Model based software design - Lab 03

Gianluca La Malfa
s290187

Gaia Sabbatini
s291532

1 Plant model

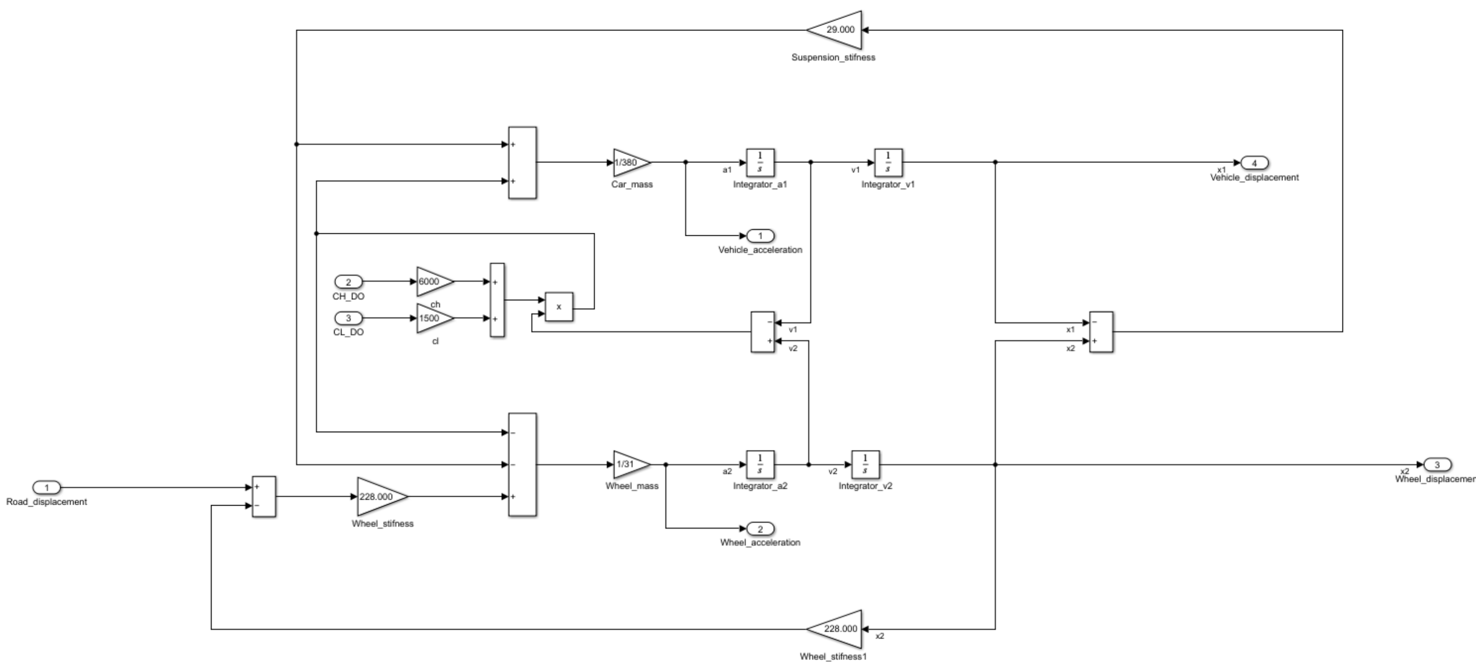

Figure 1: Quarter-car model suspension system developed through Simulink MATLAB tool

The system in Fig.1 represents a quarter-car model suspension with a semi-active controller, also known as skyhook control.
The system was built starting from dynamic eq. 1, 2 describing the suspension system.

$$\ddot{x}_1 = (k_1 x_2 - k_1 x_1 - c\dot{x}_1 + c\dot{x}_2)\frac{1}{m_1} \tag{1}$$

$$\ddot{x}_2 = [k_2 r - c\dot{x}_1 - c\dot{x}_2 + k_1 x_1 - (k_1 + k_2)x_2]\frac{1}{m_2} \tag{2}$$

Where x_1 and x_2 are respectively the car and wheel displacement with respect to their resting position. This motion is due to the fact that the vehicle is running on a non-flat surface with a relative displacement on the y-axis of r meters. The first derivatives of x_1 and x_2 are used to retrieve car and wheel speed ($\dot{x}_1$, $\dot{x}_2$), while their second derivatives return their accelerations ($\ddot{x}_1$, $\ddot{x}_2$).
In this scenario, to simulate the suspension effect the moving masses of the car (m_1) and the wheel (m_2) are considered to be attached at two springs with different stiffness (k_1, k_2). Finally, a damper (c) is positioned between the two masses to reduce the vehicle motion accordingly.

1.1 Interfaces

Table 1: Plant Interfaces table

Name	Unit*	Type	Data Type	Dimension	Min	Max
CH_DO	N.A.	Input	Boolean	Scalar	N.A.	N.A.
CL_DO	N.A.	Input	Boolean	Scalar	N.A.	N.A.
Road displacement	m	Input	Double	Scalar	-0.2	0.2
Vehicle acceleration	m/s^2	Output	Double	Scalar	-0.015	+0.015
Wheel acceleration	m/s^2	Output	Double	Scalar	-0.16	+0.16
Vehicle displacement	m	Output	Double	Scalar	-0.0008	+0.0008
Wheel displacement	m	Output	Double	Scalar	-0.00095	+0.00095

For what concerns physical interfaces, the plant receives two inputs and returns four outputs to the controller (Tab.1):

- **Inputs**:
  - *Road displacement*, a non-periodic waveform obtained adding band-limited white noise to a sine function and limiting the result between $\pm 0,2$ m.
  - *CH_DO and CL_DO*, two boolean variables generated by the controller to set the correct damping factor. When CH_DO is set to 1 (and CL_DO to 0) the vehicle suspensions are considered in their rigid configuration, while if CL_DO is set to 1 (and CH_DO to 0) the suspensions have to be considered in soft configuration.
- **Outputs**:
  - *Vehicle displacement* , represents the difference of the position of the vehicle mass with respect to its initial position.
  - *Wheel displacement* , represents the difference of the position of the wheel mass with respect to its initial position.
  - *Vehicle acceleration*, returns the current acceleration of the vehicle body.
  - *Wheel acceleration*, returns the current acceleration of the wheel.

2 Controller implementation

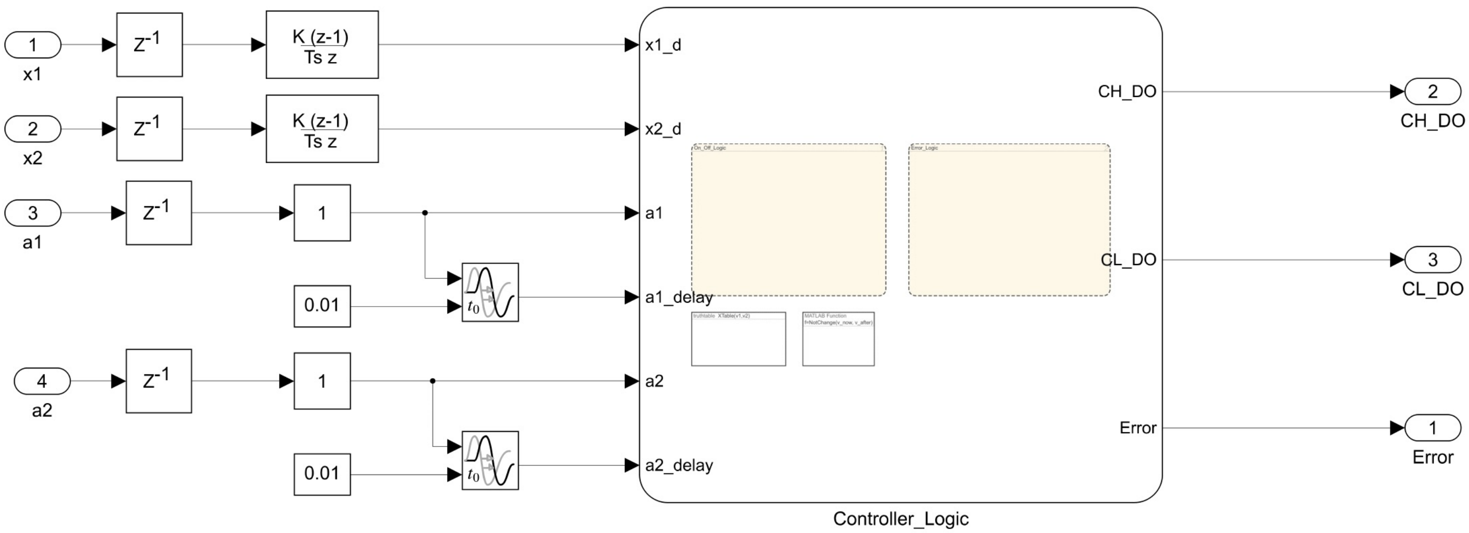

Figure 2: Controller system simulated through StateFlow MATLAB tool

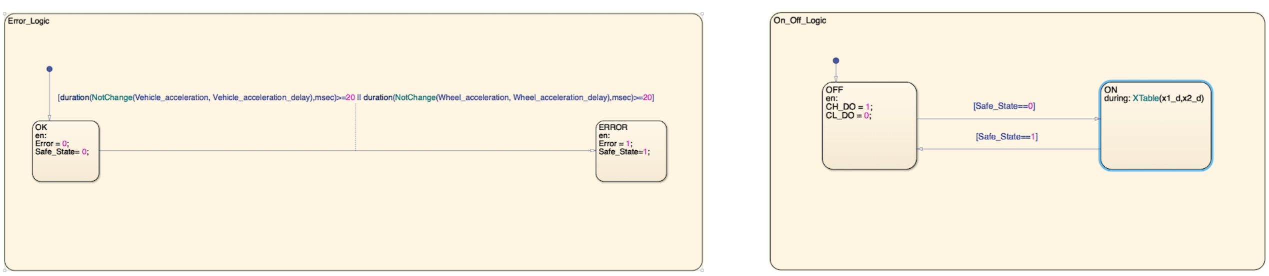

Figure 3: On/off and Error logic

Two FSMs working in parallel were implemented:

- **On/Off Logic**, used to control the activation logic of CH_DO and CL_DO (Fig.3), it was developed through the truth table function in Tab.2. The control system computes vehicle and wheel speed (x1_d, x2_d) applying discrete derivative blocks to vehicle (x1) and wheel (x2) displacement respectively. These variables are exploited (FSM Fig.3) to correctly set the appropriate damping factor following eq. 3, 4.
- **Error Logic**, returns an error if one of the input signals remains stuck and does not change in a window of $20ms$.
  For the aforementioned FSM (Fig.3) an appropriate MATLAB function (Fig.7) checking the behavior of the acceleration input signals was implemented.
  If either vehicle acceleration (a1) or wheel acceleration (a2) are stuck for more than 20 ms then the system will enter a safe state where the suspension becomes passive and the damping factor is set to $c_l = 1500\frac{Ns}{m}$.
  In addition, to check whether the Error_logic FSM works properly two slider gains were exploited, as a matter of fact an error shall rise if one or both gains are fixed to 0 (meaning that acceleration value seems to be stuck).

$$(\dot{x}_1 - \dot{x}_2)\dot{x}_1 \geq 0 \Longrightarrow c = c_h = 6000\frac{Ns}{m} \tag{3}$$

$$(\dot{x}_1 - \dot{x}_2)\dot{x}_1 < 0 \Longrightarrow c = c_l = 1500\frac{Ns}{m} \tag{4}$$

Figure 7: Matlab function

```
function f=NotChange(a_now, a_after)
    if abs(a_now-a_after)< 1e-5
        f=1;
    else
        f=0;
    end
end
```

Table 2: Truth table

DESCRIPTION	CONDITION	D1	D2	D3
More comfort is given to the driver by switching the softer damping factor.	(v1 - v2) * v1<0	T	F	-
More reactivity is given to the driver by switching the harder damping factor.	(v1 - v2) * v1 >= 0	F	T	-
	ACTIONS: SPECIFY A ROW FROM THE ACTION TABLE	**A2**	**A1**	**A1**

DESCRIPTION	ACTION
Switch to softer damping factor	A1: CH_DO = false;CL_DO = true;
Switch to harder damping factor	A2: CH_DO = true;CL_DO = false;

2.1 Interfaces

For what concerns physical interfaces, the controller receives six inputs and returns three outputs to the plant (Tab.3):

- **Inputs**:
  - *Vehicle acceleration*, returns the current acceleration of the entire vehicle.
  - *Wheel acceleration*, returns the current acceleration of the wheel.
  - *Vehicle displacement* , represents the difference of the position of the vehicle with respect to its initial position.
  - *Wheel displacement* , represents the difference of the position of the wheel with respect to its initial position.
  - *Vehicle acceleration delayed* , returns the current acceleration of the entire vehicle in a successive time.
  - *Wheel acceleration delayed* , returns the current acceleration of the wheel in a successive time.
- **Outputs**:
  - *Error*, an error flag activating if the input signals are stuck for more than $20ms$
  - *CH_DO*, a boolean signal used to activate the rigid configuration.
  - *CL_DO*, a boolean signal used to activate the soft configuration.

Table 3: Controller Interfaces table

Name	**Unit***	**Type**	**Data Type**	**Dimension**	**Min**	**Max**
Vehicle acceleration	m/s^2	Input	Double	Scalar	-0.015	+0.015
Wheel acceleration	m/s^2	Input	Double	Scalar	-0.16	+0.16
Vehicle displacement	m	Input	Double	Scalar	-0.0008	+0.0008
Wheel displacement	m	Input	Double	Scalar	-0.00095	+0.00095
Vehicle acceleration delayed	m/s^2	Input	Double	Scalar	-0.015	+0.015
Wheel acceleration delayed	m/s^2	Input	Double	Scalar	-0.16	+0.16
CH_DO	-	Output	Boolean	Scalar	-	-
CL_DO	-	Output	Boolean	Scalar	-	-
Error	-	Output	Boolean	Scalar	-	-

3 Harness

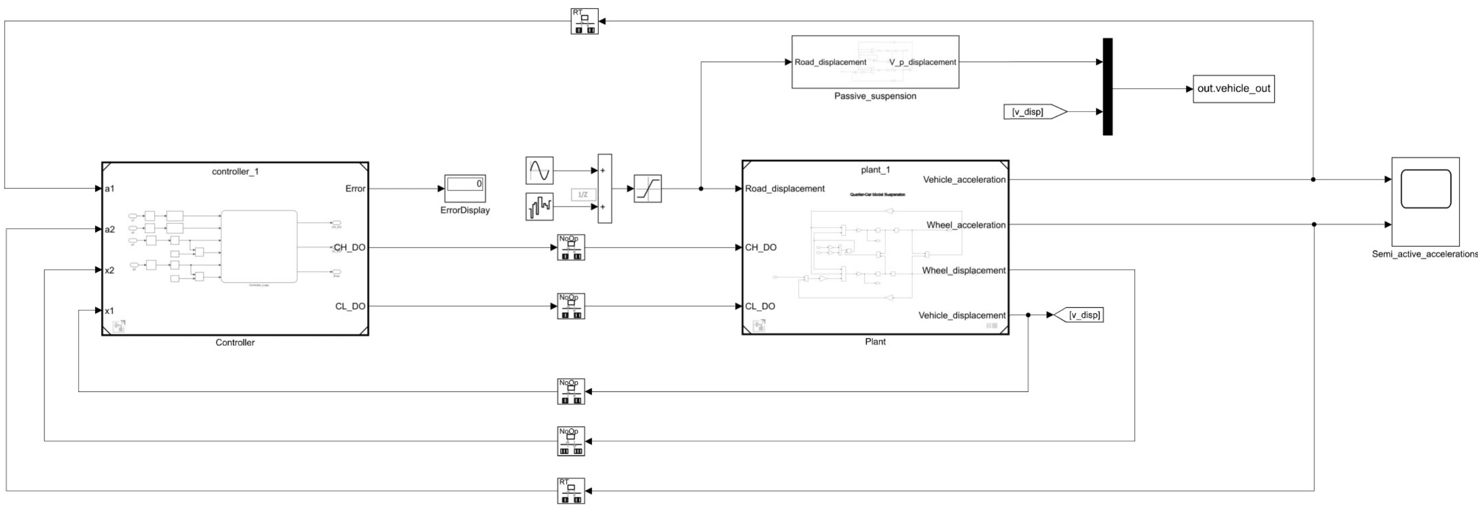

Figure 4: Harness system simulated through StateFlow MATLAB tool

3.1 Test stimuli

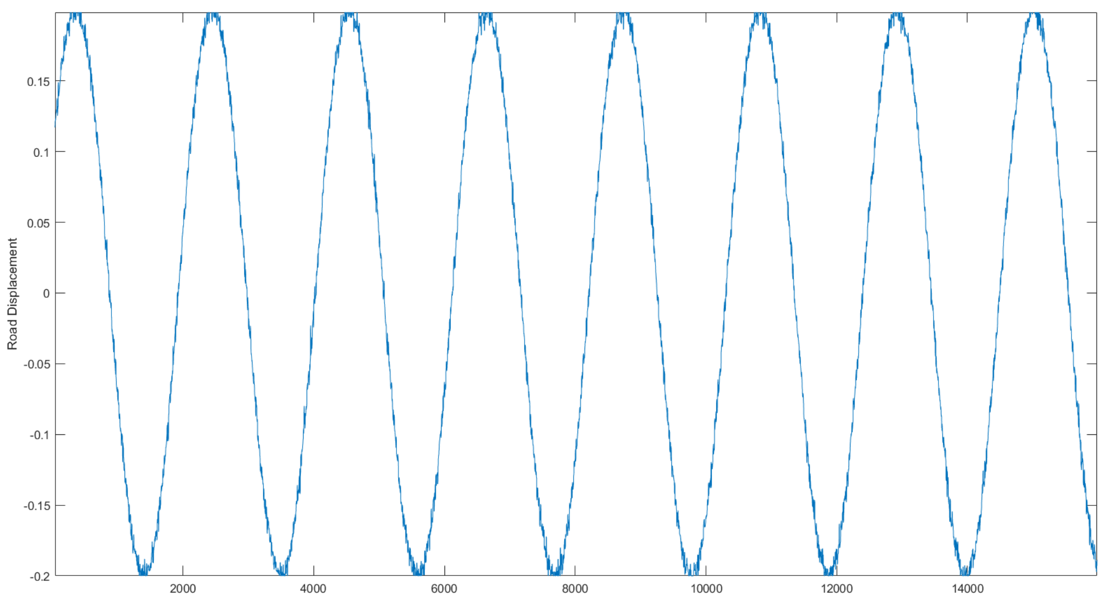

Figure 5: Test stimuli signal plotted

To test the semi-active suspension performance an appropriate signal was generated. Starting from a sine waveform with amplitude of 0.2 m and a frequency of 3 rad/s a band-limited white noise was added (noise power of 0.1μ) to avoid getting stuck in the resonance frequency of the system and generating a more realistic roadway profile. To not exceed the $\pm 0.2m$ range of road displacement a saturation block was inserted after the signal generation.

3.2 Performance Comparison

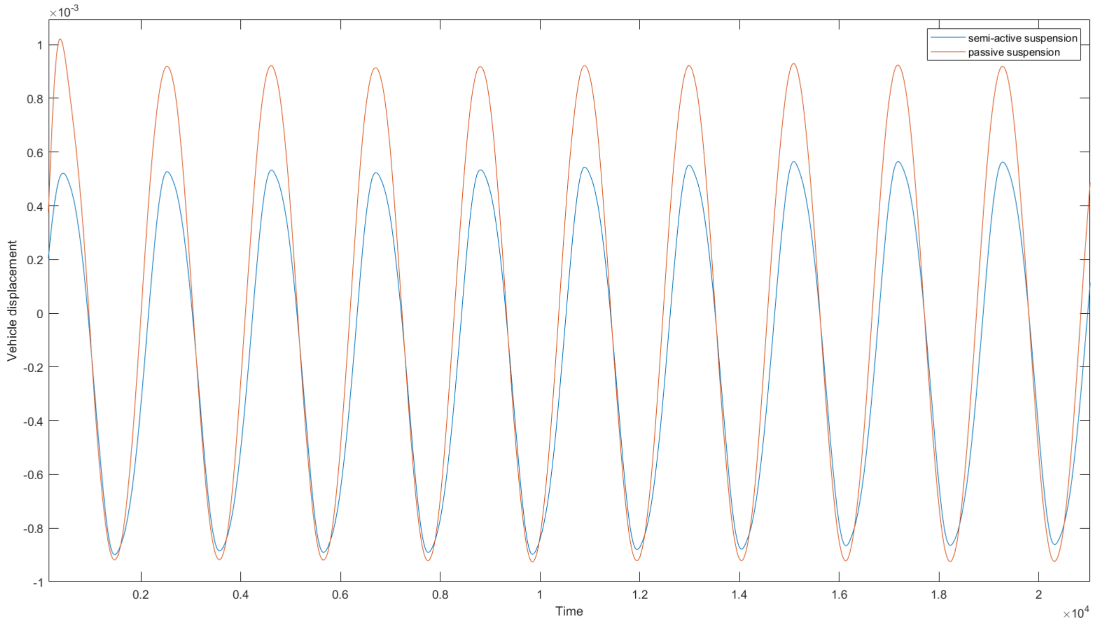

Figure 6: Graphical comparison of vehicle displacement with the semi-active (blue line) and passive (red line) suspensions

To observe the semi-active suspension performance a comparison with the passive counterpart is plotted in Fig.6. The passive suspension was build starting from the plant of Fig.1 but instead of dynamically setting the damping factor, the value of c was statically fixed at c_l.
As it happens the semi-active system makes the body vehicle a lot more stabilized, since it has a less evident displacement with respect to its starting position. On the other hand, the passive suspension returns a less stabilized system, hence the road displacement is perceived more intently by the body vehicle and, consequently, by car passengers.
So, it could be assumed that the developed system is working correctly since the graph returns the expected results in both scenarios.