# Mechatronics Modelling
# Assignment Report

**Maxime Sabbadini**
Advanced Vehicle Engineering Center
Cranfield University
Fall, 2022

December 9, 2022

---

## 1 Task 1

1. In this part, we are going to model a half car model from first principles. The system is presented figure 1 :

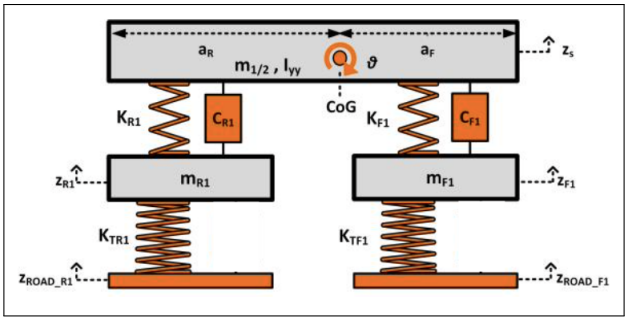


Figure 1: System we will be studying through this assignment, taken from [1]

To simplify the writing of the equations, we will note : $K_{R_1} = k_2$, $C_{R_1} = c_2$, $K_{TR_1} = k_{t_2}$, $m_{R_1} = m_2$, $K_{F_1} = k_1$, $C_{F_1} = c_1$, $K_{TF_1} = k_{t_1}$, $m_{F_1} = m_1$, $m_{1/2} = m$, $z_{\text{road}_{R1}} = u_2$, $z_{\text{road}_{F1}} = u_1$, $z_{F_1} = z_1$, $z_{R_1} = z_2$ and $z_s = z$.

We need to apply Newton's 2nd law to each of the masses :

$$m_1\ddot{z_1} = -k_{t_1}(z_1 - u_1) + k_1(z - z_1 - a_f\sin(\nu)) + c_1(\dot{z} - \dot{z_1} - a_f\dot{\nu}\cos(\nu))$$
$$m_2\ddot{z_2} = -k_{t_2}(z_2 - u_2) + k_2(z - z_2 + a_r\sin(\nu)) + c_2(\dot{z} - \dot{z_2} + a_r\dot{\nu}\cos(\nu))$$
$$m\ddot{z} = -k_1(z - z_1 - a_f\sin(\nu)) - k_2(z - z_2 + a_r\sin(\nu)) - c_1(\dot{z} - \dot{z_1} - a_f\dot{\nu}\cos(\nu)) - c_2(\dot{z} - \dot{z_2} + a_r\dot{\nu}\cos(\nu))$$

With this model, the pitch is also considered so we need an equation that will give us the pitch :

$$I_{yy}\ddot{\nu} = a_f k_1(z - z_1 - a_f\sin(\nu)) - a_r k_2(z - z_2 + a_r\sin(\nu)) + a_f c_1(\dot{z} - \dot{z_1} - a_f\dot{\nu}\cos(\nu)) - a_r c_2(\dot{z} - \dot{z_2} + a_r\dot{\nu}\cos(\nu))$$

2. As we can see from the equations of our system developed earlier, it is non linear as we have some cosine and sine. We will develop further on the non-linearities and how to linearize the problem. Our equations linearized are now :

$$m_1\ddot{z_1} = -k_{t_1}(z_1 - u_1) + k_1(z - z_1 - a_f\nu) + c_1(\dot{z} - \dot{z_1} - a_f\dot{\nu})$$
$$m_2\ddot{z_2} = -k_{t_2}(z_2 - u_2) + k_2(z - z_2 + a_r\nu) + c_2(\dot{z} - \dot{z_2} + a_r\dot{\nu})$$
$$m\ddot{z} = -k_1(z - z_1 - a_f\nu) - k_2(z - z_2 + a_r\nu) - c_1(\dot{z} - \dot{z_1} - a_f\dot{\nu}) - c_2(\dot{z} - \dot{z_2} + a_r\dot{\nu})$$

$$I_{yy}\ddot{\nu} = a_f k_1(z - z_1 - a_f\nu) - a_r k_2(z - z_2 + a_r\nu) + a_f c_1(\dot{z} - \dot{z_1} - a_f\dot{\nu}) - a_r c_2(\dot{z} - \dot{z_2} + a_r\dot{\nu})$$

We are now going to develop our system in the state space form. Let the following state space vector :

$$q = \begin{pmatrix} q_1 = z \\ q_2 = \nu \\ q_3 = z_1 \\ q_4 = z_2 \\ q_5 = \dot{z} \\ q_6 = \dot{\nu} \\ q_7 = \dot{z_1} \\ q_8 = \dot{z_2} \end{pmatrix}$$

By rewriting the equations of our system we get :

$$m\dot{q_5} = -(k_1 + k_2)q_1 + (k_1 a_f - k_2 a_r)q_2 + k_1 q_3 + k_2 q_4 - (c_1 + c_2)q_5 + (c_1 a_f - c_2 a_r)q_6 + c_1 q_7 + c_2 q_8$$

$$I_{yy}\dot{q_6} = (a_f k_1 - a_r k_2)q_1 - (a_f^2 k_1 + a_r^2 k_2)q_2 - a_f k_1 q_3 + a_r k_2 q_4 + (a_f c_1 - a_r c_2)q_5 - (a_f^2 c_1 + a_r^2 c_2)q_6 - a_f c_1 q_7 + a_r c_2 q_8$$

$$m_1\dot{q_7} = k_1 q_1 - k_1 a_f q_2 - (k_{t_1} + k_1)q_3 + 0q_4 + c_1 q_5 - c_1 a_f q_6 - c_1 q_7 + 0q_8 + k_{t_1} u_1$$

$$m_2\dot{q_8} = k_2 q_1 + k_2 a_r q_2 + 0q_3 - (k_{t_2} + k_2)q_4 + c_2 q_5 + a_r c_2 q_6 + 0q_7 - c_2 q_8 + k_{t_2} u_2$$

Now we can deduce from these equations the state space matrices such that :

$$\dot{q} = Aq + Bu \tag{1}$$

We have :

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{k_1 + k_2}{m} & \frac{k_1 a_f - k_2 a_r}{m} & \frac{k_1}{m} & \frac{k_2}{m} & -\frac{c_1 + c_2}{m} & \frac{c_1 a_f - c_2 a_r}{m} & \frac{c_1}{m} & \frac{c_2}{m} \\ \frac{a_f k_1 - a_r k_2}{Iyy} & -\frac{a_f^2 k_1 + a_r^2 k_2}{Iyy} & -\frac{a_f k_1}{Iyy} & \frac{a_r k_2}{Iyy} & \frac{a_f c_1 - a_r c_2}{Iyy} & -\frac{a_f^2 c_1 + a_r^2 c_2}{Iyy} & -\frac{a_f c_1}{I_{yy}} & \frac{a_r c_2}{Iyy} \\ \frac{k_1}{m_1} & \frac{k_1 a_f}{m_1} & -\frac{k_1 + k_{t_1}}{m_1} & 0 & \frac{c_1}{m_1} & -\frac{c_1 a_f}{m_1} & -\frac{c_1}{m_1} & 0 \\ \frac{k_2}{m_2} & \frac{k_2 a_r}{m_2} & 0 & -\frac{k_{t_2} + k_2}{m_2} & \frac{c_2}{m_2} & \frac{a_r c_2}{m_2} & 0 & -\frac{c_2}{m_2} \end{pmatrix}$$

and

$$B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{k_{t_1}}{m_1} & 0 \\ 0 & \frac{k_{t_2}}{m_2} \end{pmatrix}$$

We also need to write here the output equation which is under the form :

$$y = Cq + Du \tag{2}$$

From a Vehicle Dynamics point of view, we may be interested in the vertical relative displacement $z$ but also to the vertical relative speed $\dot{z}$. We may also be interested in the pitch angle $\nu$ and the angular pitch rate $\dot{\nu}$. So, we are going to have four outputs to our system which gives us the $C$ and $D$ matrices :

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

and

$$D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Although this can change, if we want to have an other overview of the behavior of other states of the system we can still modify that $C$ matrix to get what we want at the output. We used the following parameters for the parameters of the system (table 2) :

| Parameter | Value | Unit |
|---|---|---|
| $m_{1,2}$ | 40 | $kg$ |
| m | 400 | $kg$ |
| $c_{1,2}$ | 1500 | $Ns/m$ |
| $k_{t_{1,2}}$ | 150 000 | $N/m$ |
| $k_{1,2}$ | 21 000 | $N/m$ |
| $I_{yy}$ | 600 | $kg.m^2$ |
| $a_r$ | 1.45 | m |
| $a_f$ | 0.8 | m |

Table 1: Values of the parameters used in the modelisation

3. As we have seen before, our model has some equations depending on the sine and cosine of one of the state variables. This behavior is non linear which therefore prevents us to generate a linear state space model of our system. In order to linearize the model, we had to make assumptions. In our case, our assumption is that we will be working with very small angle $\nu$. In other word, we say that $\nu \approx 0$. With that assumption, we are able to linearize both sine and cosine as follow : $\sin\nu \approx \nu$ and $\cos\nu \approx 1$.
With this assumption, we expect our model to behave very close to the non-linear model. However, the more $\nu$ will grow, the more we will move away from the assumption $\nu \approx 0$ which means that the behavior of our linearized model will not be close anymore to the original system. What we found here is an operating point of our system around which we linearized it. If we want to study the system for greater angles, we will have to make other assumptions at other operating points in order to study it. Linearizing a system is very important because once it is done, we will be able to build controllers for this linearized system which will facilitate greatly the work of the controls engineer. Once the controller has been designed for the linearized version of the system, we can then test it on the real system and observe its behavior.

# 2 Task 2

1. In our system, we have 2 inputs and 4 outputs. So we should have 8 transfer functions linking every input with every output. In order to generate our transfer functions in Matlab, we use the following syntax (listing 1) :

```
sys = ss(A, B, C, D); % Create the state space object
tf_matrix = zpk(sys);
```

Listing 1: Syntax to create the transfer functions

This command returns us the following (figure 2) :

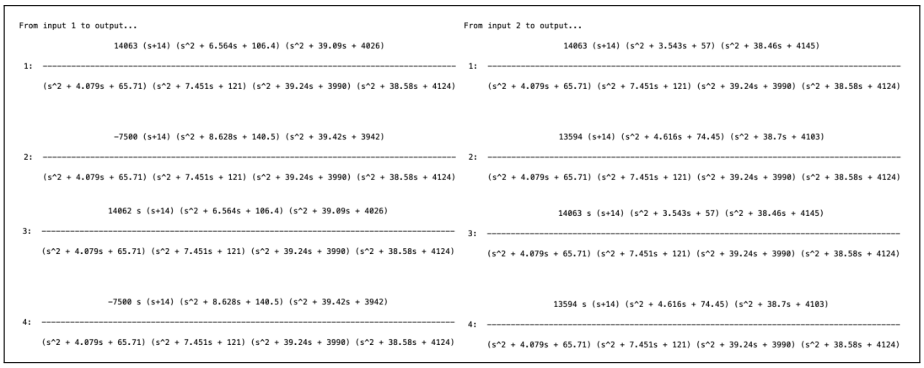


Figure 2: Output of the zpk command

We can see that we have what we expected : 8 transfer functions. Now we can move on to the next step.

2. We are now going to generate a set of inputs to test our system. The first and most obvious input we can think of is to see how the system would behave if we hit a curb. This input is shown figure 3 :

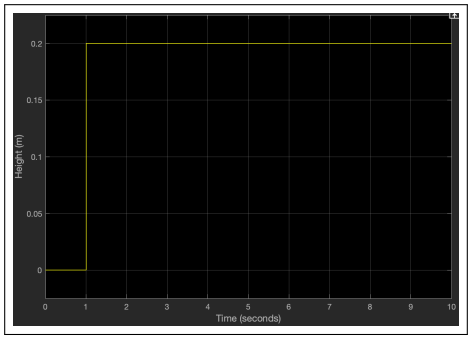


Figure 3: Curb input of the system

The other type of input we can think of is a speed bump on the road (figure 4) :

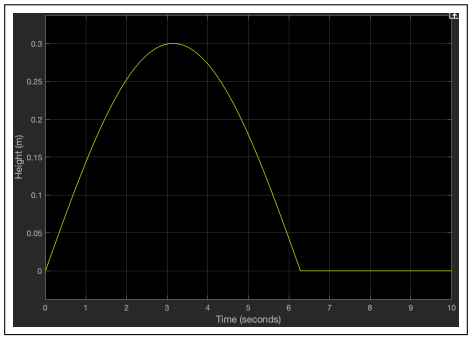


Figure 4: Bump input of the system

The last input we are going to try is when the vehicle is rolling on a bumpy road (figure 5) :

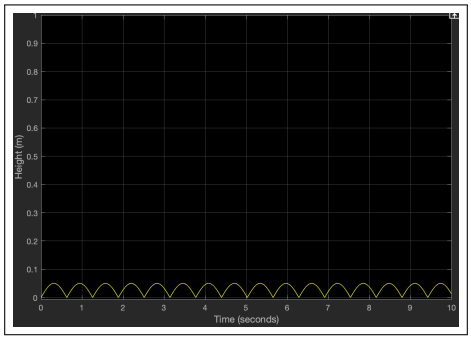


Figure 5: Bumpy road input of the system

We know that our system has two inputs, which is the height of the road under the tire. We are going to use the same input for both with a slight difference, the input at the rear is going to be slightly delayed from the input at the front so we can simulate that the vehicle is rolling. This is illustrated figure 6 :

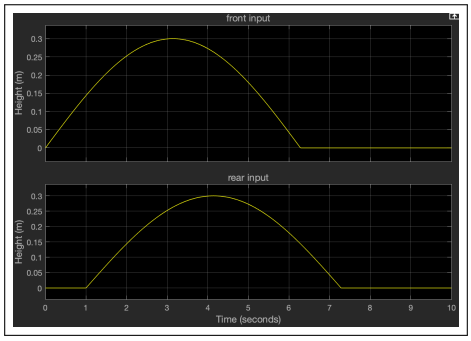


Figure 6: Illustration of the delay of the input at the rear compared to the front

3. Now we are going to analyze the response of the system to these inputs to deduce the dynamics of the system. Here, we are going to focus on the vertical position $z$ and angular position $\nu$. Let's first have a look at the response of the system when we hit a curb (figure 7) :

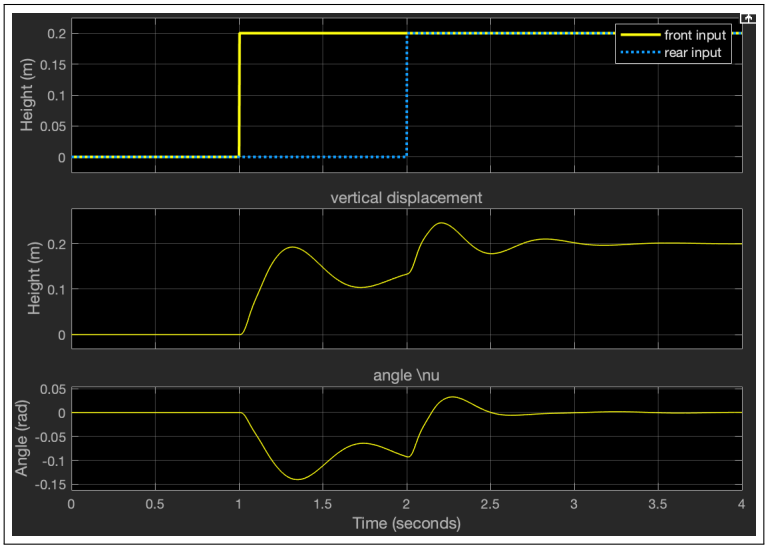


Figure 7: Response of the system when we hit a curb

With this figure, we see that when we hit the curb with the front wheel, the vertical displacement goes up while the angle goes into the negatives which is what we could expect. Then, we clearly see on the plots when the rear wheel hits the curb as well as the system oscillates again to then settle to a steady

state value which is the height of the curb for the vertical displacement and 0 for the angle. The system settles in approximately 1 second after we hit the curb which is acceptable but could be improved by tuning the stiffness of the suspension and the damping coefficients. Now we can have a look at the response of the system after rolling over a speed bump (figure 8) :

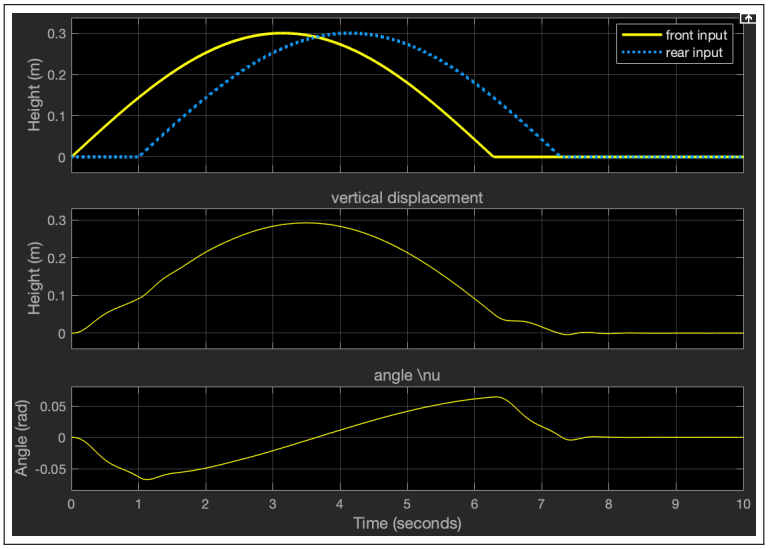


Figure 8: Response of the system when we roll over a speed bump

Here, one thing remarkable is that the angle $\nu$ has some variations which are almost linear while the vehicle rolls over a speed bump, defined by a sine function. The system is behaving as we would expect it to. Finally, we can observe the reaction of the system to a bumpy road (figure 9) :

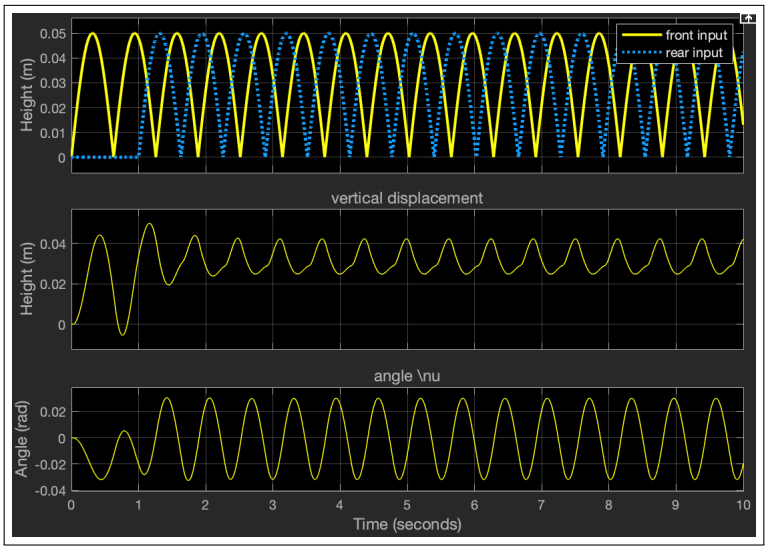


Figure 9: Response of the system when we roll on a bumpy road

Here, we can see that as soon as both wheels are on the road, the vehicle oscillates between a vertical position of 2 and 4cm above ground, while the angle oscillates also as the car moves up and down. However, the oscillations are reduced from the ground so we can say that our suspension system works well.

4. We are now going to focus on the non-linearities of the system. As we have seen during the mathematical formulation of the problem, the original system is non linear because we have to take the sine and cosine of one of the states (the angle $\nu$). In order to simplify the problem we have made an assumption which is that we will be working with very small angles so we could use the following simplifications : $\cos\nu \approx 1$ and $\sin\nu \approx \nu$. In Matlab, we have modeled the linearized system, but we have also modeled the non linear system to compare both of them and highlight the non-linearities. In order to highlight the

linearities, we will be considering a step input but only on the rear so that the vehicle has a big enough angle $\nu$ such that we can see the non-linearities of the system (figure 10) :

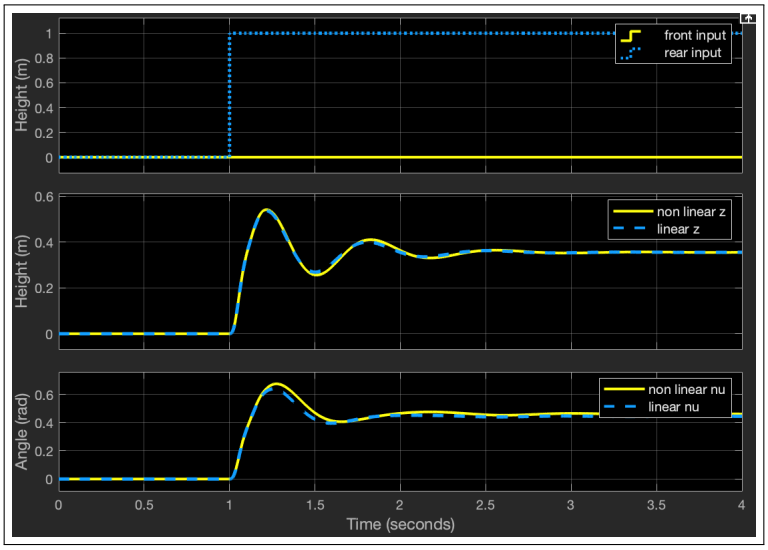


Figure 10: Highlighting the non-linearities of the system

Here we can clearly see the non-linearities of the system as the linear model is no more corresponding to the linearized version of it. Indeed, the 1m curb input on the rear does not make real sense as here we would not have to worry about the travel of the suspension, we would have to worry for the integrity of the car. But, this shows that at some point our assumptions become wrong and we would need to make other to study the system around these points. Linearizing a system is very subtle and requires to have knowledge of all the phenomenons that occur in order to better choose the operating point around which you will linearize the system.

The system we modeled on Matlab/Simulink can be found figure 11.

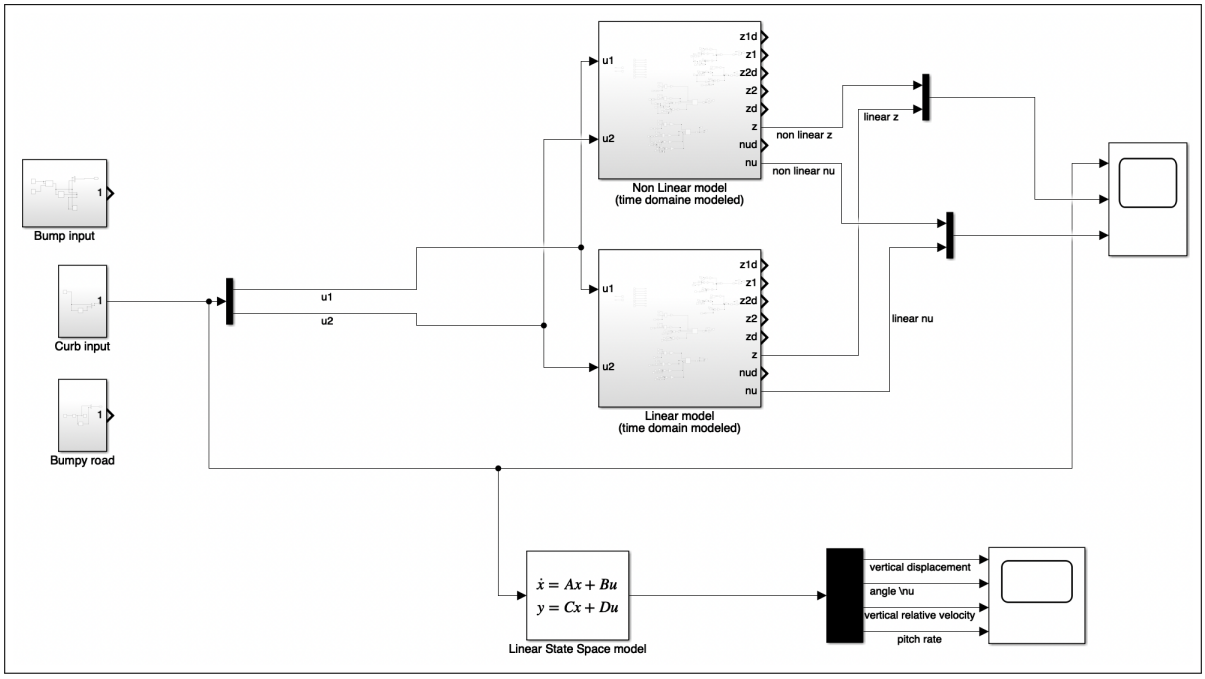


Figure 11: System modeled on Matlab/Simulink

# References

**[1]** Simulation of Vehicle Dynamic Behavior lecture slides, Dr Georgios Papaioannou, KTH Royal Institute of Technology