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

Mohammed Ahsan Adib Murad, Luigi Vanfretti,
Federico Milano

September 22-24, 2020

Outline

- Motivation
- Objectives
- Filippov Theory and Proposed Modeling Formulation
- Case Studies
- Conclusions

Motivation

- Modelica is an industry standard to model hybrid (discrete-continuous) dynamical systems described by ODEs or DAEs.
- A subclass of hybrid systems referred to as **Filippov systems** where discontinuities appear in the right hand side of the model's equations.
- 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.
- This significantly restrains the performance of the solvers of Modelica simulation tools and can lead to a simulation halt.
- A generalized formulation is required for smooth continuation of trajectories of Fillippov system models in Modelica tools.

Objectives

- To propose a generic formulation based on Filippov Theory (FT) for the implementation and direct numerical simulation of Filippov systems with one sliding surface using Modelica.
- To validate the proposed formulation comparing the results with a Matlab implementation and via simulation in two Modelica tools, namely OpenModelica and Dymola.

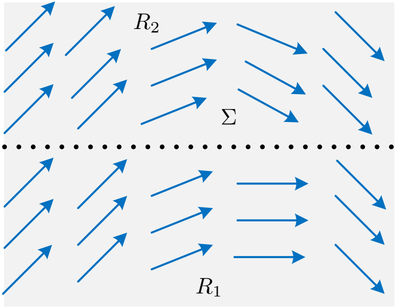
Filippov Theory

- Consider the following switched dynamical system:

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

- The state space $\mathbb{R}^n$ is split into two regions R_1 and R_2 separated by a hyper-surface Σ. Where,

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



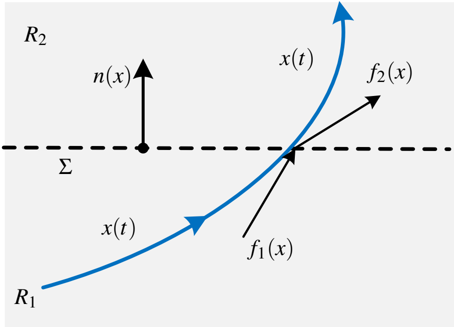
- *Filippov convex method:* the vector field on the surface of discontinuity is a convex combination of the vector fields in the different regions of the state space.

Filippov First Order Theory

- Filippov first order theory defines the vector field if the solution approaches the discontinuous surface.
- The trajectory of $\dot{\boldsymbol{x}} = \boldsymbol{f_1(x)},$ with $\boldsymbol{x(0)} = \boldsymbol{x_0}$ reaches at Σ in finite time.
- Let $\boldsymbol{n(x)}$ is the unit normal to Σ at $\boldsymbol{x}$ i.e. $\boldsymbol{n(x)} = \frac{\boldsymbol{h_x(x)}}{\|\boldsymbol{h_x(x)}\|}$ where, $\boldsymbol{h_x(x)} = \nabla h(\boldsymbol{x})$ and $\nabla = \frac{\partial}{\partial \boldsymbol{x}}$.
- **Transversal Crossing:** If at $\boldsymbol{x} \in \Sigma$,

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

  leave Σ:
  - if $\boldsymbol{n}^T\boldsymbol{(x)}\boldsymbol{f_1(x)} > 0$, to R_2,
  - if $\boldsymbol{n}^T\boldsymbol{(x)}\boldsymbol{f_1(x)} < 0$, to R_1.



[I]

Filippov Theory: Sliding

- **Sliding**:

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

  - Attracting: Have existence and uniqueness (a_1 in Fig. [II]):

$$(\boldsymbol{n}^T(\boldsymbol{x})\boldsymbol{f_1}(\boldsymbol{x})) > 0 \text{ and } (\boldsymbol{n}^T(\boldsymbol{x})\boldsymbol{f_2}(\boldsymbol{x})) < 0$$

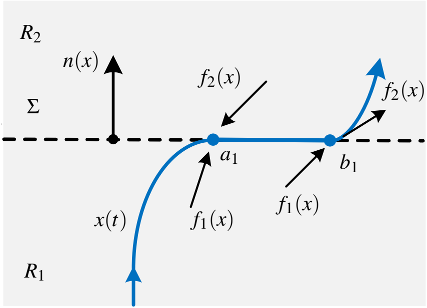
  - Repulsive: No uniqueness. Not covered.

- **Filippov vector field**: While sliding along Σ, time derivative $\boldsymbol{f}_F$ is given by,

$$\boldsymbol{f}_F(\boldsymbol{x}) = (1-\alpha(\boldsymbol{x}))\boldsymbol{f_1}(\boldsymbol{x}) + \alpha(\boldsymbol{x})\boldsymbol{f_2}(\boldsymbol{x}).$$

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

- Exit: during sliding, if one of the vector fields starts to point away, the solution continues above or below the sliding surface.



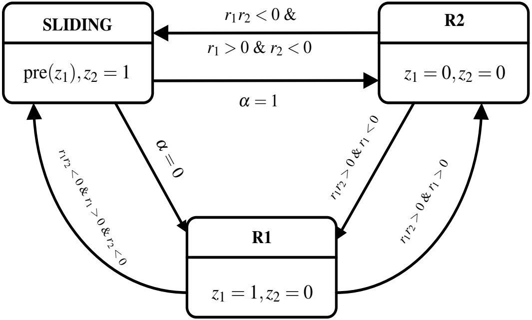
[II]

Proposed Modeling Formulation

- According to FT, a system can have three states: $h(x) < 0$ (R1). $h(x) > 0$ (R2), and $h(x) = 0$ (SLIDING).
- To implement in Modelica, we introduce two discrete variables: say z_1 and z_2, into the differential equations, as follows:

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

- Depending on the values of z_1 and z_2 (e.g. 1 or 0), a proper vector field is activated.
- In the SLIDING state the value of $z_2 = 1$.
- SLIDING deactivates both $f_1(x)$ and $f_2(x)$, without the need of changing the value of z_1. So the previous value ($\mathrm{pre}(z_1)$) is retained.



Case Study I: Stick-slip System

- Consider the two-dimensional Stick-slip system:

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

where $h(x) = x_2 - 0.2$ with

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

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

- Simulation issues are observed with a direct implementation of this system using Modelica in OpenModelica and Dymola.
- **OpenModelica:** *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).*
- **Dymola:** DASSL fails to continue the simulation. However RkFix2 and Euler allows to continue exposing chattering.
- Due to chattering during the simulation, the results are not mathematically accurate.
- It is not possible to understand the dynamic behavior of the real physical system.

Case Study I: Stick-slip System

- Using this FT based implementation the simulation of this system can be successfully carried out in both OpenModelica and Dymola without numerical issues.

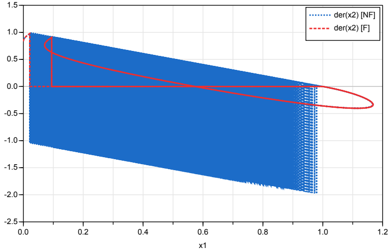


Figure 1: Time derivative of state variable ($\dot{x}_1$) of stick-slip system without (NF) and with (F) Filippov theory simulated in Dymola.

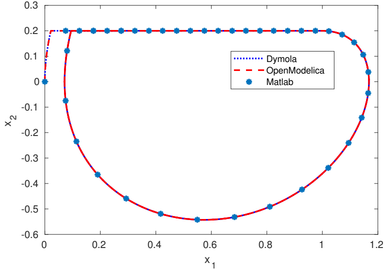


Figure 2: Periodic trajectories of the stick-slip system obtained in different simulation software tools.

Case Study II: A Realy Feedback System

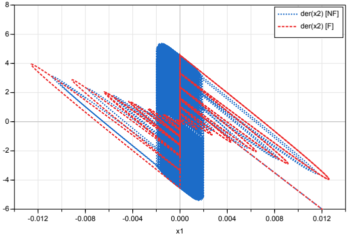
- A relay feedback system with single-input and single-output is as follows:

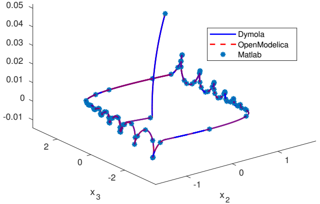
$$\dot{x} = f(x) = \begin{cases} f_1(x) & \text{when } h(x) < 0 \\ f_2(x) & \text{when } h(x) > 0 \end{cases}$$

where $h(x) = x_1$ with

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

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





Case Study III: Anti-windup PI in an SMIB

- DAEs:

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

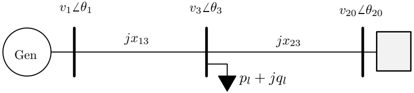
- $h(\boldsymbol{x}) = k_p v_a + x_i - v^{\max}$

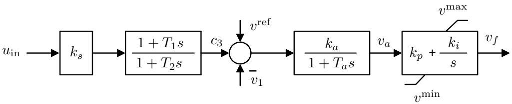
- $\boldsymbol{f_1}(\boldsymbol{x}, \boldsymbol{y})$:

$$\dot{\delta} = \omega$$
$$\dot{\omega} = \frac{1}{M}(p_m - p_e - D\omega)$$
$$\dot{e}'_q = \frac{1}{T'_{d0}}\left(v_f - \frac{x_d}{x'_d}e'_q + \frac{x_d - x'_d}{x'_d}v_1\cos(\delta - \theta_1)\right)$$
$$\dot{v}_a = (k_a(v^{\mathrm{ref}} + c_3 - v_1) - v_a)/T_a$$
$$\dot{x}_i = k_i v_a$$
$$\dot{s}_1 = \frac{1}{T_2}(c_2 - s_1)$$

- $\boldsymbol{f_2}(\boldsymbol{x}, \boldsymbol{y})$:

$$\dot{e}'_q = \frac{1}{T'_{d0}}\left(v^{\max} - \frac{x_d}{x'_d}e'_q + \frac{x_d - x'_d}{x'_d}v_1\cos(\delta - \theta_1)\right)$$
$$\dot{x}_i = 0$$





Case Study III: Implementation Using Filippov Theory

- Calculating, $h_x(x) = [\frac{\partial h(x)}{\partial x_1} \quad \frac{\partial h(x)}{\partial x_2} \; ... \; \frac{\partial h(x)}{\partial x_6}]^T = [0\ 0\ 0\ k_p\ 1\ 0]^T$.
- On the switching manifold,

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

- If an attractive sliding occurs on Σ, then $\alpha(x,y)$ is given by:

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

- Thus during the sliding:

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

- These expressions are used in the Modelica implementation.

Case Study III: Results of SMIB System

- The SMIB system is implemented considering the FT-based formulation and a deadband based method in Modelica.
- Disturbance: step change in the voltage reference set-point ($v^{\text{ref}} = 1.01$) and load ($p_{l0} = 0.71$ pu, $q_{l0} = 0.016$ pu) at $t = 5$ s.

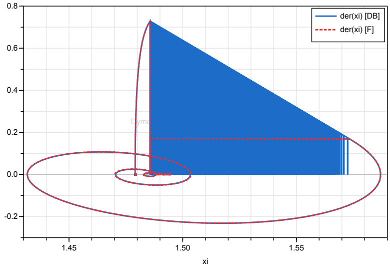


Figure 3: Time derivative of the integrator state variable ($\dot{x}_i$) in the AW PI controller with respect to the state variable (x_i) using DB and Filippov (F) methods simulated in Dymola.

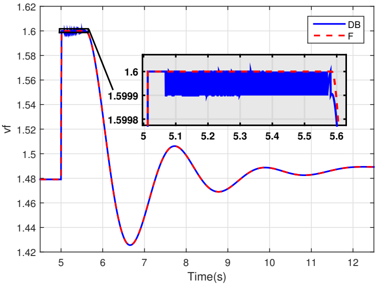


Figure 4: Trajectories of the field voltage (v_f) using DB and Filippov (F) methods simulated in Dymola.

Conclusions and Future Work

- 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 and implementation details are given.
- Simulation results in different Modelica tools indicate accurate dynamic response without any chattering or simulation halt.
- **Future Work:**
  - Future work will extend the FT-based design for multiple discontinuity surface.
  - Study the advantages from computational point of view.

The case studies are posted on-line!

`https://github.com/ALSETLab/Modelica_Fillipov_Sliding_Models`

Thank you!