**Report On,**

**Simulation of Anti-Lock Braking system using Simulink.**

**Report Submitted From,**

**S R Mithun**

**2005560**

**Table of Contents:**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Title** | **Page No** |
| 1 | Introduction | 3 |
| 2 | System Modeling | 4 |
| 3 | Result and Conclusion | 7 |

**Table of Figures:**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Title** | **Page No** |
| 1 | Model of ABS | 4 |
| 2 | Signal built using signal builder | 4 |
| 3 | Solver Characteristics | 5 |
| 4 | Data declared through call back function | 5 |
| 5 | Subsystem of Wheel Speed | 6 |
| 6 | Matlab Code to calculate slip | 6 |
| 7 | block parameter for 1-D look-up table | 7 |
| 8 | values of mu and slip | 7 |
| 9 | Slip obtained based on the desired slip value. | 8 |
| 10 | vehicle speed and wheel speed. | 9 |
| 11 | Comparison of stopping distance with ABS and without ABS. | 9 |

**Introduction**

An anti-lock braking system (ABS) is a [safety](https://en.wikipedia.org/wiki/Automobile_safety) anti-[skid](https://en.wikipedia.org/wiki/Skid_(automobile)) braking system used on [aircraft](https://en.wikipedia.org/wiki/Aircraft) and on land [vehicles](https://en.wikipedia.org/wiki/Motor_vehicle), such as [cars](https://en.wikipedia.org/wiki/Car), [motorcycles](https://en.wikipedia.org/wiki/Motorcycle) and [buses](https://en.wikipedia.org/wiki/Bus). ABS operates by preventing the [wheels](https://en.wikipedia.org/wiki/Wheel) from locking up during [braking](https://en.wikipedia.org/wiki/Brake), thereby maintaining [tractive](https://en.wikipedia.org/wiki/Traction_(engineering)) contact with the road surface and allowing the driver to maintain more control over the vehicle.

ABS is an automated system that uses the principles of [threshold braking](https://en.wikipedia.org/wiki/Threshold_braking) and [cadence braking](https://en.wikipedia.org/wiki/Cadence_braking), techniques which were once practiced by skillful drivers before ABS was widespread. ABS operates at a much faster rate and more effectively than most drivers could manage. Although ABS generally offers improved vehicle control and decreases stopping distances on dry and some slippery surfaces, on loose gravel or snow-covered surfaces ABS may significantly increase [braking distance](https://en.wikipedia.org/wiki/Braking_distance), while still improving steering control.

A model for an Anti-Lock Braking System (ABS) is simulated using Simulink using the all the knowledge provided during the training period. It simulates the dynamic behavior of a vehicle under hard braking conditions. The model represents a single wheel, which may be replicated a number of times to create a model for a multi-wheel vehicle.

The signal logging feature in Simulink is used in the model. The model logs signals to the MATLAB workspace where we can analyze and view them. In this model, the wheel speed, vehicle speed and slip is calculated to analyze the model with given reference slip. This component is then referenced using a 'Model' block. Note that both the top model and the referenced model use a variable step solver, so Simulink will track zero-crossings in the referenced model.

The wheel rotates with an initial angular speed that corresponds to the vehicle speed before the brakes are applied. We used separate integrators to compute wheel angular speed and vehicle speed. We use two speeds to calculate slip, which is determined by Equation 1. Note that we introduce vehicle speed expressed as an angular velocity (see below).

**Equation 1**

From these expressions, we see that slip is zero when wheel speed and vehicle speed are equal, and slip equals one when the wheel is locked. A desirable slip value is 0.2 and 0.3, which means that the number of wheel revolutions equals 0.8 times the number of revolutions under non-braking conditions with the same vehicle velocity. This maximizes the adhesion between the tire and road and minimizes the stopping distance with the available friction.

**System Modeling**

Anti-Lock braking system is modeled using the Simulink software. Figure (1) explains the blocks used for the simulation and connections between them.

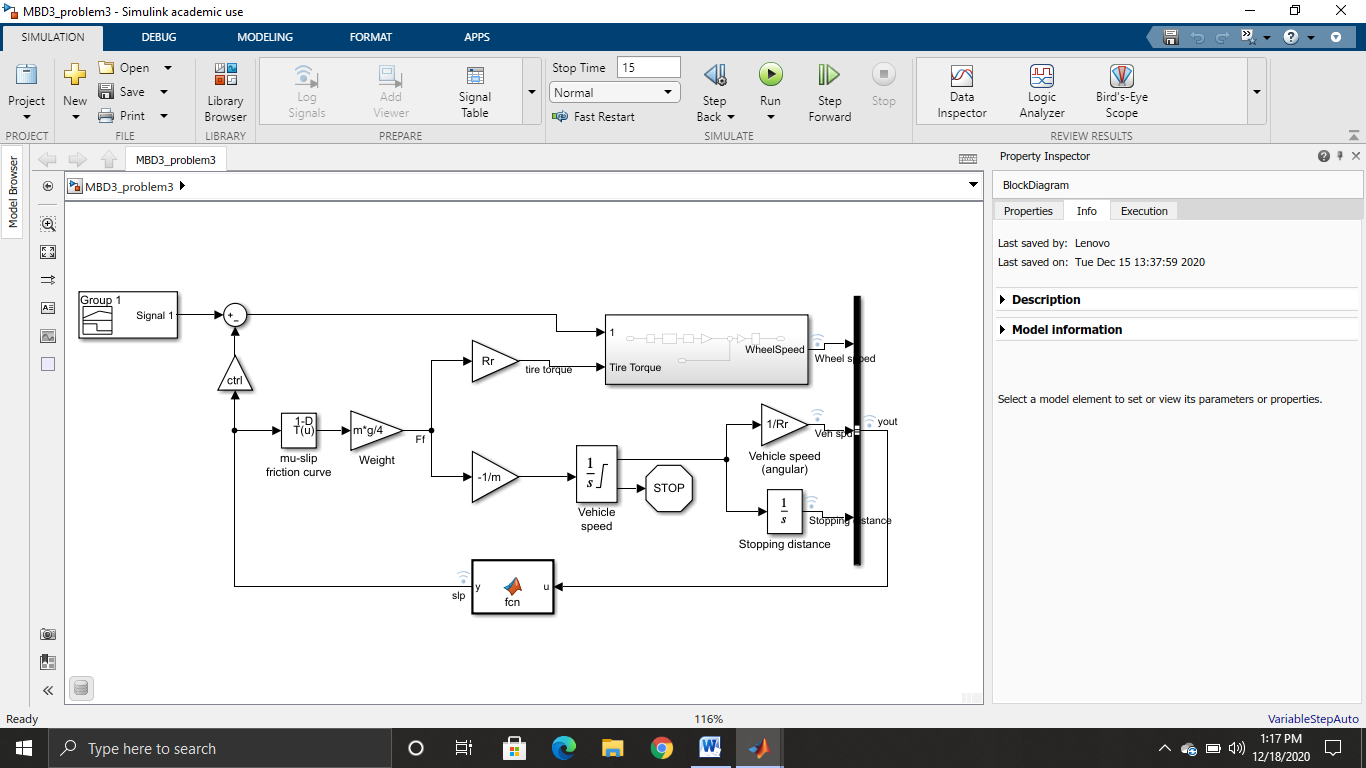
****

Figure 1: Model of ABS.

Initially signal builder is used to build a reference signal for the model. This reference signal is a desired slip required for the model to have ABS successfully achieved. Figure 2 depicts the signal used for the desired slip.

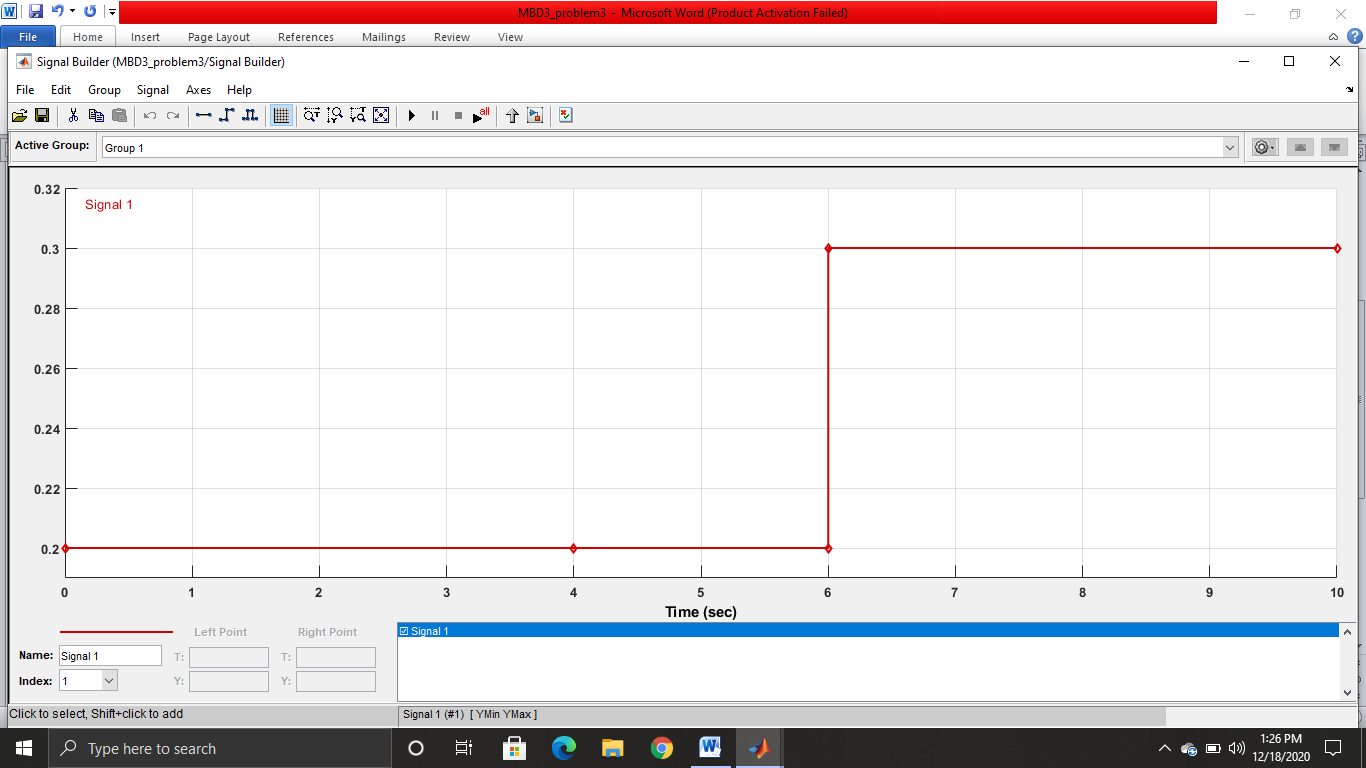


Figure 2: Signal built using signal builder.

In the signal builder I have used signal similar to step signal to make sure the signal works for different slip value. The signal here used with 0.2 and 0.3 desired slip. The model works perfectly for both the values of the slip. The signal values after 10 seconds continue with 0.3 till 15 seconds.

Next solver characteristics are changed based on the required settings. Since the model does not require much of accuracy. So the ode45 (Dormand-Prince) is used as a solver and the maximum step size for incrimination is 0.01 to have a smooth curve compared to automatic selection of the solver and the maximum step size. Figure 3 depicts the solver characteristics. The model is simulated for 15 seconds with two desired slip.

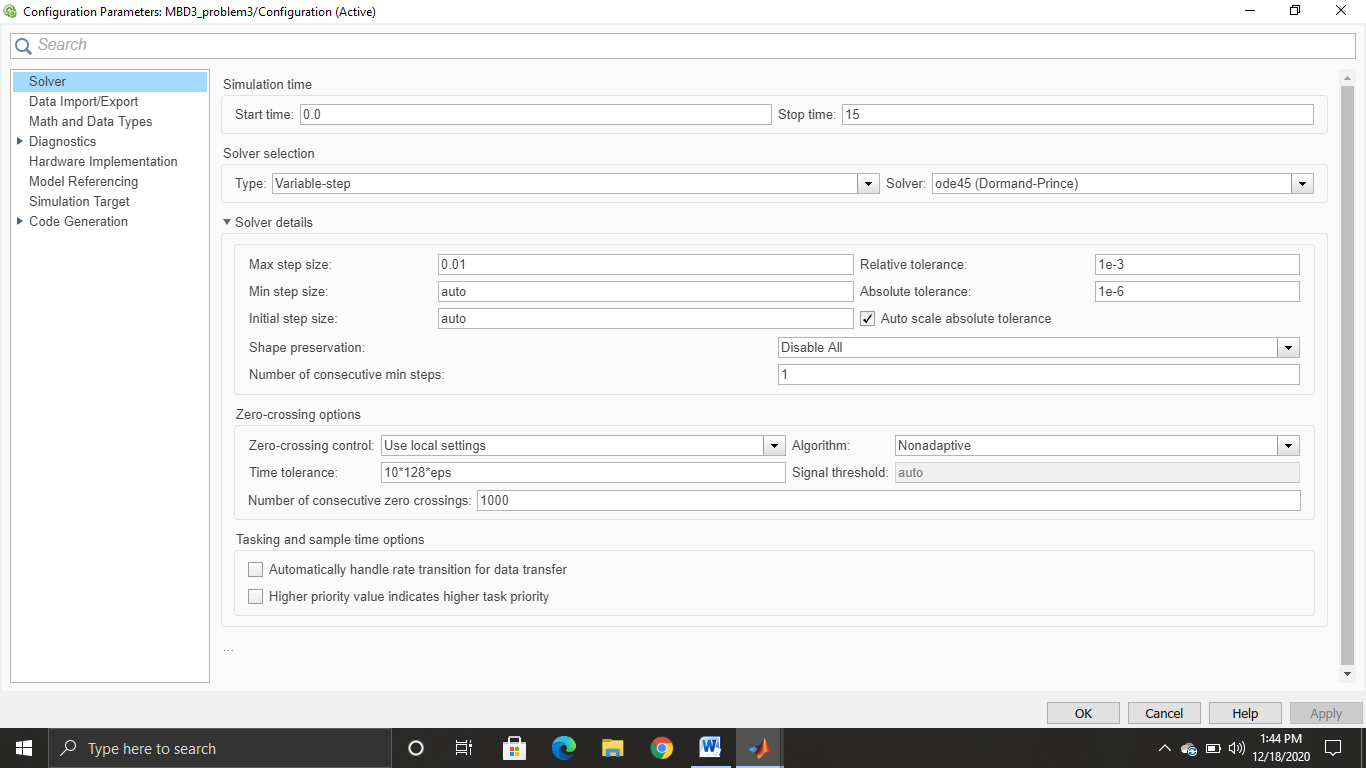


Figure 3: Solver characteristics

To declare the constants in the model I have used callback function to name and declare the variable with the required number. The call back function is used in the property inspector in initial function. Also the data is been linked to the data dictionary to see the variables modified with time and time. It will be useful in the projects to have a modified data. Figure 4 shows some of the constants declared in the model.

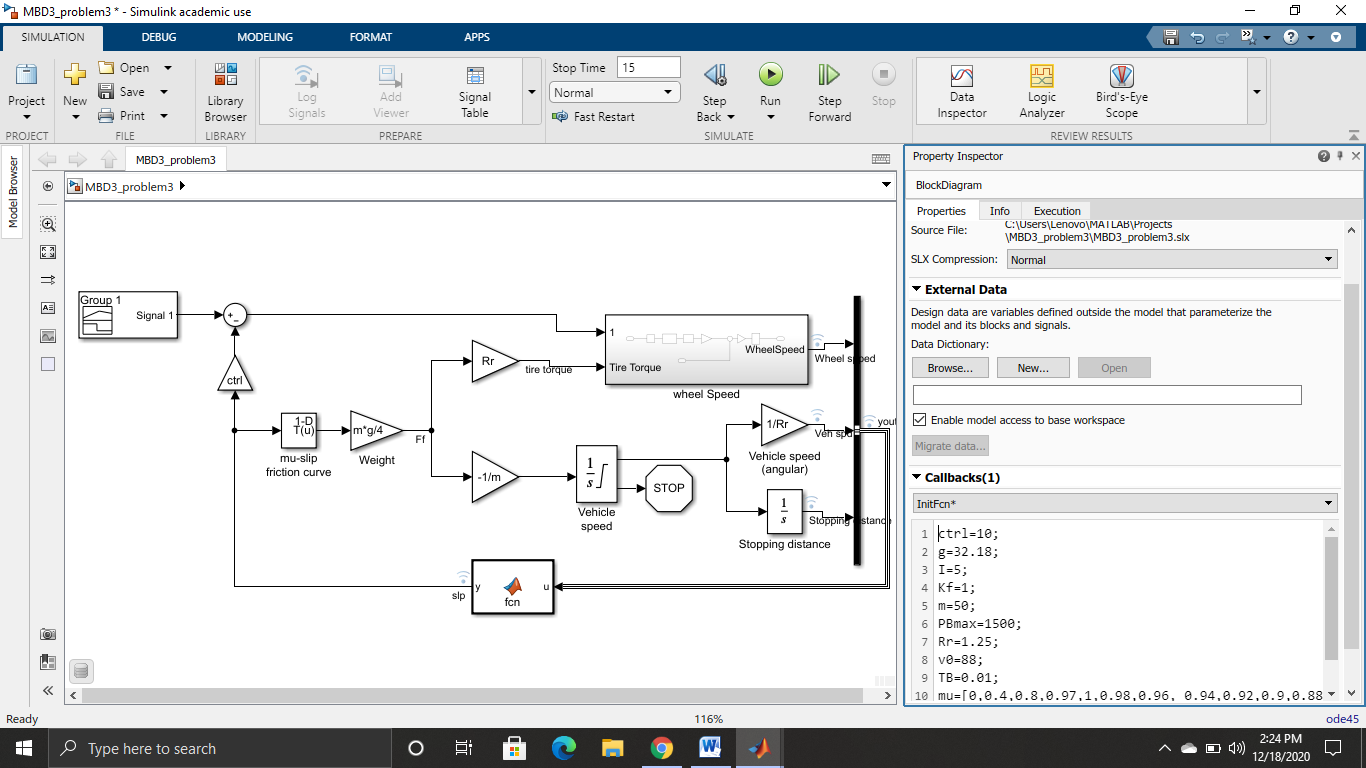


Figure 4: Data declared through call back function.

To calculate the wheel speed, wheel speed subsystem is created with tire torque and error signal as input and wheel speed as output. Figure 5 explains the model used in the wheel speed using the bang-bang controller and other blocks.

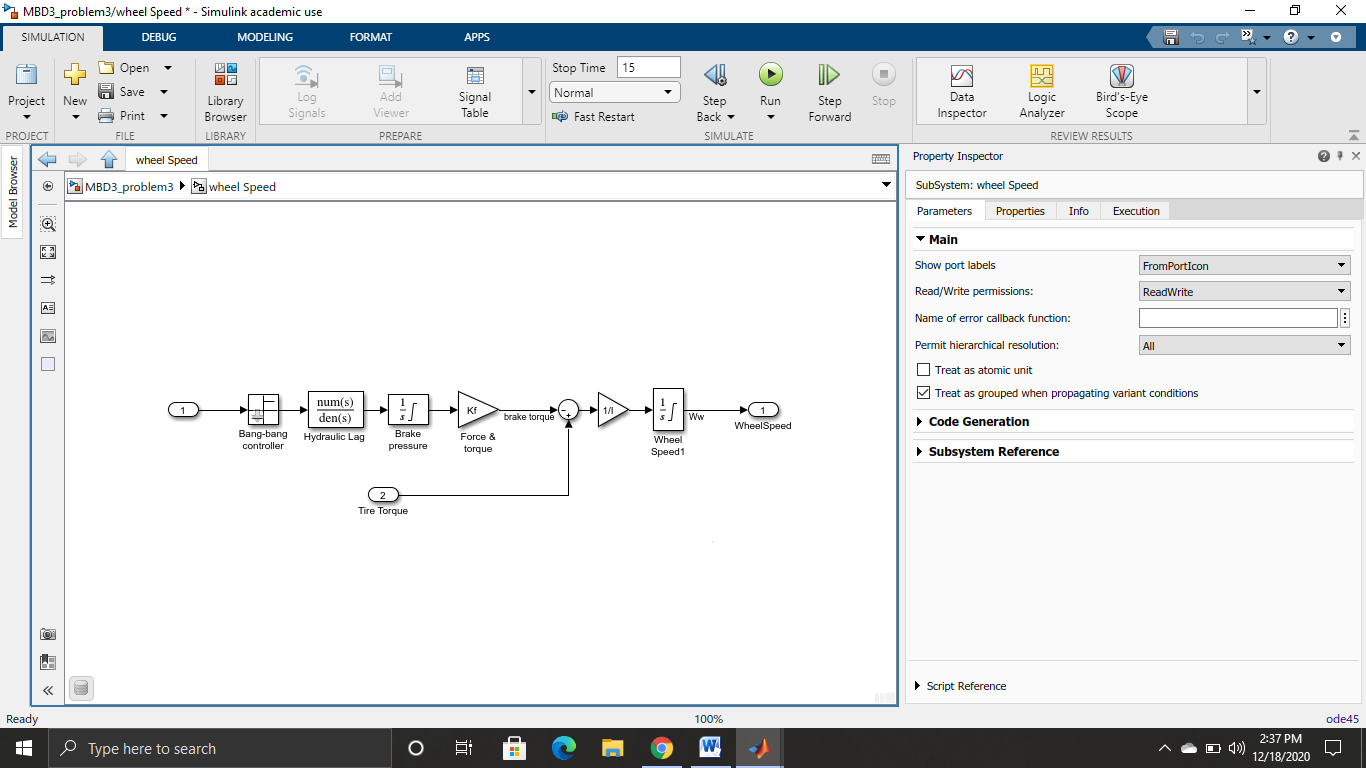


Figure 5: Subsystem of wheel speed

Here sub-system uses the Bang- bang controller. To control the rate of change of brake pressure, the model subtracts actual slip from the desired slip and feeds this signal into a bang-bang control (+1 or -1, depending on the sign of the error). This on/off rate passes through a first-order lag that represents the delay associated with the hydraulic lines of the brake system. The model then integrates the filtered rate to yield the actual brake pressure. The resulting signal, multiplied by the piston area and radius with respect to the wheel (Kf), is the brake torque applied to the wheel.

The model multiplies the frictional force on the wheel by the wheel radius (Rr) to give the accelerating torque of the road surface on the wheel. The brake torque is subtracted to give the net torque on the wheel. Dividing the net torque by the wheel rotational inertia (I), yields the wheel acceleration, which is then integrated to provide wheel velocity. In order to keep the wheel speed and vehicle speed positive, limited integrators are used in this model.

Matlab function is used to calculate the slip of the wheel. Here, output of the system is given to the function and the expression is written in the function using C-code. Figure 6 shows the matlab code written as a function to calculate slip.

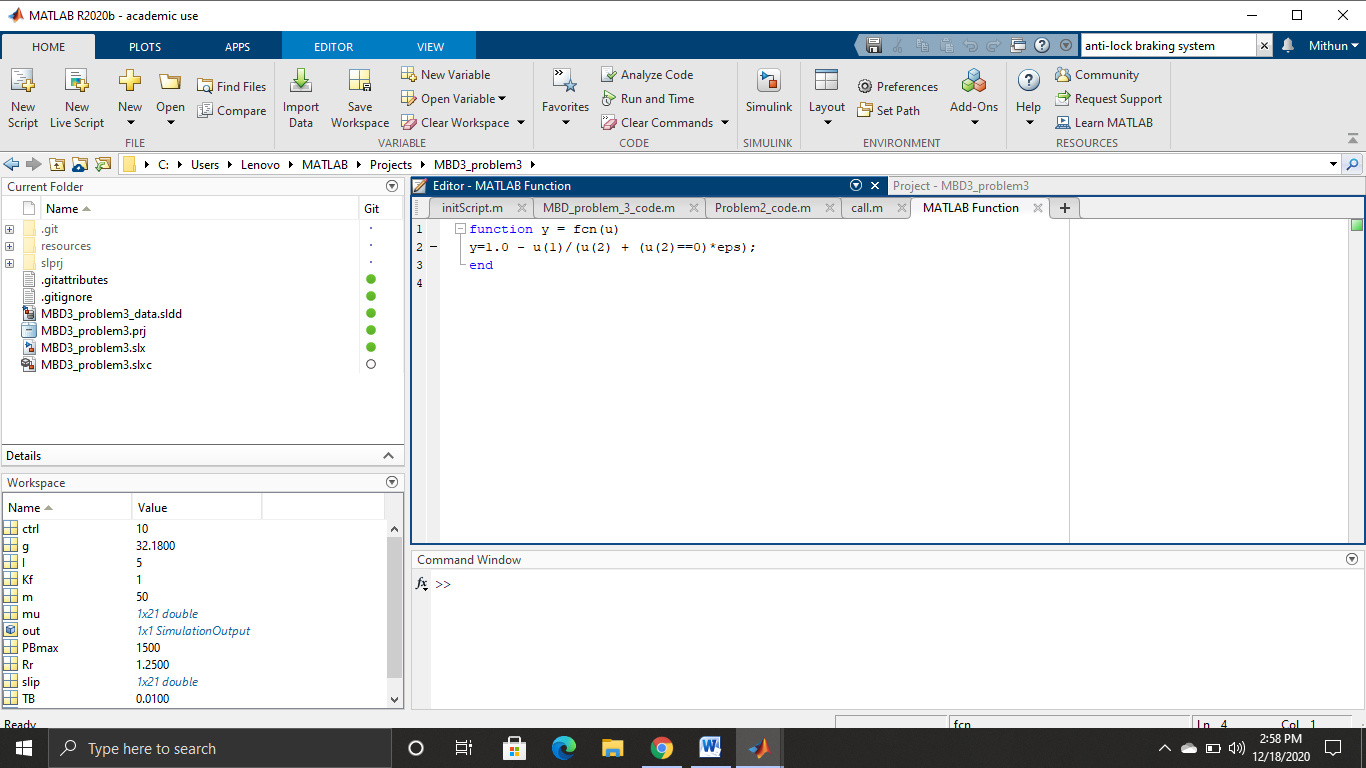
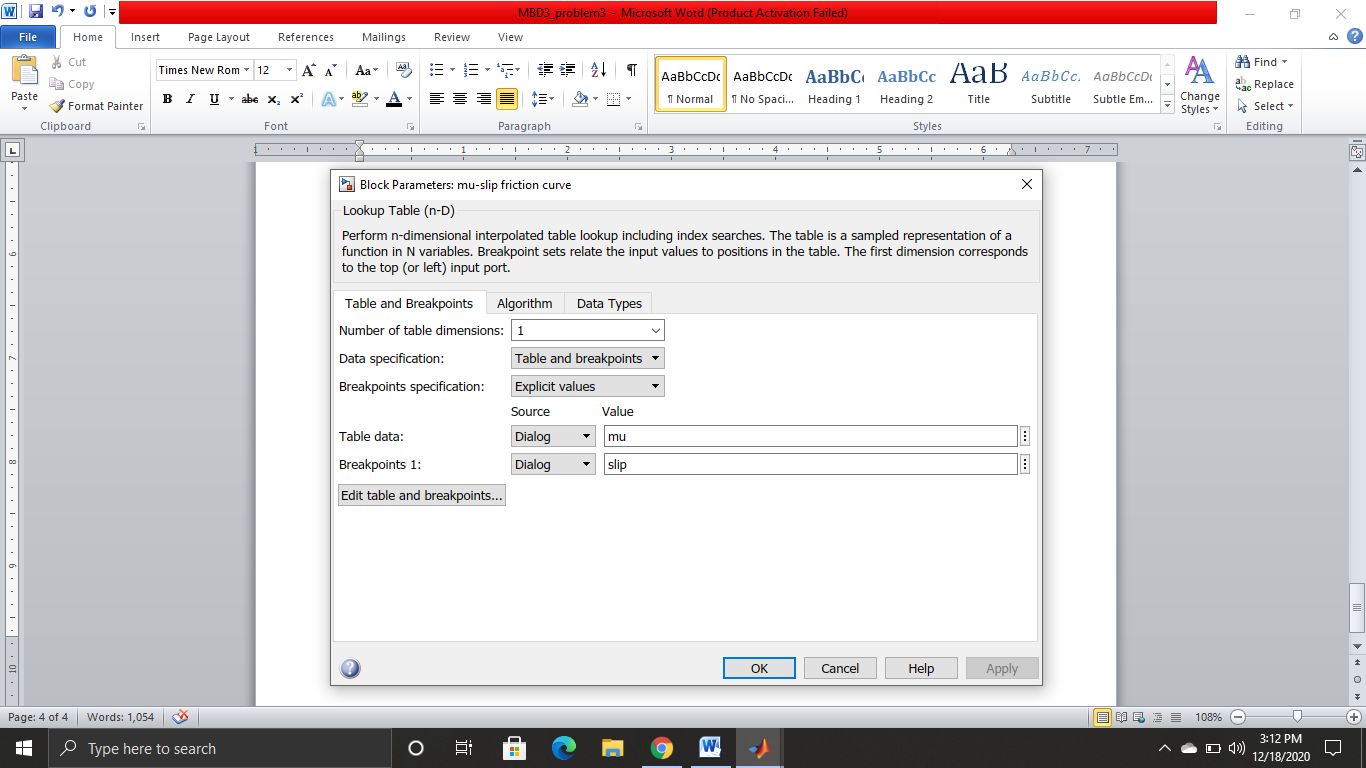


Figure 6: Matlab code to calculate slip

To graphical curve with mu- slip 1-D Loop-up table is used based on the valued stored in the data dictionary. Using this curve values the data has been used. Figure 7 is the screenshot of the block parameter of the 1-D look-up table and figure 8 is the values stored in dictionary for mu and slip.



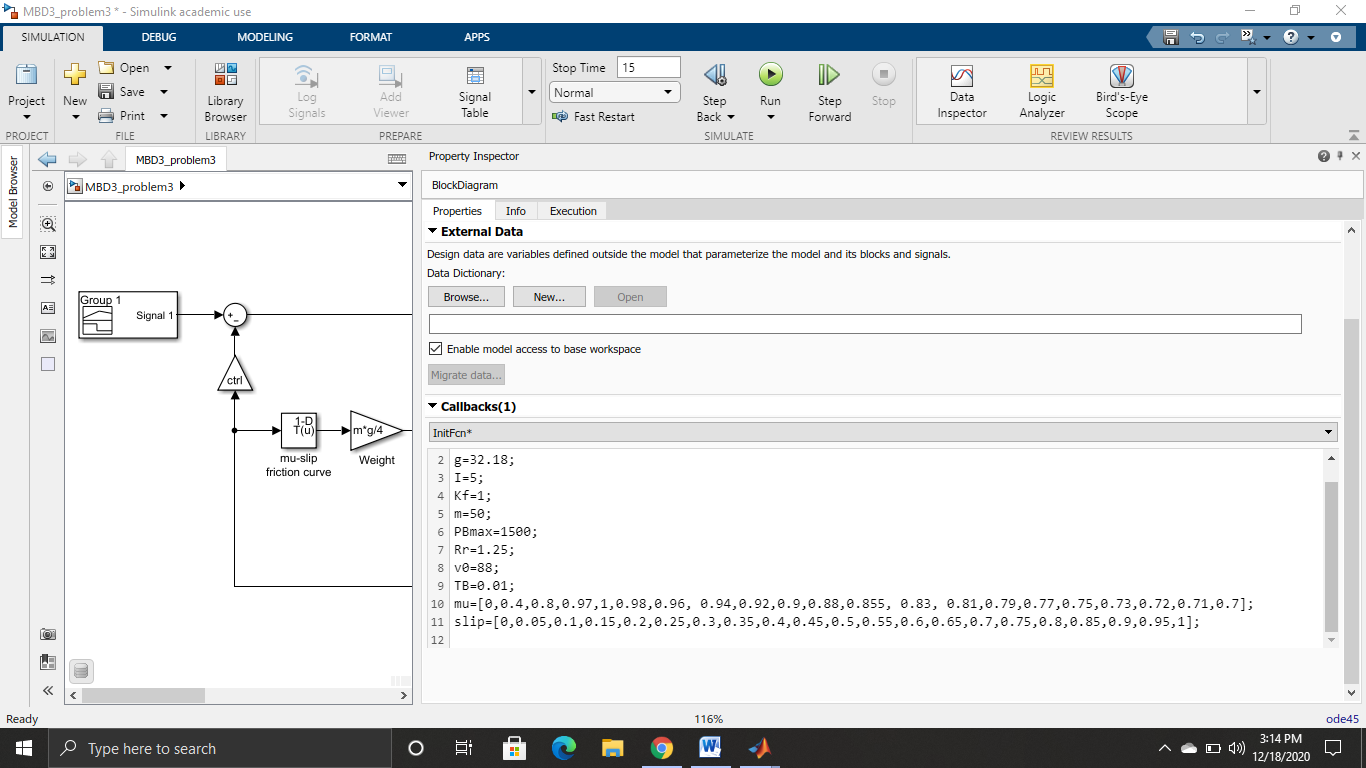


Figure 7: block parameter for 1-D look-up table

Figure 8: values of mu and slip

**Results and Conclusion**

The model is simulated using the parameters explained above with simulation time of 15 seconds. Initially slip is calculated for the desired slip value of 0.2 and after 6 seconds desired slip value is changed to 0.3, to test whether the model work for all the desired slip values.

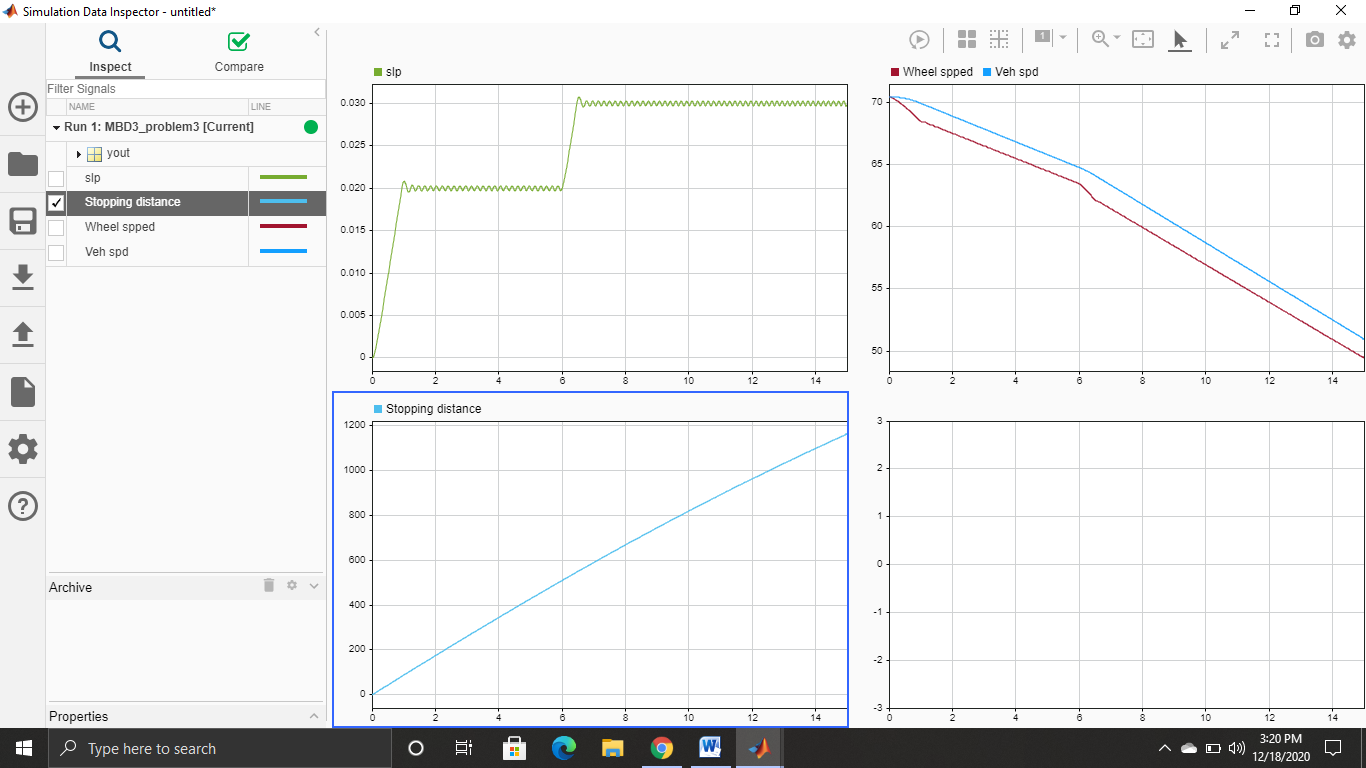


Figure 9: Slip obtained based on the desired slip value.

Simulation of the model is used have the vehicle speed and the wheel speed based on the model. Here wheel speed is slow when compared to the vehicle speed. So the at the slip change we can have a look at the slight deep in the wheel speed but not in the vehicle speed.

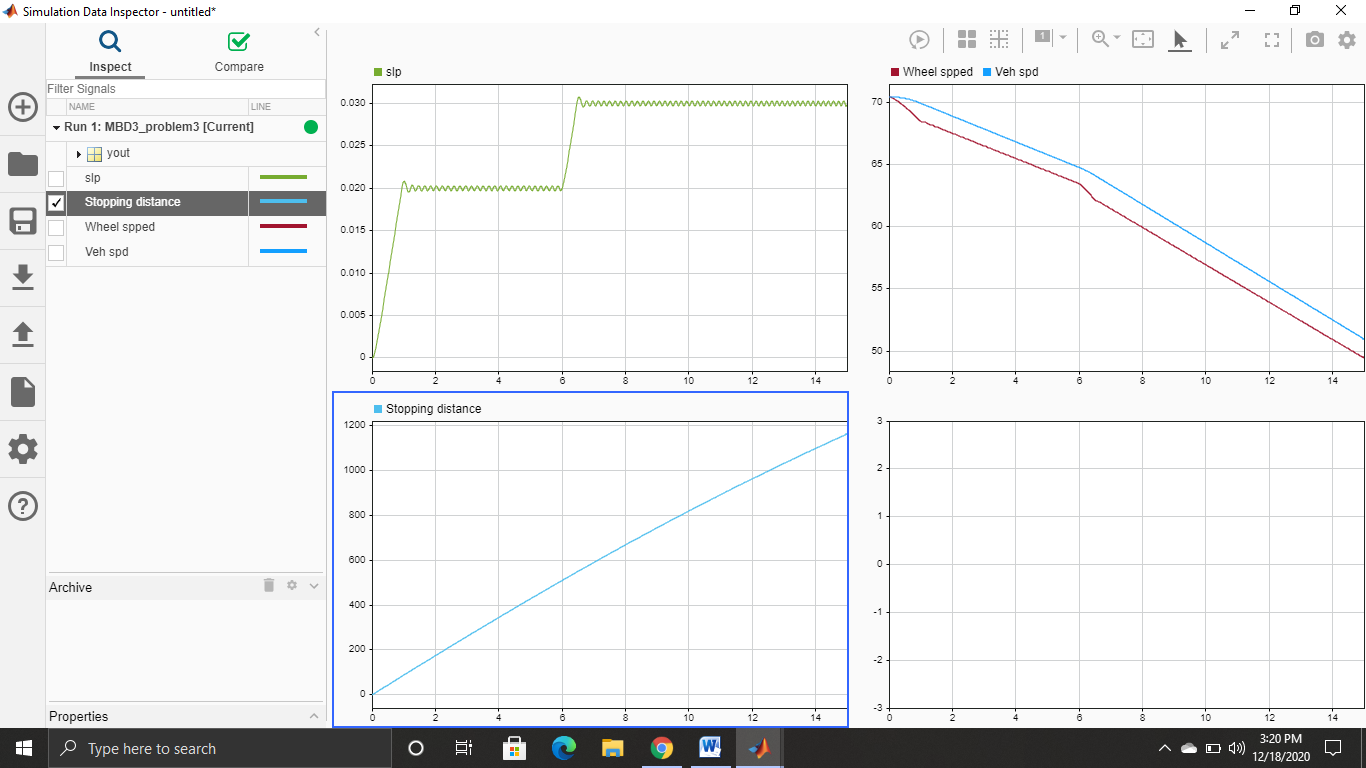


Figure 10: vehicle speed and wheel speed.

Next we have the stopping distance simulation which shows when the vehicle gets stops after applying a hard brake. This comparison can be done with the stopping distance for the model with no ABS. By this we can have different stopping distance for different situation.

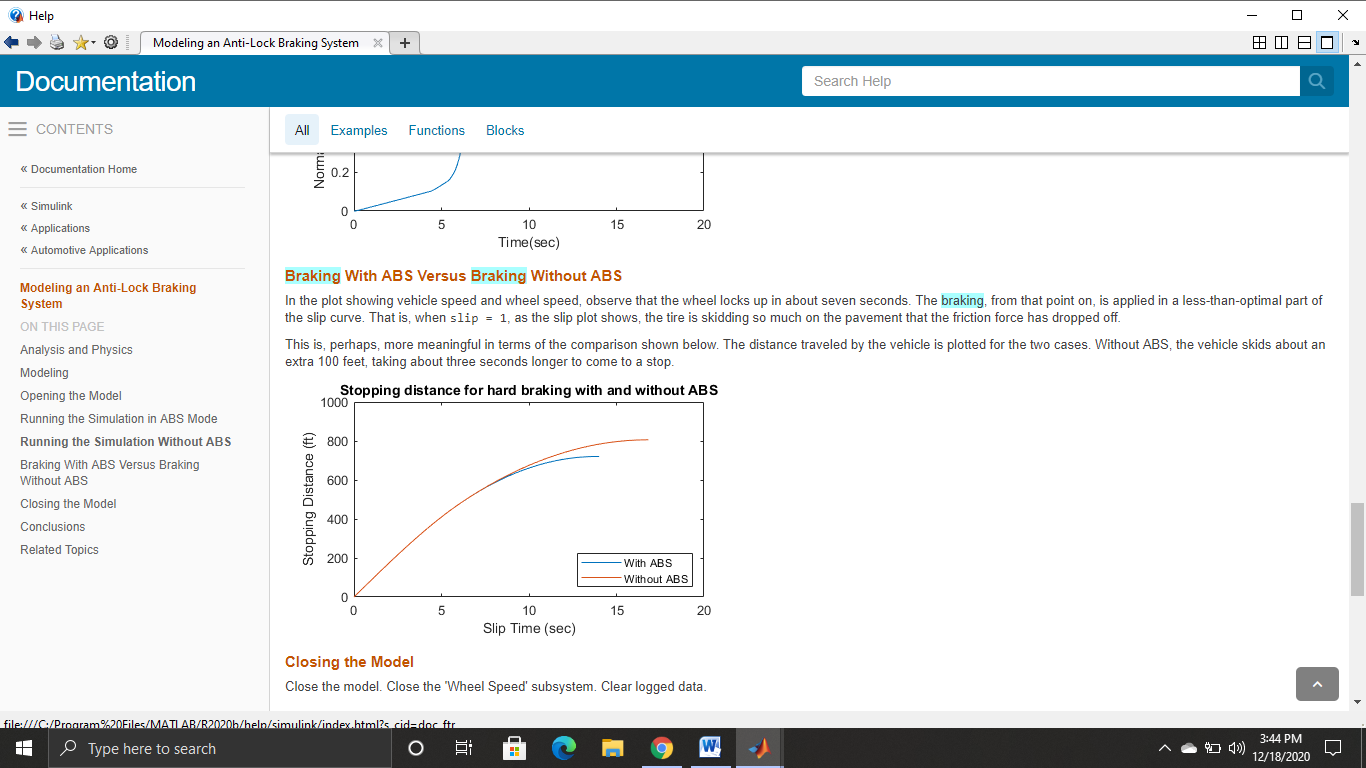


Figure 11: Comparison of stopping distance with ABS and without ABS.

**Conclusion**

This model simulates a braking system under the action of an ABS controller. The controller in this model is idealized, but you can use any proposed control algorithm in its place to evaluate the system's performance. You can also use the Simulink Code with Simulink as a valuable tool for rapid prototyping of the proposed algorithm. C code is generated and compiled for the controller hardware to test the concept in a vehicle. This significantly reduces the time needed to prove new ideas by enabling actual testing early in the development cycle.

For a hardware-in-the-loop braking system simulation, you can remove the 'bang-bang' controller and run the equations of motion on real-time hardware to emulate the wheel and vehicle dynamics. You can do this by generating real-time C code for this model using the Simulink Coder. You can then test an actual ABS controller by interfacing it to the real-time hardware, which runs the generated code. In this scenario, the real-time model would send the wheel speed to the controller, and the controller would send brake action to the model.