Institute Supérieur de l'Aéronautique et de l'Espace

Master in Aerospace Engineering

Hybrid Control

Bouncing Ball

Professor:QUEINNIC Isabelle

Author: DHANSOIA Rohan, SHARMA Akash

1 Integrator

1.1 Simulation

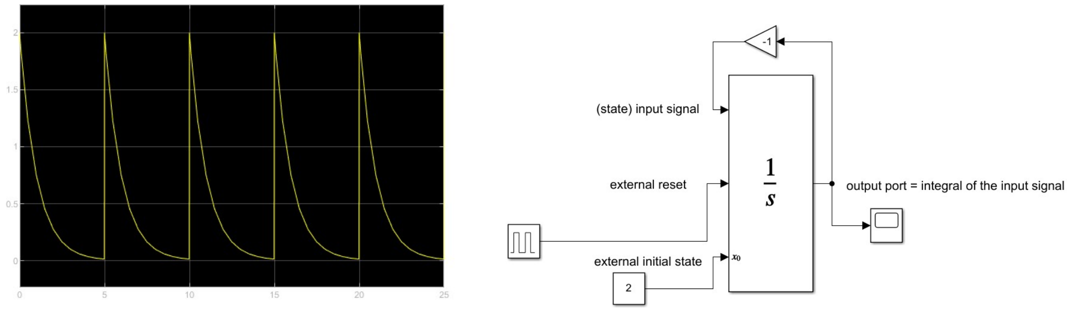

Figure 1: Behaviour plot and simulink model of the flow dynamics $\dot{x} = -x$

The dynamics of $\dot{x} = -x$ correspond to an exponential decrease in the signal which is seen in the plots. The initial state of the system is defined as a constant to 2 while the resetting of the state is carried out via a pulse generator with a period of 5 seconds which is triggered using 'level hold' mechanism.

1.2 The Integrator in a hybrid system

This model can be compared with a hybrid system where the input signal serves as the flow map f. Since the pulse generator resets the state every 5 seconds, we can think of the flow state C as the values the state takes in those 5 seconds (excluding at t=5s). The jump map g corresponds to the initial state input to the integrator since the value jumps to its initial state after every interval. Finally, the jump set D contains values of the state after every 5 seconds during which the jump takes place.

2 Counter

2.1 Does it look as a timer?

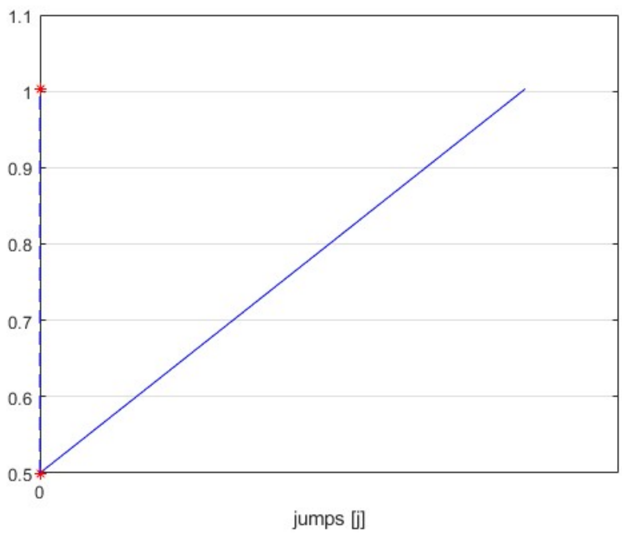

Figure 2: Behaviour of the counter model after the first run

After the first run of the simulation, it does not seem to be able to mimic a timer. The initial state of the system is set to 0.5. However, since the jump set D only contains state values that are equal to x^* whose value is 1, the simulation stops right after the state crosses this value since all values above 1 and below 0 as well neither belong to the flow set C not the jump set D. Once the jump set is corrected to take all values that are not between 0 and 1, the resulting behaviour looks like a timer.

2.2 Can the state x be arbitrarily initialized to generate a trajectory that represents a timer with resets to zero? Explain.

Since all values of x are observed in either the C or D sets after the correction, the initial state can be arbitrarily initialized to represent a timer. Any value outside of 0 and 1 resets the state to 0 after which the same behaviour is observed as expected.

2.3 Define new flow and jump sets so that the issues in a) and b) are resolved. Validate it numerically and plot for 10 seconds (and submit) three representative trajectories.

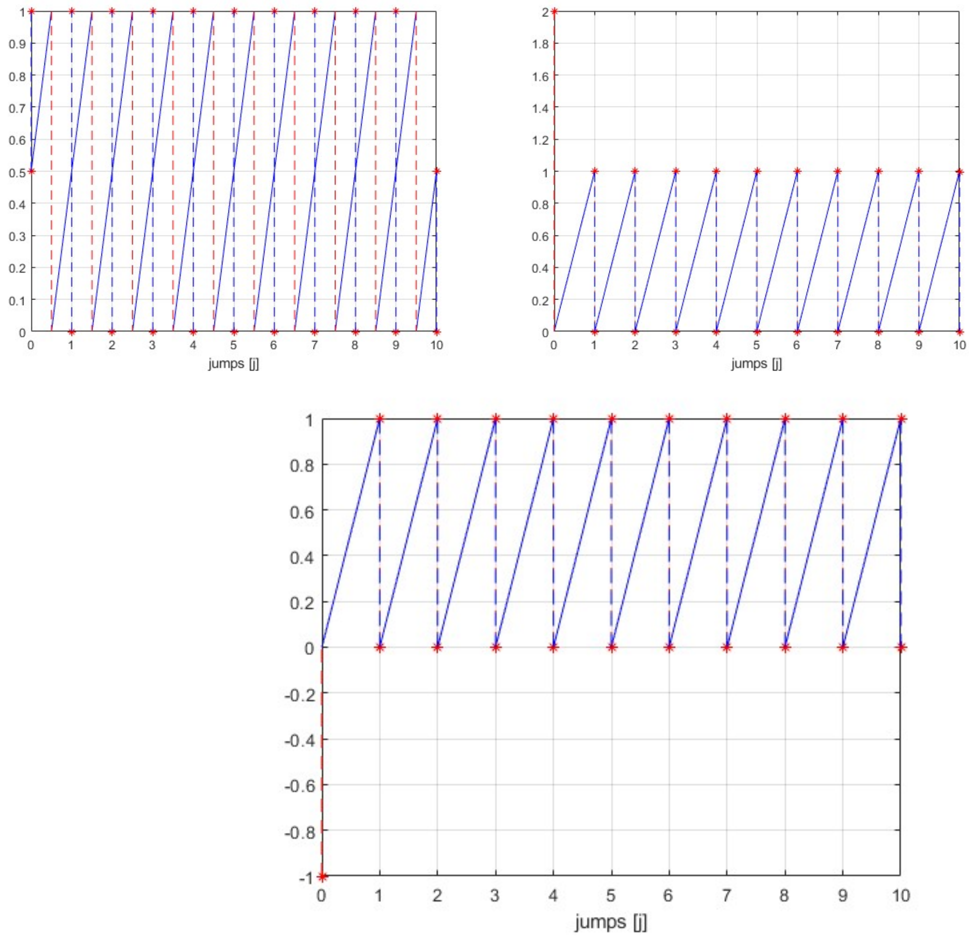

Figure 3: Behaviour plots for $x_0 = 0.5, 2, -1$ respectively

The flow set D has been redefined as follows.

```
if x >= xstar || x < 0
    insideD = 1;
else
    insideD = 0;
end
```

This new logic takes into account all possible values of the state x which allows the model to function properly as a timer.

2.4 In the simulink model, explain the role of the stop logic block hidden in the IntegratorSystem.

The stop logic block terminates the system in the event that the system fails both, the continuous jump condition and the discrete jump condition. Additionally, the stop logic terminates the system on completion of the earliest simulation horizon condition. i.e. when the system reaches the set max time or jump limit, it stops.

2.5 In the simulink model, explain the role of the Jump logic block hidden in the IntegratorSystem.

The jump block triggers the system to "jump", i.e. reset the integrator the given value based on either the discrete jump condition or both continuous and discrete jump conditions. "The output of this block, which is connected to the integrator external reset input, triggers a reset of the integrator".This priority can be varied. rule=1 would trigger the jump based only on the discrete condition. rule=2 would do so considering both, the outputs of the C block and D block.

3 The famous bouncing ball

3.1 Explain if there are any issues in obtaining appropriate trajectories.

There is an issue in the generation of the trajectory due to the discrete jump condition imposed. The system is to perform a jump when the ball hits the ground, i.e. x1 = 0. There are numerical issues with imposing this as x1==0 as the value might not be exactly 0 to machine precision, causing the condition to not be executed. Additionally, since the position then becomes slightly negative, the system is is no longer within C or D, which causes it to stop.

3.2 Can the state x be arbitrarily initialized to generate a trajectory that represents a ball bouncing?

Once the jump condition is fixed, we can initialise the position state to any arbitrary positive value and the system trajectory will represent a bouncing ball.

3.3 Define a new jump set so that the issue in a) is resolved. Validate it numerically and plot for 20 seconds (and submit) a trajectory starting with unitary height and unitary velocity (both positive). Report any problem you may experience.

The flow set D has been redefined as follows.

```
if (xtemp(1) <= 0 && xtemp(2) <= 0)  % jump condition
    insideD = 1;  % report jump
else
    insideD = 0;   % do not report jump
end
end
```

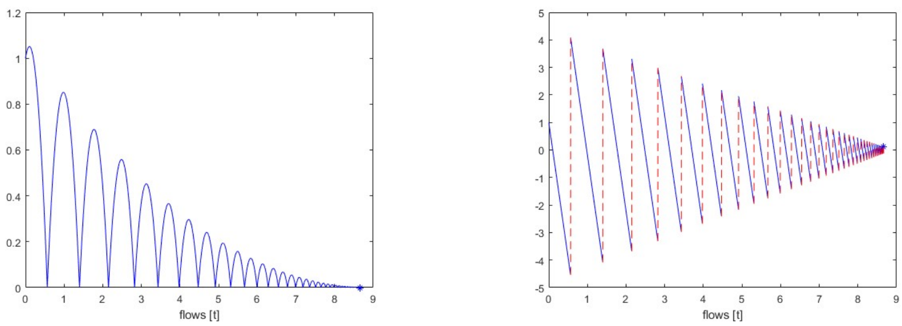

Figure 4: Behaviour plot and simulink model of the flow dynamics $\dot{x} = -x$

The behavior of the ball is as expected. The velocity continues to decrease and become negative until the ball hits the floor (x1<=0), following which the direction of the velocity vector changes due to impact and velocity becomes positive causing the ball to rise. This takes place until the number of jumps limit is reached.

3.4 For the same trajectory in c), plot the energy of the ball as a function of ordinary time. Justify its shape.

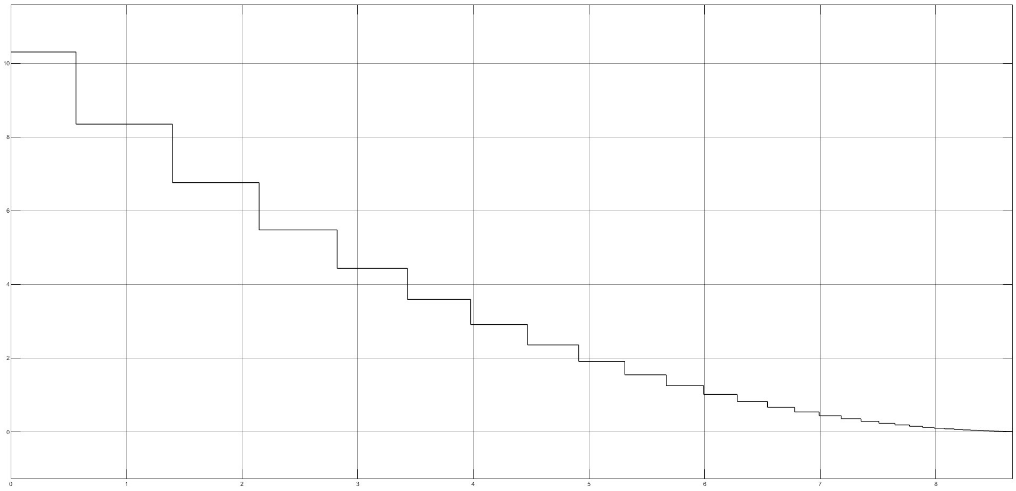

Figure 5: Behaviour of the counter model after the first run

The energy of the ball keeps decreasing with each bounce. This is due to the coefficient of restitution being lower than 1. This means there is energy loss everytime the ball colides with the surface, i.e. in this case with every jump, there is a lowering in energy.