$

4.4.2 Implementation using Filippov theory

We consider $f_1(x,y)$ is (15)-(20) and $f_2(x,y)$ is (15), (16), (21), (18), (22) and (20). Calculating, $h_x(x) = \begin{bmatrix} \frac{\partial h(x)}{\partial x_1} & \frac{\partial h(x)}{\partial x_2} & \dots & \frac{\partial h(x)}{\partial x_6} \end{bmatrix}^T = [0\ 0\ 0\ k_p\ 1\ 0]^T$, and the normal to the switching surface is: $n^T(x) = [0\ 0\ 0\ k_p\ 1\ 0]$. On the switching manifold, calculating

$$h_x^T(x)f_1(x,y) = k_p((k_a(v^{\text{ref}} + c_3 - v_1) - v_a)/T_a\ ) + k_i v_a \ ,$$
$$h_x^T(x)f_2(x,y) = k_p((k_a(v^{\text{ref}} + c_3 - v_1) - v_a)/T_a\ ) \ .$$

Therefore according to (6), if an attractive sliding occurs on Σ, then using equation (8) $\alpha(x,y)$ is given by:

$$\alpha(x,y) = \frac{k_p((k_a(v^{\text{ref}} + c_3 - v_1) - v_a)/T_a\ ) + k_i v_a}{k_i v_a} \ .$$

According to (9), during the sliding requires (22):

$$f_F(x,y) = -k_p((k_a(v^{\text{ref}} + c_3 - v_1) - v_a)/T_a\ ) \ .$$

These expressions are used in the Modelica implementation.

4.4.3 Simulation Results

The SMIB system is implemented considering the FT-based formulation and the DB based method in Modelica. The DB implementation is the same as in (Hiskens, 2012) and DB value used is 0.0001. The parameters of different components of the SMIB system are given in Table 1 and initial values for all variables are given in (Murad et al., 2019).

The SMIB system is simulated by increasing the voltage reference set-point ($v^{\text{ref}} = 1.01$) and load ($p_{l0} = 0.71$

Table 1. PARAMETERS OF THE SMIB NETWORK

Name	Values
Generator	$M = 8$, $D = 0$, $x'_d = 0.25$, $x_d = 1$, $p_m = 1$, $T'_{d0} = 6$
Line	$x_{13} = 0.3$, $x_{23} = 0.5$
Load	$p_{l0} = 0.7$, $q_{l0} = 0.01$
AVR	$k_a = 2$, $T_a = 0.005$, $k_p = 5.5$, $k_i = 35$, $v^{\max} = 1.6$, $v^{\min} = -1.5$, $v^{\mathrm{ref}} = 1$
PSS	$k_s = 1.5$, $T_1 = 0.23$, $T_2 = 0.12$

pu, $q_{l0} = 0.016$ pu) at $t = 5$ s. Figs. 11 and 12 show the response of the time derivative of integrator state variable ($\dot{x}_i$) and field voltage (v_f) for both DB and FT based method's respectively. Following the disturbance, the integrator state variable (x_i) enters into a deadlock region and using the DB based implementation it shows chattering. Therefore the field voltage (v_f) shows numerous switching bounded by the DB (see zoom in Fig. 12). However, a smooth response is achieved using FT based method. In addition, except for the chattering the FT based model shows the same trajectories. It is important to mention that without DB or FT based techniques simulation in OpenModelica fails for all solvers while DASSL fails for Dymola.
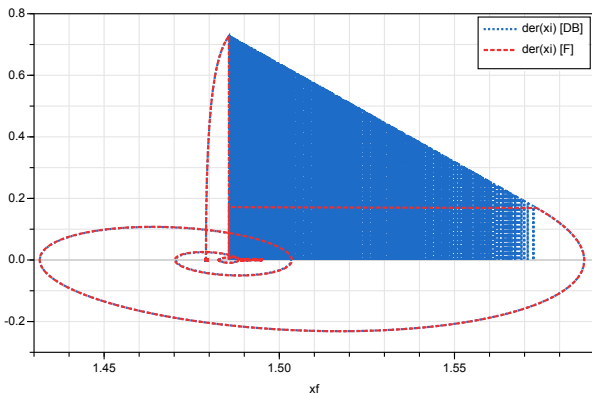

Figure 11. Trajectories of time derivative of the integrator state variable (x_i) in the AW PI controller using DB and Filippov (F) methods simulated in Dymola.

5 Conclusions

A generic formulation to implement Filippov system models with sliding motion using Modelica is proposed. Three examples are presented considering such a general-purpose design with a single sliding surface. Simulation results in different Modelica tools indicate accurate dynamic response without any chattering or simulation halt.

Future work will extend the FT-based design for multiple discontinuity surface (Piiroinen and Kuznetsov, 2008). In addition a numerical performance of FT-based AW PI controller will be investigated considering the Modelica power system library: OpenIPSL (Vanfretti et al., 2016).
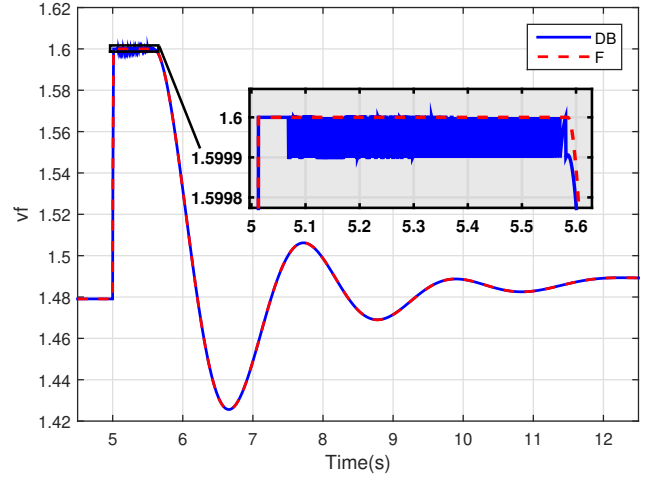
Figure 12. Trajectories of the field voltage (v_f) using DB and Filippov (F) methods simulated in Dymola.

6 Acknowledgment

References

Ayman Aljarbouh and Benoit Caillaud. Chattering-free simulation of hybrid dynamical systems with the function mock-up interface 2.0. In *The First Japanese Modelica Conferences, May 23-24, Tokyo, Japan*, number 124, pages 95–105. Linköping University Electronic Press, 2016.

Ayman Aljarbouh, Yingfu Zeng, Adam Duracz, Benoit Caillaud, and Walid Taha. Chattering-free simulation for hybrid dynamical systems semantics and prototype implementation. In *2016 IEEE Intl. Conf. on Comp. Science and Eng. (CSE) and IEEE Intl Conf. on Embedded and Ubiquitous Computing (EUC)*, pages 412–422. IEEE, 2016.

J. Bonilla, L.J. Yebra, and S. Dormido. A heuristic method to minimise the chattering problem in dynamic mathematical two-phase flow models. *Mathematical and Computer Modelling*, 54(5):1549 – 1560, 2011. ISSN 0895-7177.

Luca Dieci and Luciano Lopez. Sliding motion in Filippov differential systems: Theoretical results and a computational approach. *SIAM Journal on Numerical Analysis*, 47(3):2023–2051, 2009.

Luca Dieci and Luciano Lopez. A survey of numerical methods for IVPs of ODEs with discontinuous right-hand side. *Jour-*

nal of Computational and Applied Mathematics, 236(16): 3967 – 3991, 2012. ISSN 0377-0427.

Davide Fabozzi, Stefan Weigel, Bernd Weise, and Fortunato Villella. Semi-implicit formulation of proportional-integral controller block with non-windup limiter according to IEEE Standard 421.5-2016. In *Bulk Power Systems Dynamics and Control Symposium (IREP)*, pages 1–7, 2017.

A. F. Filippov. *Differential Equations with Discontinuous Right-hand Sides*. Kluwer Academic Publishers, 1988.

FMI. Functional mockup interface, [Online]. URL `https://fmi-standard.org/`.

Peter Fritzson. *Principles of object-oriented modeling and simulation with Modelica 3.3: a cyber-physical approach*. John Wiley & Sons, 2014.

I. A. Hiskens. Dynamics of type-3 wind turbine generator models. *IEEE Transactions on Power Systems*, 27(1):465–474, Feb 2012. ISSN 0885-8950.

IEEE. IEEE recommended practice for excitation system models for power system stability studies. *IEEE Std 421.5-2016 (Revision of IEEE Std 421.5-2005)*, pages 1–207, Aug 2016.

Arie Levant. Chattering analysis. *IEEE transactions on automatic control*, 55(6):1380–1389, 2010.

F. Milano. *Power System Modelling and Scripting*. Power Systems. Springer Berlin Heidelberg, 2010. ISBN 9783642136689.

M. A. A. Murad, B. Hayes, and F. Milano. Application of Filippov theory to the IEEE standard 421.5-2016 anti-windup PI controller. In *2019 IEEE Milan PowerTech*, pages 1–6, June 2019.

Petri T. Piiroinen and Yuri A. Kuznetsov. An event-driven method to simulate Filippov systems with accurate computing of sliding motions. *ACM Trans. Math. Softw.*, 34(3):13:1–13:24, May 2008. ISSN 0098-3500.

Damian Suski and Radosław Pytlak. Simulation of hybrid systems with sliding modes. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pages 419–424. IEEE, 2017.

Walid Taha, Adam Duracz, Yingfu Zeng, Kevin Atkinson, Ferenc A Bartha, Paul Brauner, Jan Duracz, Fei Xu, Robert Cartwright, Michal Konečný, et al. Acumen: An open-source testbed for cyber-physical systems research. In *International Internet of Things Summit*, pages 118–130. Springer, 2015.

L. Vanfretti, T. Rabuzin, M. Baudette, and M. Murad. itesla power systems library (iPSL): A Modelica library for phasor time-domain simulations. *SoftwareX*, 5:84 – 88, 2016. ISSN 2352-7110.