HYBRID SYSTEMS REPORT

MS-EMS 2015-2016

Date: 17/03/2016

Report by:

Prerana SHAMSUNDAR PUNJABI
Sandeep Gopala Krishnan

1 Table of Contents

1 Introduction

Hybrid systems are systems whose evolution depends on the continuous time and the discrete events. In this assignment we analyze the bouncing ball problem, as a hybrid system. First, we derive the Hybrid model in view of the stability analysis. Then we focus on the use of MATLAB/Simulink for study of the hybrid systems.

In this assignment, we consider hybrid automata in Figure 1 Hybrid Automata of a bouncing ball representing the behavior of a bouncing ball (with unit mass).

$$x_1 = 0 \wedge x_2 \leq 0$$

$$x_2 := -Cx_2$$

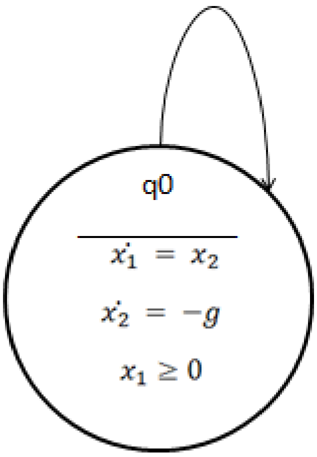

Figure 1 Hybrid Automata of a bouncing ball

Vertical position and speed are denoted by x_1 and x_2 respectively. The variable g = 9.8 represents the gravity constant and c = 0.8 is the restitution coefficient.

2 Stability Analysis

2.1 Answer – 1

x is defined as $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, we now define functions f, g and subsets of the state space F, J such that

$$\dot{x} = f(x) \ if \ x \, \epsilon F$$

$$x^+ = g(x) = if \ x \ \in J$$

The function f and g and subsets of the state space F, J are as defined below:

$$\dot{x} = f(x) = \begin{bmatrix} x_2 \\ -g \end{bmatrix} \text{if x} \,\epsilon\, \text{F} = \{\text{x} \,\epsilon \text{R}^2;\ \text{x}_1 > 0\} \cup \{\text{x} \,\epsilon \text{R}^2; \text{x}_1 = 0, \text{x}_2 > 0\}$$

$$x^+ = g(x) = \begin{bmatrix} 0 \\ -cx_2 \end{bmatrix} \text{if x} \,\epsilon\, \text{J} = \{\text{x} \,\epsilon \text{R}^2;\ \text{x}_1 = 0 \text{ and } \text{x}_2 < 0\}$$

2.2 Answer – 2

Now, the following candidate Lyapunov function (total energy of the system) with $K > \sqrt{2}\frac{1+C}{1-C}$ is given.

$$V(x) = x_2 + k\sqrt{\frac{1}{2}x_2^2 + gx_1}$$

To prove that the origin x = 0 is globally asymptotically stable for the system, we need to prove that there exists a function $V: R^n \to R_+$ such that

$$V(x = 0) = V(0) = 0$$

$$V(x) > 0, \forall x \neq 0$$

$$\dot{V}(x) < 0, \forall x \neq 0$$

$$V(x) \to \infty, when\ x \to \infty$$

where $V(x) is\ a\ Lyapunov\ function$

2.2.1 Proof of V(x=0) = 0

Replacing the value of x = 0 in the equation, we observe that V(0)=0.

2.2.2 Proof of V(x)>0

Using the formulae

$$\left(k\sqrt{\frac{1}{2}x_2^2 + gx_1}\right)^2 > 2\left(\frac{1+c}{1-c}\right)^2\left(\frac{1}{2}x_2^2 + gx_1\right) > 2\left(\frac{1}{2}x_2^2 + gx_1\right) > x_2^2$$

Therefore $k\sqrt{\frac{1}{2}x_2^2 + gx_1} > |x_2|$

Hence $V(x) = x_2 + k\sqrt{\frac{1}{2}x_2^2 + gx_1} > 0,\ \forall x \neq 0$

2.2.3 Proof of $\dot{V}$ (x)<0

$$\frac{\partial V}{\partial x_1} = \frac{1}{2} kg \left(\frac{1}{2} x_2^2 + g x_1\right)^{-\frac{1}{2}}$$

$$\frac{\partial V}{\partial x_2} = 1 + \frac{1}{2} k x_2 \left(\frac{1}{2} x_2^2 + g x_1\right)^{-\frac{1}{2}}$$

Therefore

$$\dot{V}(x) = \begin{bmatrix} \frac{\partial V}{\partial x_1} & \frac{\partial V}{\partial x_2} \end{bmatrix} f(x) = \frac{1}{2} kg x_2 \left(\frac{1}{2} x_2^2 + g x_1\right)^{-\frac{1}{2}} - g\left(1 + \frac{1}{2} k x_2 \left(\frac{1}{2} x_2^2 + g x_1\right)^{-\frac{1}{2}}\right) = -g$$

which implies that $\dot{V}(x) < 0$, $\forall x \neq 0$

2.2.4 Proof of $V(x) \rightarrow \infty$

Replacing the value of x to ∞, we notice that $V(x) \rightarrow \infty$ when x → ∞.

Since all the above properties are satisfied, it is proved that the origin x = 0 is globally asymptotically stable for the system.

3 Simulation

3.1 Answer – 1

The Simulink diagrams were updated to add information related to properties of the system. The snapshots of the updated models are as per the Figure 2 Ball.mdl: Model ball->ball, Figure 3 Ball.mdl: Model ball->Continuous Dynamic and Figure 4 Ball.mdl: Model ball->Discrete Dynamic . The 'Jump Conditions', 'Initial Conditions', 'Gravity' and the computation of x_2 were updated into the model.

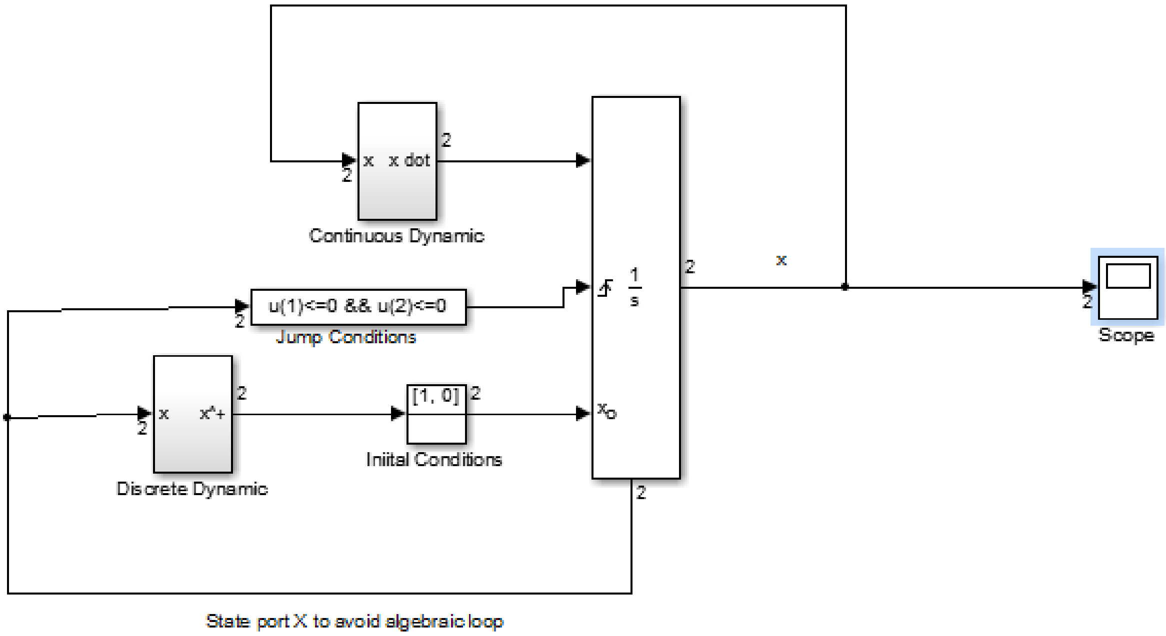

Figure 2 Ball.mdl: Model ball->ball

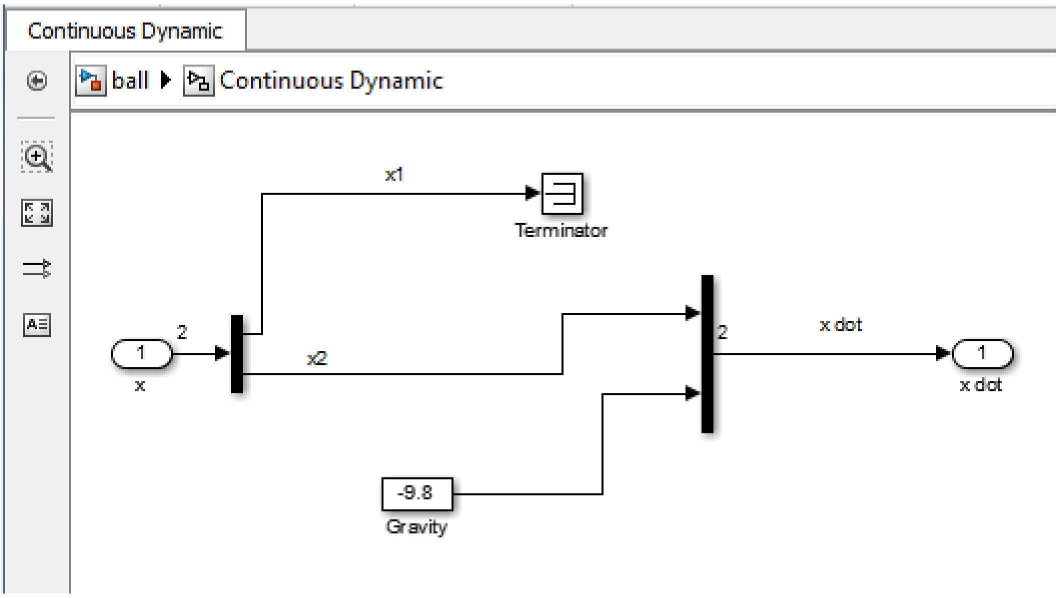

Figure 3 Ball.mdl: Model ball->Continuous Dynamic

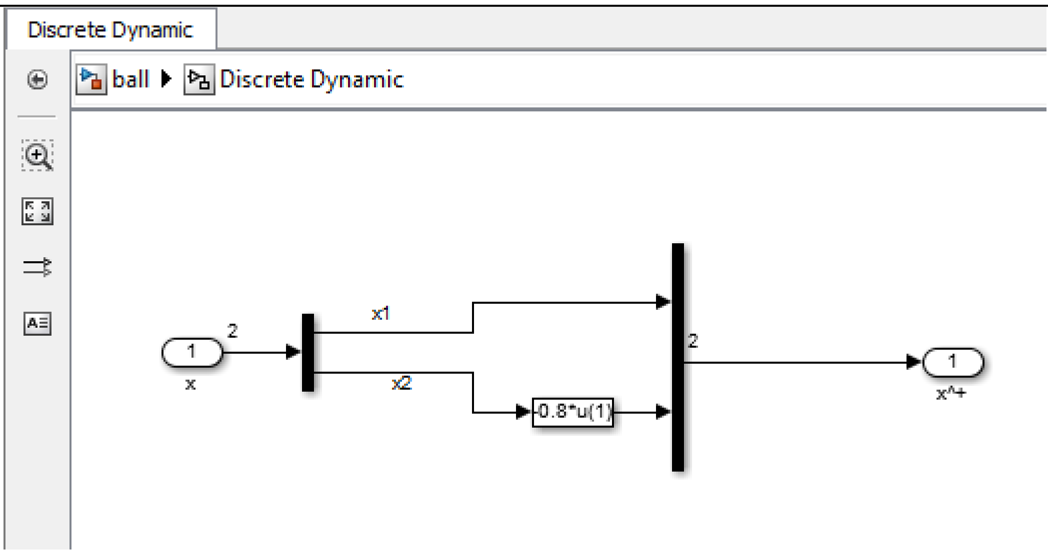

Figure 4 Ball.mdl: Model ball->Discrete Dynamic

The 'ball.mdl' model was simulated and the output was generated. The output is as shown in the Figure 5 Bouncing Ball Simulation Results. From the output, we observe that the behavior is as desired. It can be observed from the simulation result that the ball position and the speed are initially high. Eventually, the ball position and the speed damps down and reaches 0.

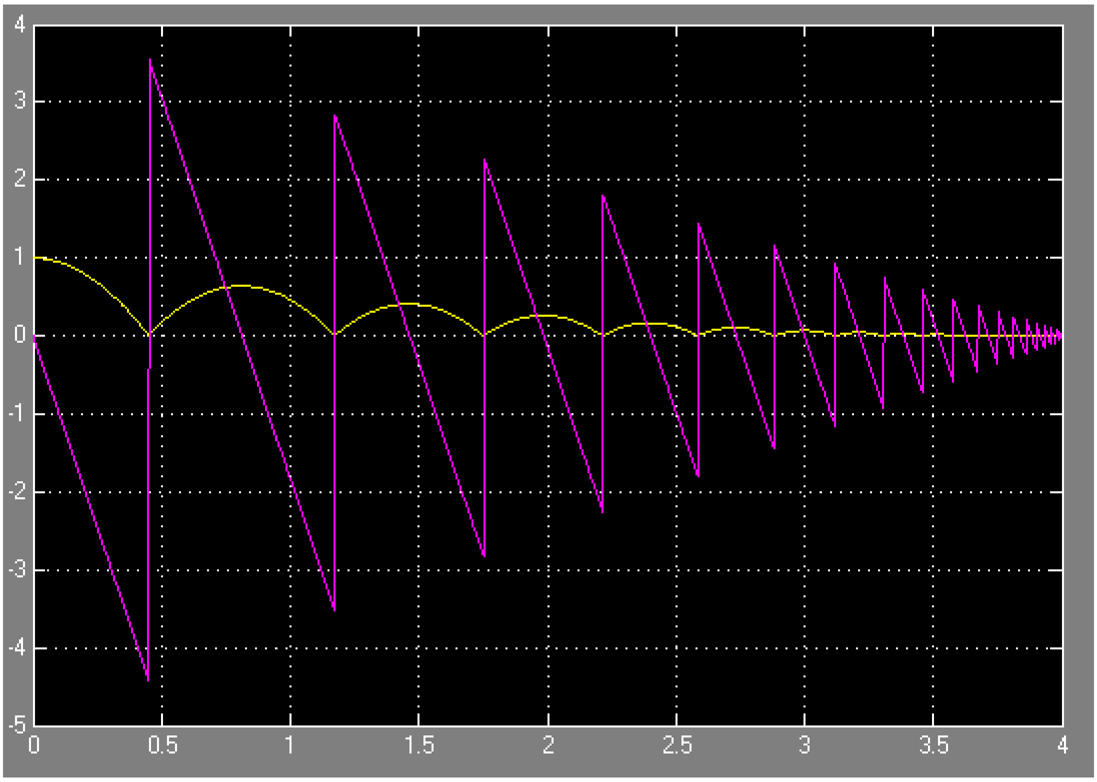

Figure 5 Bouncing Ball Simulation Results

3.2 Answer – 2

The Figure 6 Bouncing Ball Simulation with Xeno Condition depicts the snapshot of the simulation with increased simulation horizon time. When we increase the simulation window, we observe that after the time of 4 seconds, the position and speed of the ball decreases. After the 4th second, the system becomes unstable (i.e. The system enters the Xeno condition).

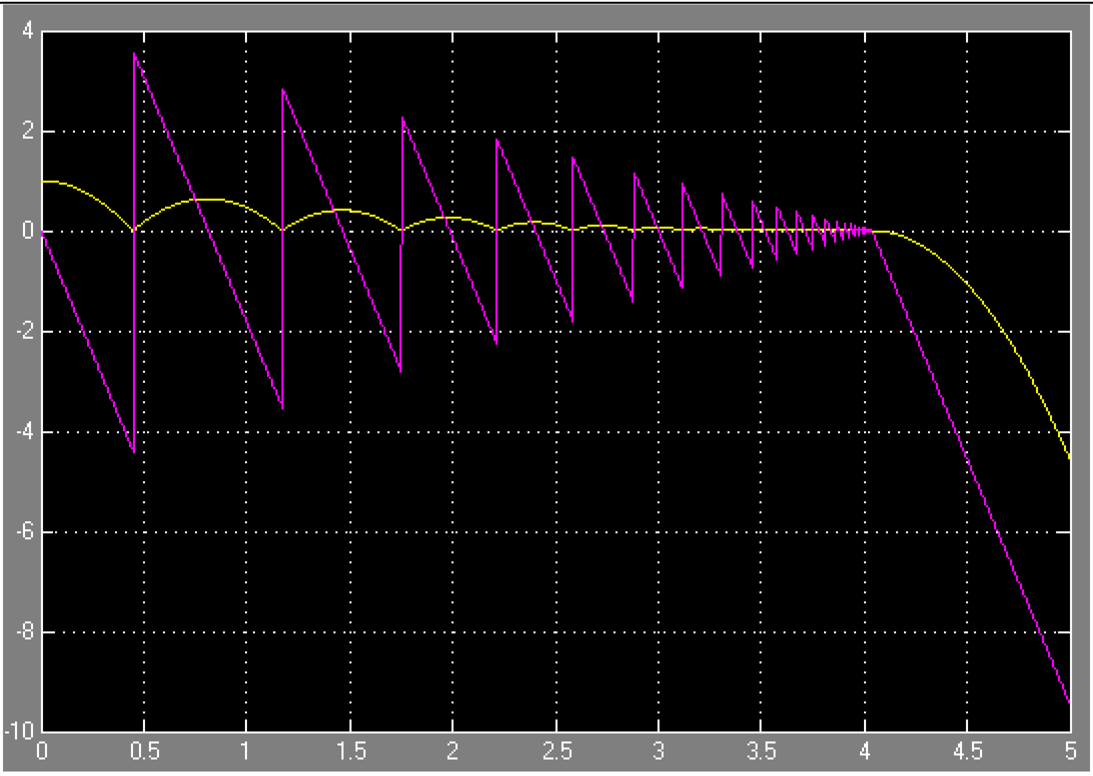

Figure 6 Bouncing Ball Simulation with Xeno Condition

To compute the sup t, we use the following formulae:

$$sup\, t = t_1 + \frac{2c\sqrt{x_2(0)^2 + 2gx_1(0)}}{g(1-c)}$$

where

$$t_1 = \sqrt{\frac{2x_1(0)}{g}}$$

The value of t_1 evaluates to 0.45 seconds. The t_1 represents the first bounce of the ball. It can be observed in the figure above that the first bounce of the ball occurs at around 0.45 seconds.

The value of Sup t evaluates to 4.06 seconds.

The Xeno time t_z is equal to sup t which is equal to 4.06 seconds. From the bouncing ball simulation in the above figure, we see that the system exhibits the Xeno behaviour starting at about 4.06 seconds.

4 Complex Simulation Tool

4.1 Answer – 1

On the execution of the Matlab file runBB.m the flows and jumps of the vertical position(x1) are as displayed in the Figure 7 RunBB.m: Bouncing Ball Vertical Position. From the figure, we can observe that the flow of vertical position initially starts at 1 (at t = 0), then drops down to 0(at t = 0.4), then jumps up again to around 0.65 (at t = 0.75). This damping behavior continues and the ball position eventually reached zero at around t = 4.1.

Similarly, the jump starts at 1 (at j = 0) and reduces with time in a damped manner. The jump eventually damps to 0(at j =10). The jump then settles at 0.

Also note that the position value for jumps and flows of x1 is always positive.

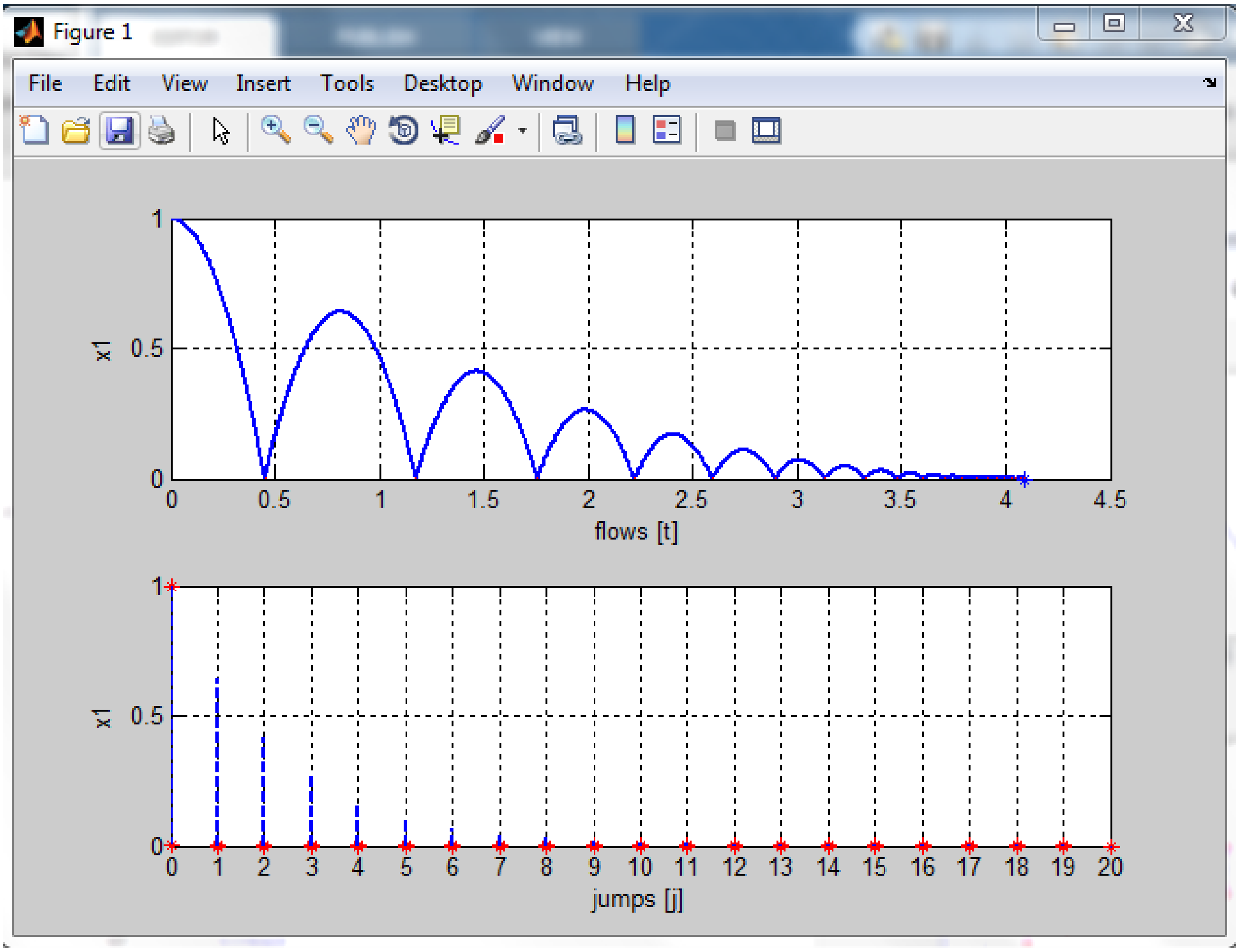

Figure 7 RunBB.m: Bouncing Ball Vertical Position

On the execution of the Matlab file runBB.m the flows and jumps of the speed(x2) are as per the snapshots in the Figure 8 RunBB.m: Bouncing Ball Speed. From the figure, we can observe that the flow of speed initially starts at 0 (at t = 0), then drops down to -4(at t = 0.4). There is a discontinuity, so the flow starts again from 3.5(at t =0.4) and decreases to around -3 (at t=1.2). This damping behavior is repeated and continues and the speed eventually reached zero at around t = 4.1.

Similarly, for the jumps, the value symmetrically damps from around 4(at j = 1) to 0(at around j =15).

Note that the speed varies between positive and negative values depending on the direction in which the ball is travelling.

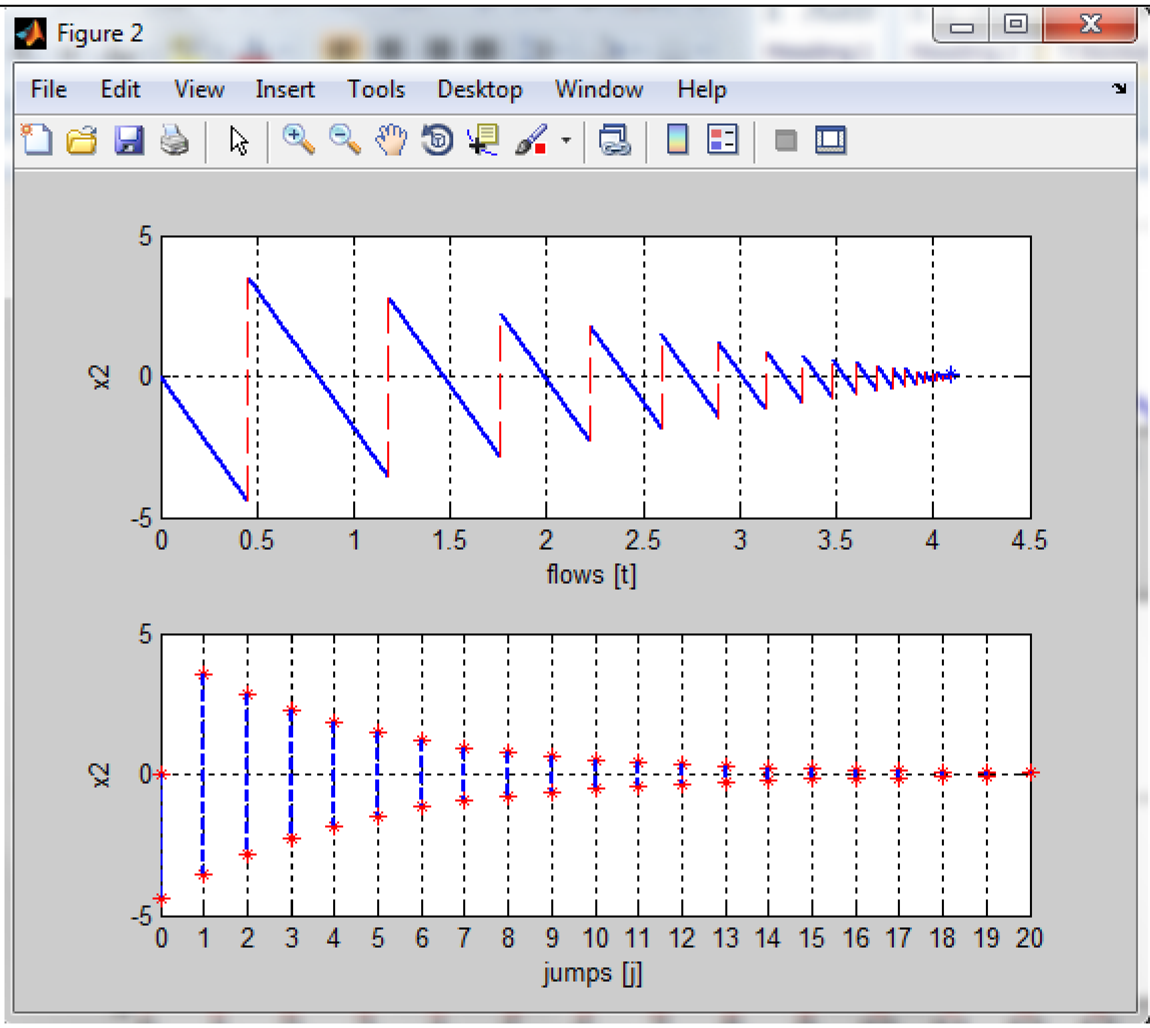

Figure 8 RunBB.m: Bouncing Ball Speed.

4.2 Answer – 2

The implementation of the stop logic is as shown in the Figure 9 Implementation of Stop Logic

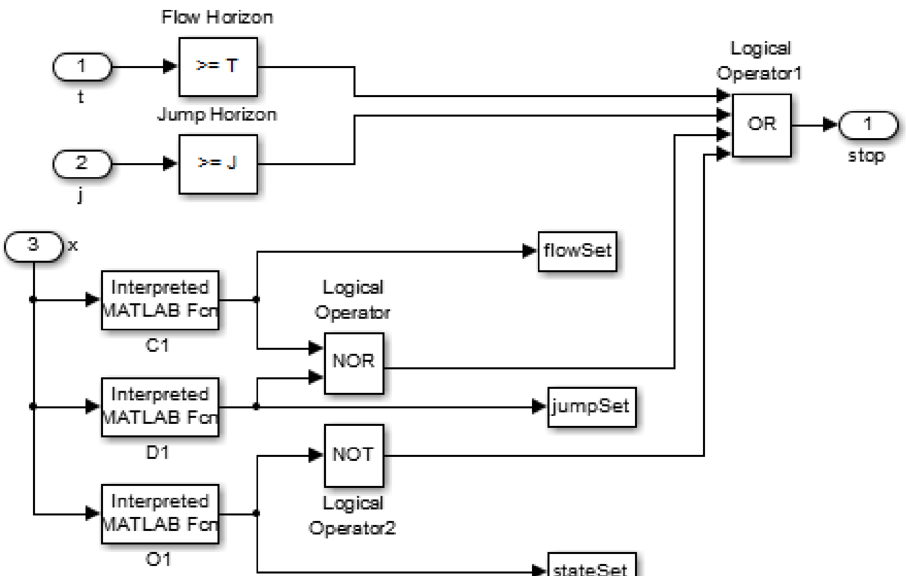

Figure 9 Implementation of Stop Logic

The stop block is used to remove the Zeno condition from the system. The stop block stops the execution of the algorithm once the ball has stopped bouncing and is on ground. It makes the system stable.

From the above figure and analyzing the matlab script files D.m, O.m and C.m, we can see that the simulation would stop if any of the following conditions are met:

1. t value is greater than or equal to 10(value of T is 10).
2. j value is greater than or equal to 20(value of J is 20).
3. The variable C1(representing the Flow Set) and the variable D1(representing the Jump set) are both false.

The variable O1 has no impact as it is hardcoded to 1 in the matlab script file O.m.

5 Back to Stability Analysis

The Lyapunov function is plotted with the vertical position variable (x1) in the Figure 10 V(x(t)) with X1.

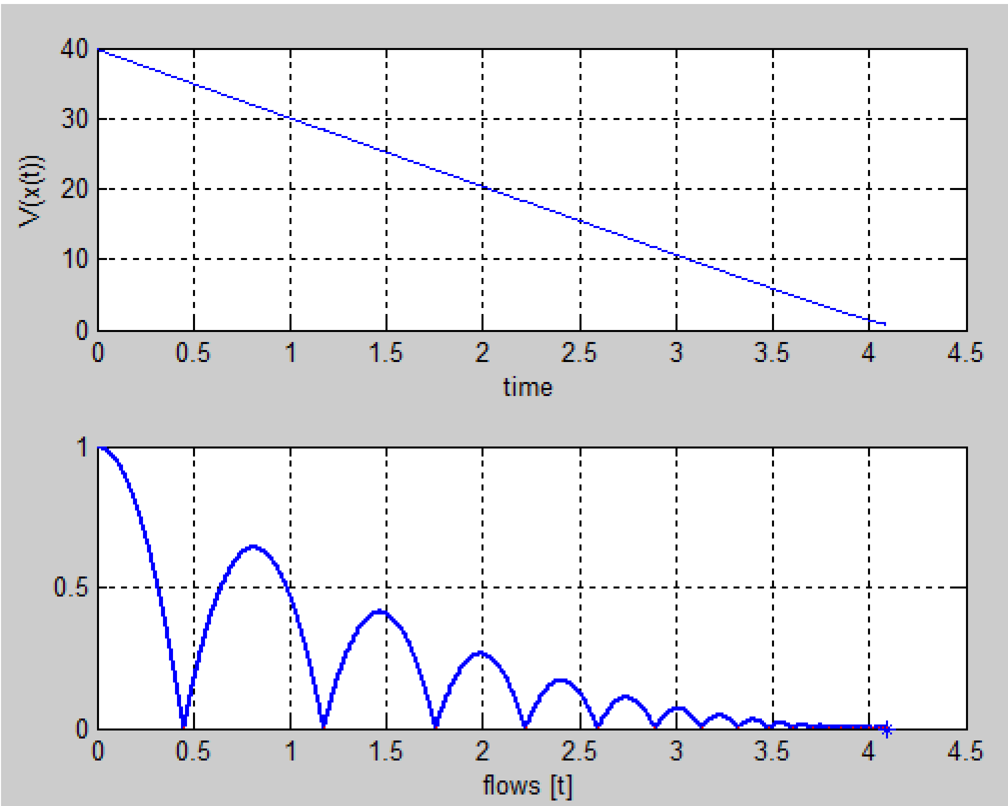

Figure 10 V(x(t)) with X1

The Lyapunov function is plotted with the speed variable (x2) in the Figure 11 V(x(t)) with X2.

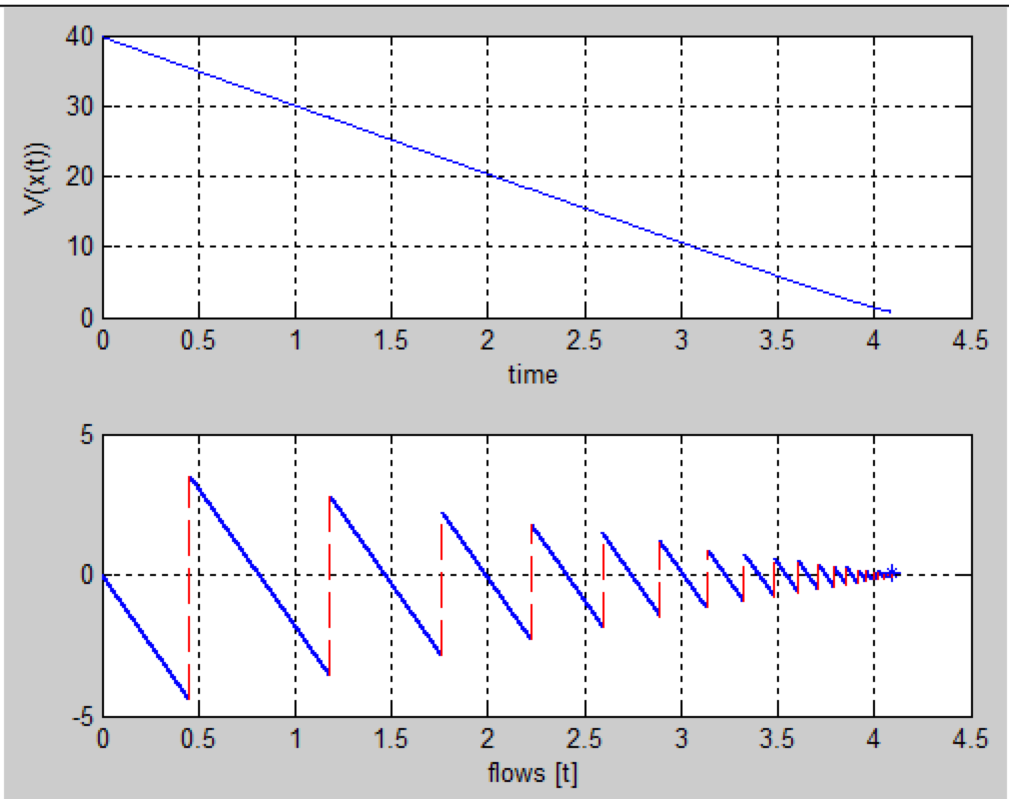

Figure 11 V(x(t)) with X2

During the flow trajectory, we observe that at the time when the jump in flow value occurs, there is a slight increase in the value of V(x(t)). This is as represented in Figure 12 Values of V(x(t)) at the X2 flow.

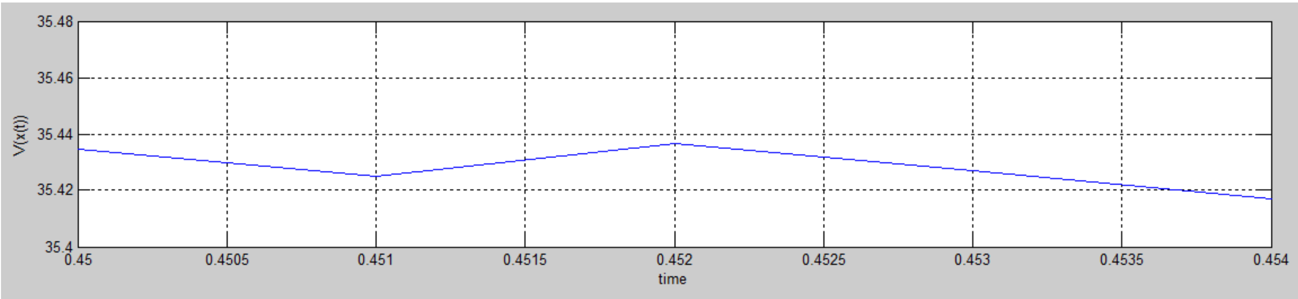

Figure 12 Values of V(x(t)) at the X2 flow.