Out: March 16, 2020
Due: April 26, 2020

NTNU – Trondheim
Norwegian University of Science and Technology

TTK4130 Modeling and Simulation

Assignment 11

Introduction

Systems that include complex nonlinear phenomena such as hysteresis, dead-zones, saturation or friction, often lead to dynamics that switch between partial models, and lead to discontinuous dynamics. Moreover, the transition to one partial model to another does not only take place at specific state values, but could also require that the state values are varying in a particular direction.

In such situations, it is crucial to determine the exact moment when transitions between partial models happen. This is known as event detection.

In this assignment, we will study event detection and apply it to simulate different friction models.

Problem 1 (Bouncing Ball)

Let us consider a ball that bounces on a flat surface.

The ball is modeled as a point with mass $m = 1\,\text{kg}$ that is subjected to both gravity and viscous friction. Hence, the dynamics of the ball in mid-air are

$$m\ddot{\boldsymbol{p}} = -m \begin{bmatrix} 0 \\ g \end{bmatrix} - \zeta \|\dot{\boldsymbol{p}}\| \dot{\boldsymbol{p}}\,, \tag{1}$$

where $\boldsymbol{p}$ is the position of the ball and $\zeta = 10^{-2}\text{kg}\,\text{m}^{-1}$ is the friction coefficient.

The flat surface is located at $p_2 = 0$, and the ball bounces on it without energy loss. In other words, the ball bounces when the condition

$$p_2(t) = 0\,, \quad \dot{p}_2(t) < 0 \tag{2}$$

is met, and in such case, its velocity $\dot{\boldsymbol{p}}(t)$ is updated by

$$\dot{\boldsymbol{p}}(t) \leftarrow \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \dot{\boldsymbol{p}}(t). \tag{3}$$

(a) Explain why the condition (2) gives the instant t when the ball bounces. In particular, explain why the second condition ($\dot{p}_2(t) < 0$) is necessary.
Moreover, explain why the velocity has to be updated as given by (3).

Solution: The ball bounces if and only if the ball is on the flat surface and on its way down.
In the moment the ball bounces, the vertical component of the velocity, $\dot{p}_2(t)$, changes sign. This is what (3) means.

We will start by simulating the bouncing ball with a somewhat naive approach. Here, the ODE is integrated as usual with a RK method, and at each time step the alternative condition

$$p_{k,2} \leq 0, \quad \dot{p}_{k,2} < 0 \tag{4}$$

is checked, where $\boldsymbol{p}_k$ is the approximation of $\boldsymbol{p}(t_k)$ given by the integration method.

If the condition (4) is met, then the velocity is updated as given by (3), and a new simulation is started with the last updated state values as the initial conditions.

(b) Write a forward Euler integrator that performs (3) when the alternative condition (4) is met.
Simulate the bouncing ball system, and add a plot of the ball's height $p_2(t)$ to your answer.
Hint: The template `BouncingBallRoutineTemplate.m` *can help you get started.*

Solution: The necessary modifications in `BouncingBallRoutineTemplate.m` are:

```
for nt = 2:Nt
    % simulate for next time step
    xNext = ERK(ForwardEuler,BallDynamics,T(nt-1:nt),xt);
    xt = xNext(:,end);
    % check event and update if so
    if xt(2)<=0 && xt(4)<0
        xt(4) = -xt(4);
    end
    x(:,nt) = xt;
end
```

The plot for a time step of $\Delta t = 0.05$ is shown in Figure 1.

We will now simulate the bouncing ball system using proper event detection as given by (2).

(c) Read the Matlab documentation about *ODE Event Location* (link here).

In particular, study the example `ballode.m`.

Simulate the bouncing ball system using Matlab's tools for event detection.

Add the implemented code to your answer.

Hint: This task can be solved by modifying `ballode.m` *slightly. Note that our state-space model* (1) *is different from the one used in the example.*

Solution: Necessary modifications are to change the vector field of the ODE, and to correct the indexing accordingly.

(d) Simulate the bouncing ball model for both approaches (parts b. and c.) and for different time steps Δt.

Explain what you see.

Add a comparative plot of the simulated ball's height $p_2(t)$ to your answer.

Solution:

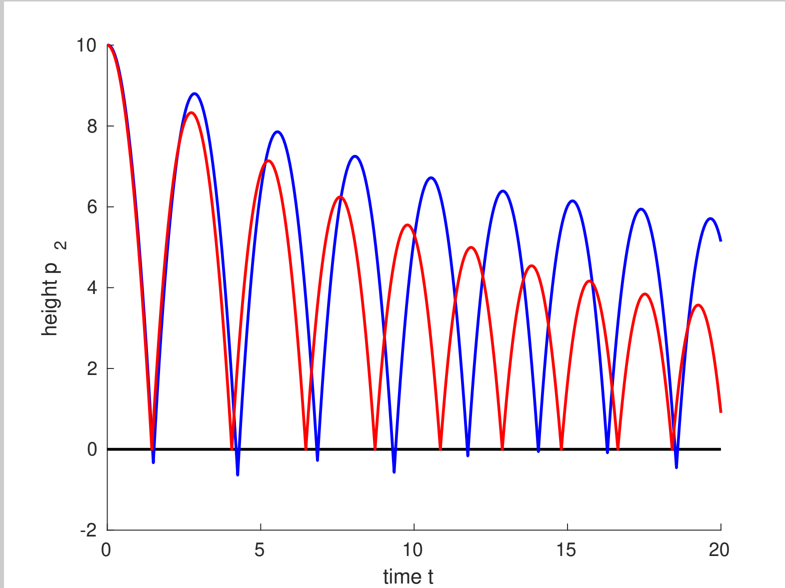

Figure 1: Simulated height using the approach from part b. with a time step of $\Delta t = 0.05$ (blue) and using `ode45` together with event detection (red).

For relatively large time steps, the naive approach gives trajectories that end up below the flat surface. This causes the ball to experience less friction than when using proper event detection (see Figure 1). As the simulation goes on, the results from both approaches diverge more and more from each other.

Problem 2 (Stick-Slip)

Consider two point masses $m_1 = 10\,\text{kg}$ and $m_2 = 0.1\,\text{kg}$ that move along a rail. m_1 moves frictionless on the rail, subject to a constant force $F_u = 1.1\,\text{N}$, while m_2 is attached to m_1 via a spring of rigidity $k = 0.5\,\text{N}\,\text{m}^{-1}$ with a rest length of $l_0 = 2\,\text{m}$. Moreover, m_2 is subject to a dry friction force that has a static force $F_s = 1\,\text{N}$ and a dynamic force $F_c = 0.8\,\text{N}$.

Hint: The different friction models that are mentioned in this problem, are explained in section 5.2. in the book.

(a) Use the position of the point masses (x_1 and x_2) and their respective velocities (v_1 and v_2) as states, and assume that the friction force is given by the Coulomb force with stiction model.

Write down the state-space model for this system.

Solution: The state-space model is

$$\begin{aligned}
\dot{x}_1 &= v_1 \\
\dot{x}_2 &= v_2 \\
\dot{v}_1 &= \frac{F_u - F_{\text{spring}}}{m_1} \\
\dot{v}_2 &= \frac{F_{\text{spring}} - F_{\text{friction}}}{m_2},
\end{aligned}$$

where

$$F_{\text{spring}} = k(x_1 - x_2 - x_0)$$
$$F_{\text{friction}} = \begin{cases} F_{\text{spring}} & , |F_{\text{spring}}| \le F_s \quad \text{and} \quad v_2 = 0 \\ F_c \text{sign}(v_2) & , \text{otherwise.} \end{cases}$$

(b) Simulate the model from part a. as it is, i.e., without implementing event detection, by using the in-built Matlab solvers `ode45` and `ode15s`.
Use the initial values $x_1(0) = 0\,\text{m}$, $x_2(0) = -2\,\text{m}$, $v_1(0) = 1\,\text{m}\,\text{s}^{-1}$ and $x_2(0) = 0\,\text{m}\,\text{s}^{-1}$.
Simulate for 15 s, and add a comparative plot of the simulated state values to your answer.
Comment on the results.

Solution: The results are shown in Figure 2. The simulation for `ode15s` fails because event detection has not been implemented.
Note that for different initial conditions or parameter values, both solvers may give correct equivalent results. In other words, the lack of event detection leaves the success or failure of the simulation to chance.

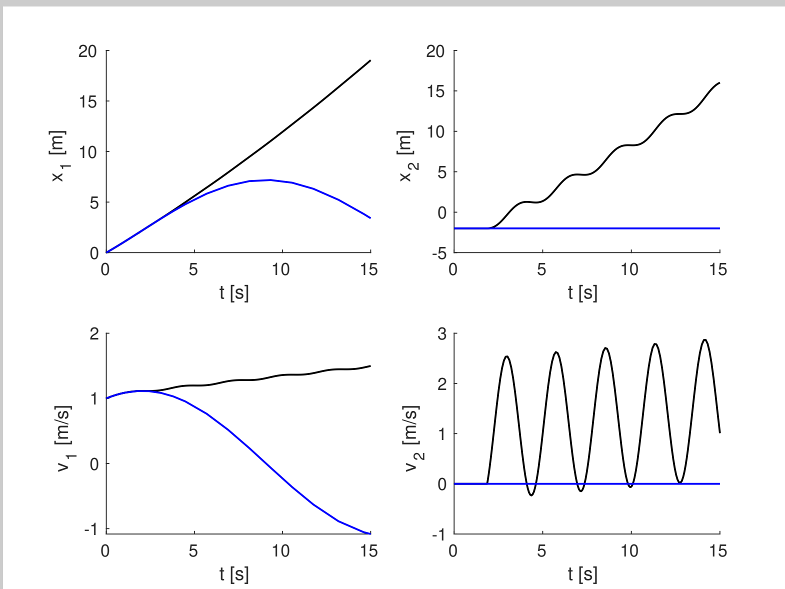

Figure 2: Results for `ode45` (black) and for `ode15s` (blue).

(c) Study and run the delivered Matlab routine `Main.m`, which simulates the model from part a. using event detection.
What events have been implemented?
Explain why there are 2 events and what is the difference between them.

Solution: The dynamics are given by two partial models. One where m_2 "sticks", i.e., it does not move and the static friction force dominates the spring force. The other partial model is where m_2 "slips", i.e., where the spring force has overcome the static friction force. In this case, the friction force experienced by m_2 is the dynamic friction force.
The 2 implemented events correspond then to the possible transitions from one model to the other one. The event `EventStick.m` checks if the absolute value of the spring force has become

larger than the static friction force. In such case, the system goes from ”sticking” to ”sliping”. Conversely, the event `EventSlip.m` checks if the velocity of m_2 has become zero. In such case, the system goes from ”sliping” to ”sticking”.

Note that the system alternates between the ”slip” and ”stick” partial models, starting in the ”stick” model.

We will now use the Armstrong-Hélouvry friction model (eq. (5.10) in the book). However, since integration codes with error control can have serious difficulties with discontinuous dynamics, we will approximate the sign function in the Armstrong-Hélouvry model by

$$\mathrm{sgn}(\dot{x}_2) \approx \tanh\left(\gamma \dot{x}_2\right). \tag{5}$$

The approximation (5) is asymptotically exact for $\gamma \to \infty$.

(d) Plot the Armstrong-Hélouvry friction force as a function of velocity, and its approximation using (5) with $\gamma = 10^3$. Use a viscous velocity of $v_s = 0.1\,\mathrm{m\,s^{-1}}$.

Moreover, plot the difference of these quantities (signed error).

Comment on the results, and add the plots to your answer.

Solution:

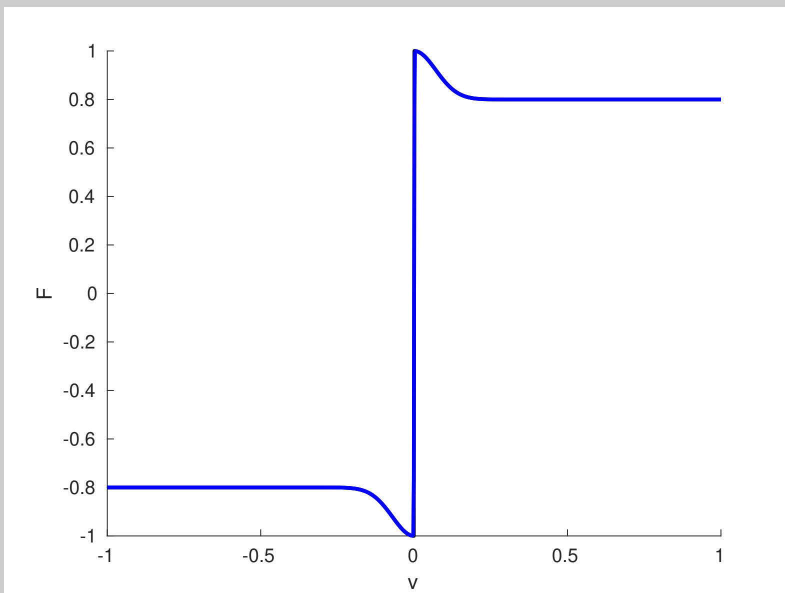

Figure 3: $F(v)$ for Armstrong-Hélouvry model (black) and approximation (blue).

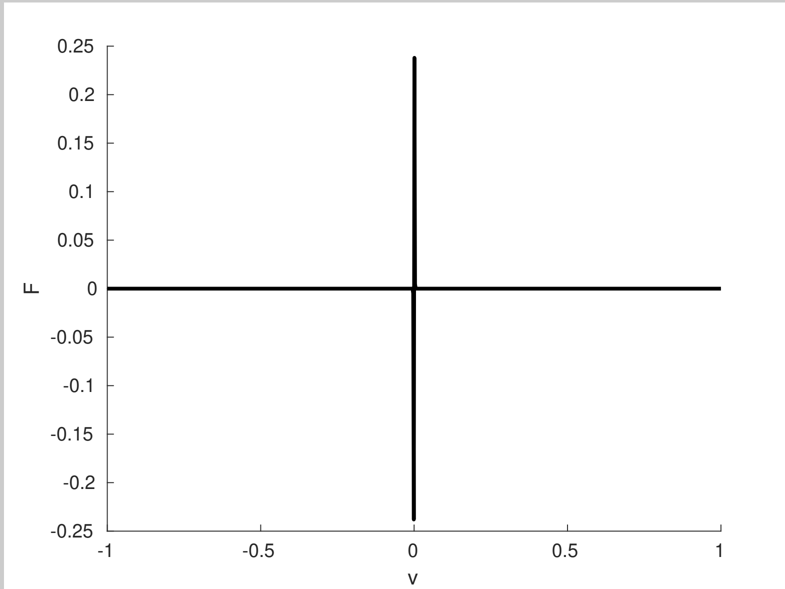

Figure 4: Approximation error of F(v).

The sign approximation (5) gives a good approximation of the friction force in the Armstrong-Hélouvry model as shown in Figure 3. However, the approximation is not so good for low velocities as shown in Figure 4.

(e) Assume now that the friction force is given by the approximated Armstrong-Hélouvry model. Write down the state-space model for this system.
What events need to be detected when simulating this model? Why?

Solution: The state-space model is

$$\begin{aligned}
\dot{x}_1 &= v_1 \\
\dot{x}_2 &= v_2 \\
\dot{v}_1 &= \frac{F_u - F_{\text{spring}}}{m_1} \\
\dot{v}_2 &= \frac{F_{\text{spring}} - F_{\text{friction}}}{m_2},
\end{aligned}$$

where

$$\begin{aligned}
F_{\text{spring}} &= k(x_1 - x_2 - x_0) \\
F_{\text{friction}} &= \left(F_c + (F_s - F_c)e^{-\left(\frac{v_2}{v_s}\right)^2}\right)\tanh(\gamma v_2).
\end{aligned}$$

There is no need to implement event detection since the dynamics can be described by one continous model.

(f) Expand the delivered routine `Main.m` in order to compare the results from the first friction model (parts a. and c.) and the approximated Armstrong-Hélouvry friction model (parts d-e.).
Use a viscous velocity of $v_s = 0.1\,\mathrm{m\,s^{-1}}$ and $\gamma = 10^3$.
Add a plot that compares the simulated state values to your answer.

Solution: The new friction model gives a continous ODE, and its numerical integration is standard.

Figure 5 compares the results from both models. The numerical values are quite similar at the beginning of the simulation, but begin to diverge from each other as the simulation goes on.

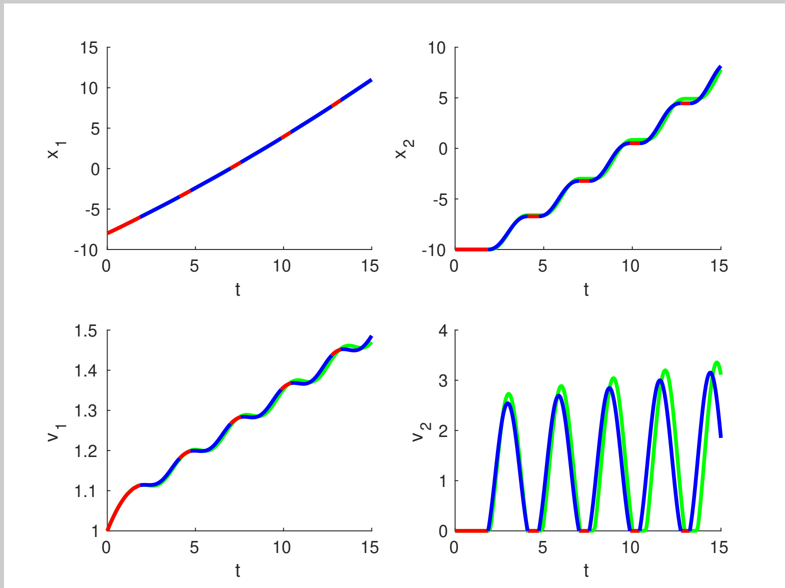

Figure 5: Results from the Coulomb force with stiction model (red and blue) and from the approximated Armstrong-Hélouvry model (green)