**Antilock Braking**

**System**

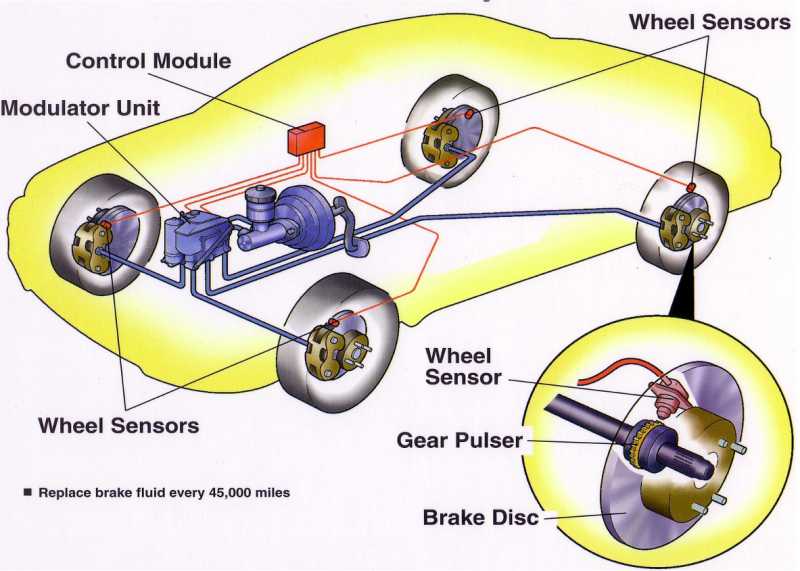
Name: Kanchan Anand Dighole

Emp Id:214856

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  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), [trucks](https://en.wikipedia.org/wiki/Truck), 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.

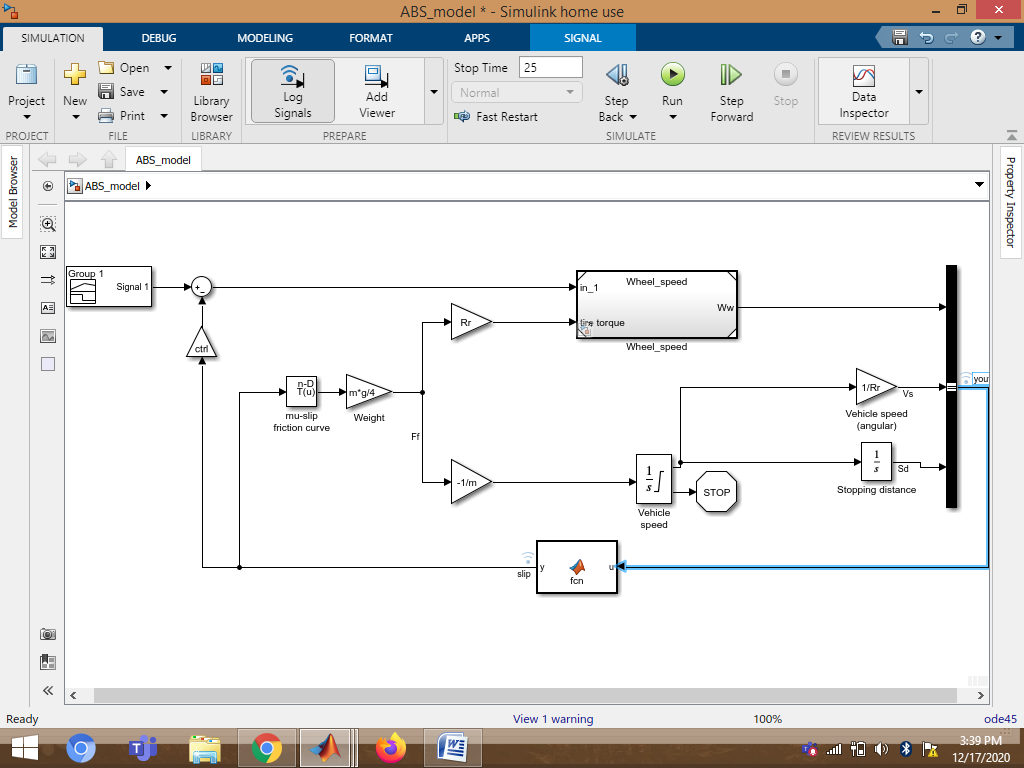
It is an automated system that uses the principles of [threshold braking](https://en.wikipedia.org/wiki/Threshold_braking) and [cadence braking](https://en.wikipedia.org/wiki/Cadence_braking), techniques ABS operates at a much faster rate and more effectively than most drivers could manage. Although 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. Modern versions may not only prevent wheel lock under braking, but may also alter the front-to-rear brake bias.



Antilock Braking system

**2. Modeling ABS in Simulink**

This is a simple model for an Anti-Lock Braking System (ABS). 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. In this model, the wheel speed is calculated in a separate model named wheel speed.



Modeling in Simulink

### 3.Analysis:

### 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

$$\omega_v = \frac{V}{R} \mbox{ (equals the wheel angular speed if there is no slip)}$$

Equation 1

$$slip=1-\frac{\omega_w}{\omega_v}$$

$$ \omega_v = \frac{V_v}{R_r}$$$$\omega_v = \mbox{ vehicle speed divided by wheel radius}$$

$$ V_v = \mbox{ vehicle linear velocity}$$

$$ R_r = \mbox{ wheel radius}$$

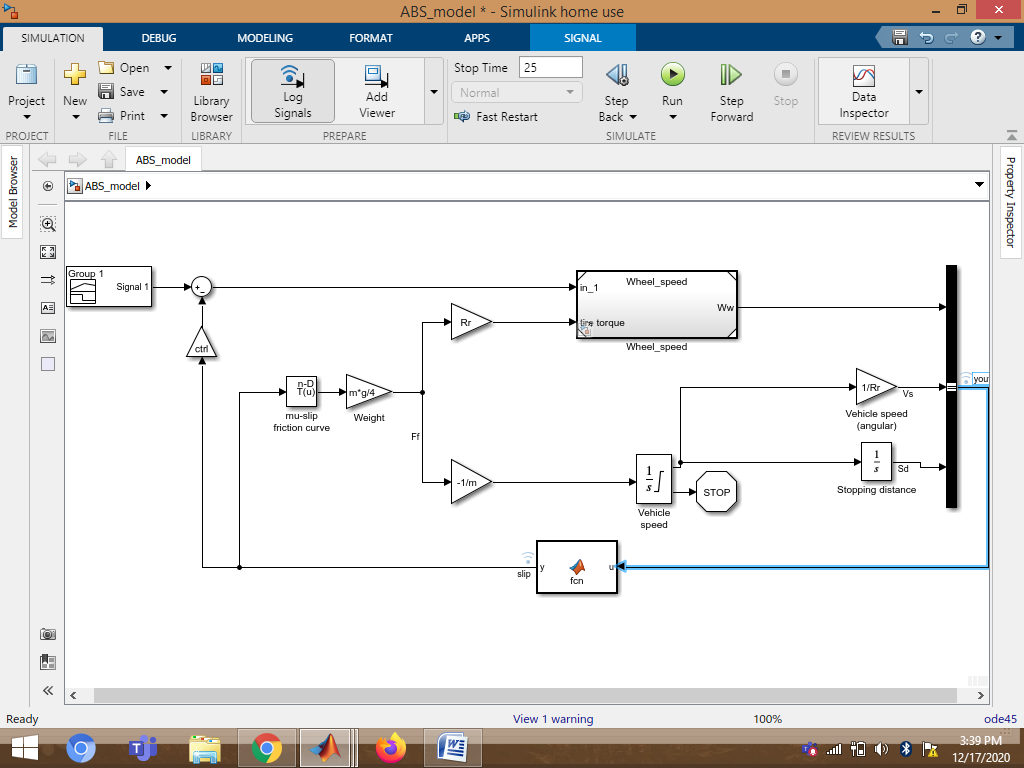
$$ \omega_w = \mbox{ wheel angular velocity}$$

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, 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.

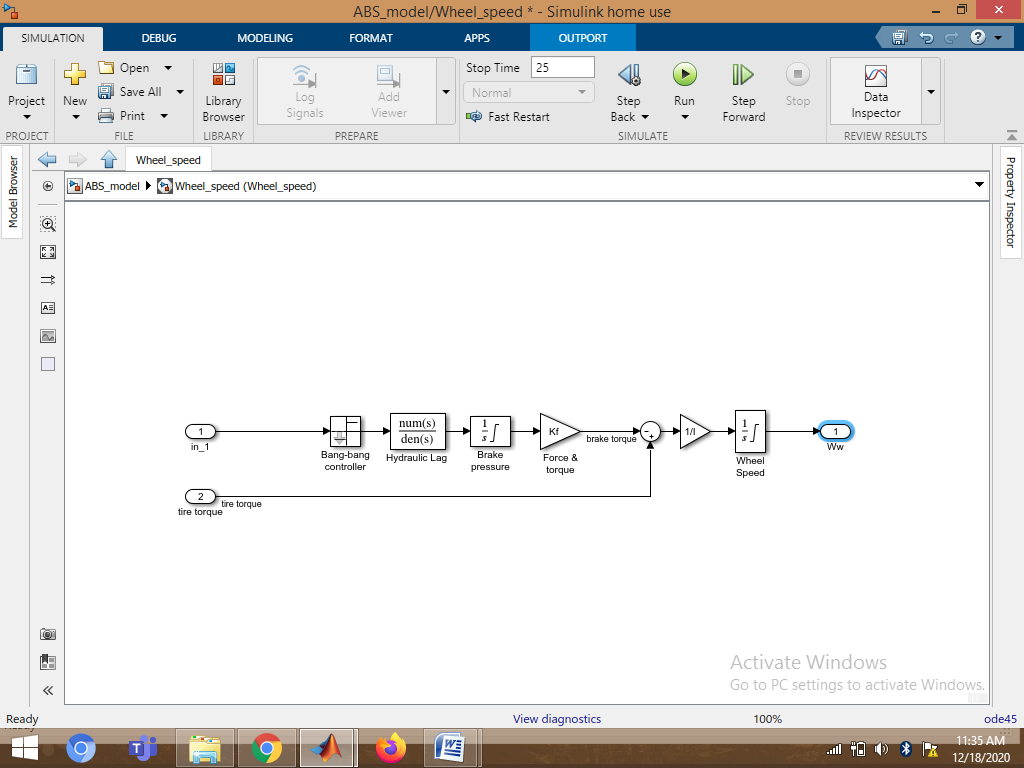
**4.Modeling:**

The friction coefficient between the tire and the road surface, mu, is an empirical function of slip, known as the mu-slip curve. We created mu-slip curves by passing MATLAB variables into the block diagram using a Simulink lookup table. The model multiplies the friction coefficient, mu, by the weight on the wheel, W, to yield the frictional force, Ff, acting on the circumference of the tire. Ff is divided by the vehicle mass to produce the vehicle deceleration, which the model integrates to obtain vehicle velocity.

In this model, we used an ideal anti-lock braking controller that uses 'bang-bang' control based upon the error between actual slip and desired slip. We set the desired slip to the value of slip at which the mu-slip curve reaches a peak value, this being the optimum value for minimum braking distance.



ABS Model

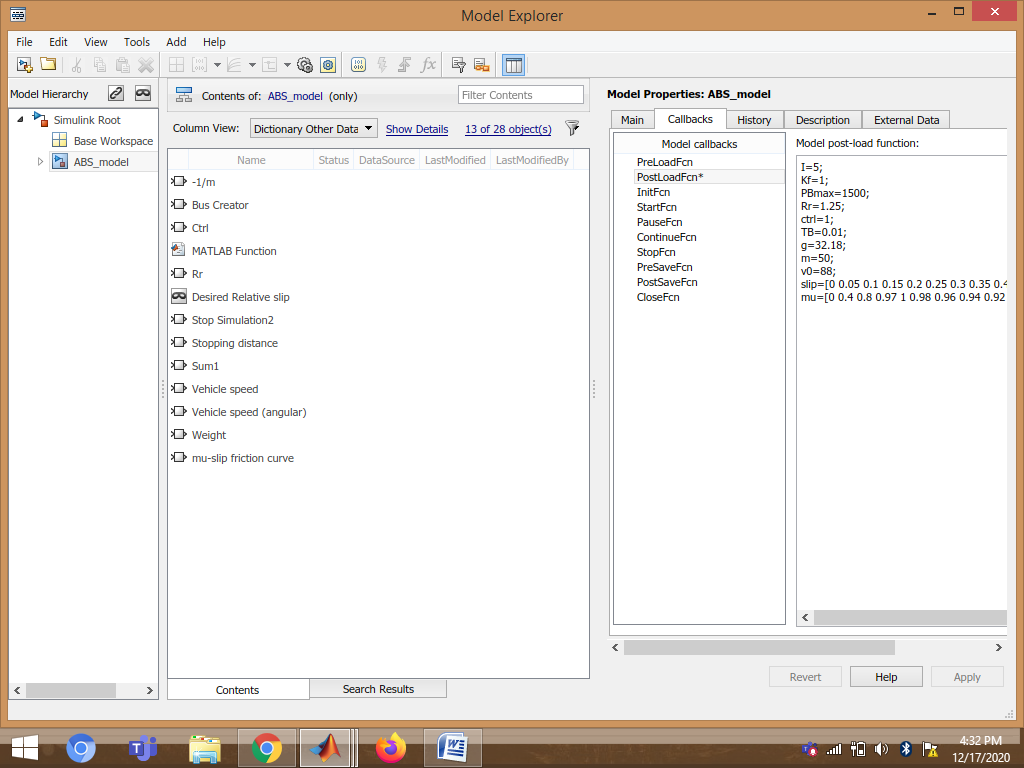


Wheel Speed subsystem

**5.Skills used in model:**

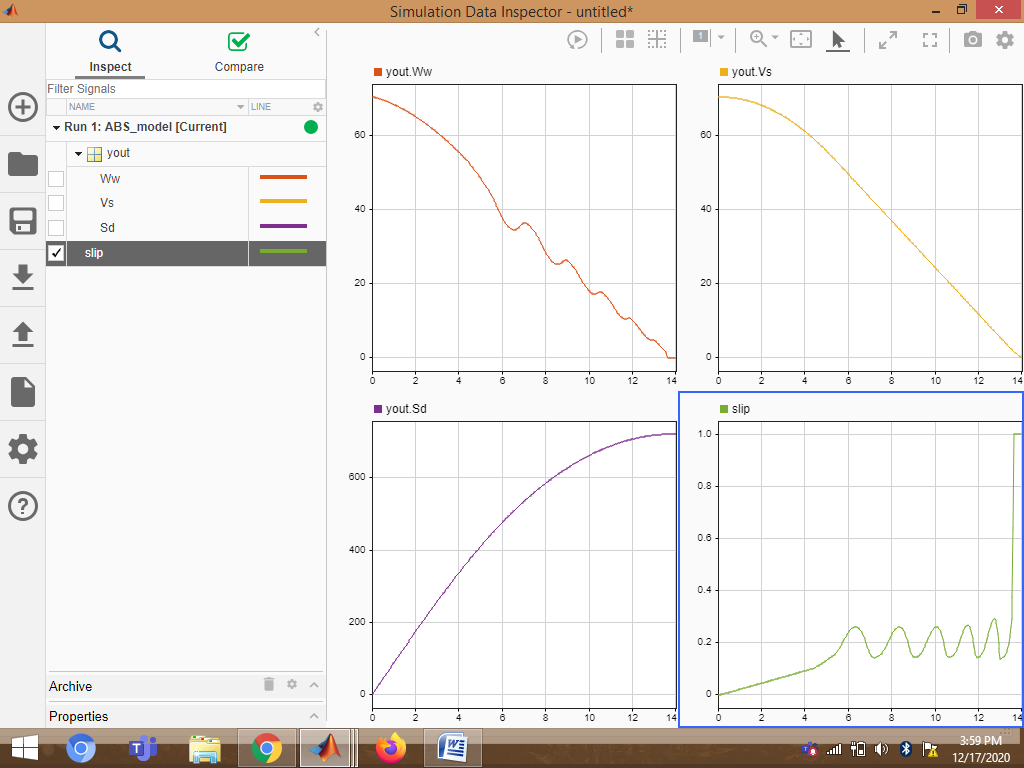
1. Callback:

In a model hierarchy, the execution of callbacks reflects the order in which the top model and the models it references execute their callbacks. In this model I have used PostLoadFcn callback that creates variable in MATLAB workspace.



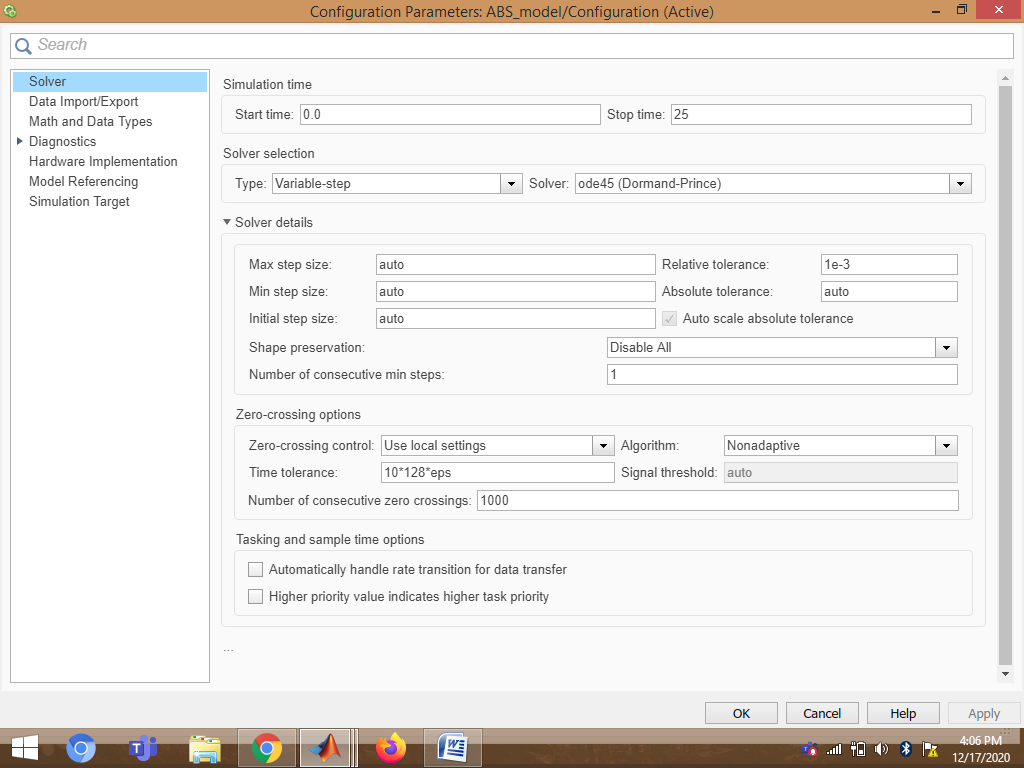
1. Data Inspector:

Data inspector is used to inspect and compare data and simulation results to validate and iterate model design.



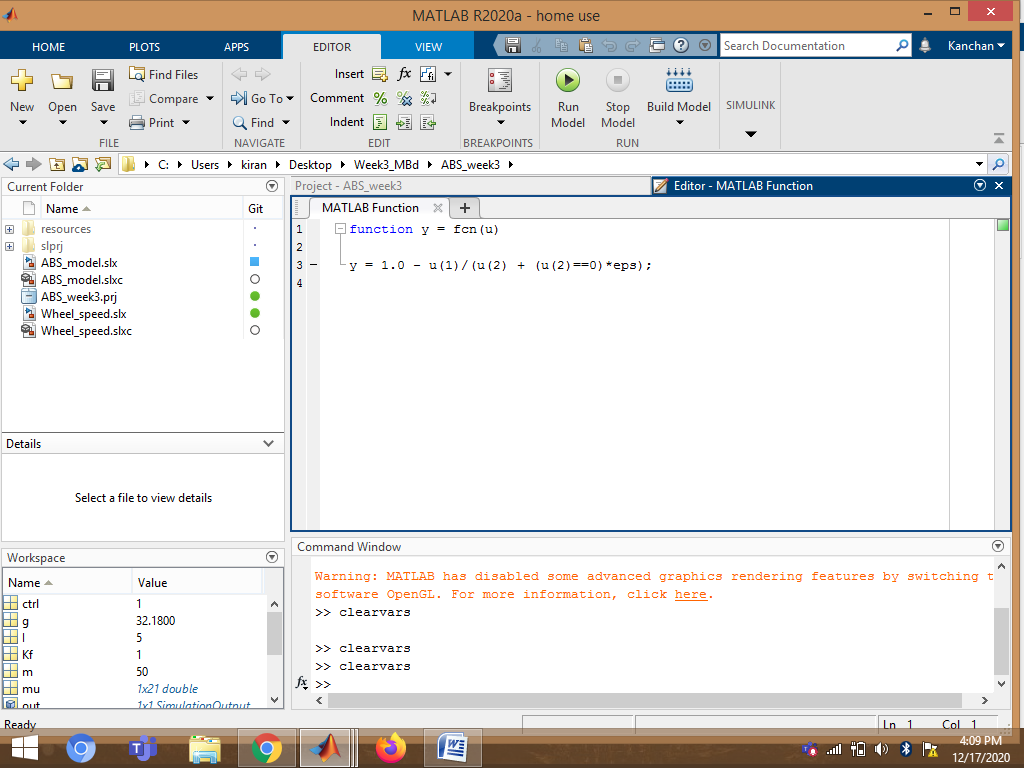
1. Solver selection strategy:

In this model I have used ode45 solver.ode45 performs well with most ODE problems and should generally be first choice of solver. It is used for Non-Stiff problems and it provides medium level accuracy.



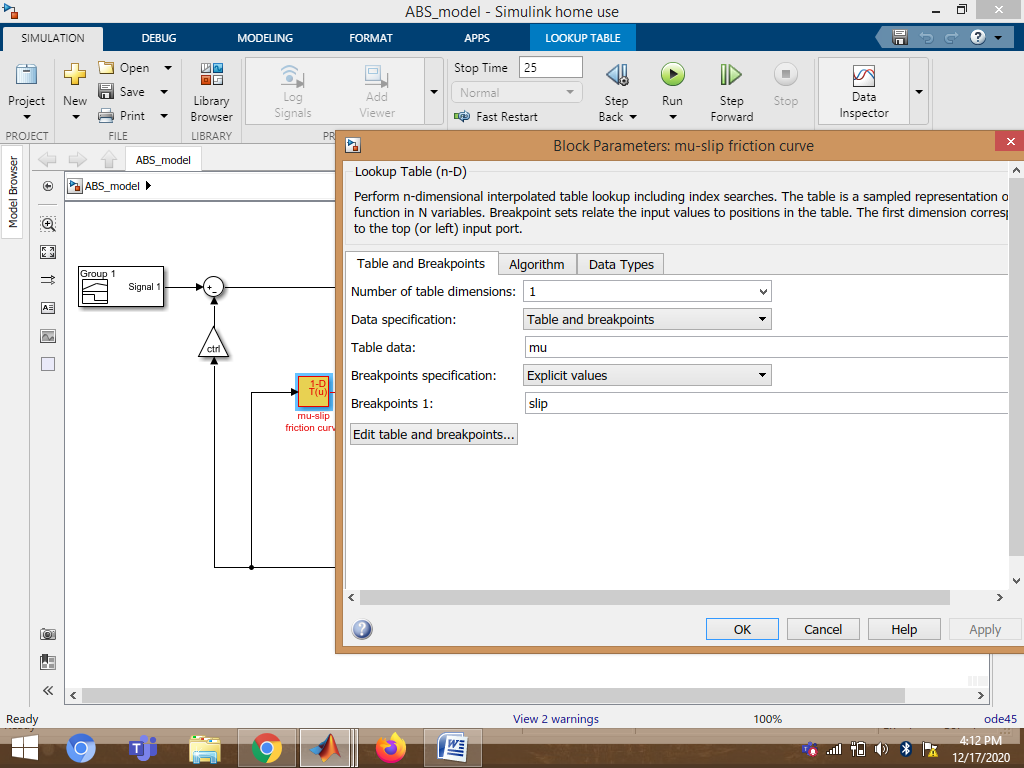
1. MATLAB function block:

MATLAB Function block is used to implement MATLAB functions to Simulink models to deploy code and embed code in processors.  MATLAB Function block is used to generate readable, efficient, and compact C/C++ code for deployment to desktop and embedded applications

.

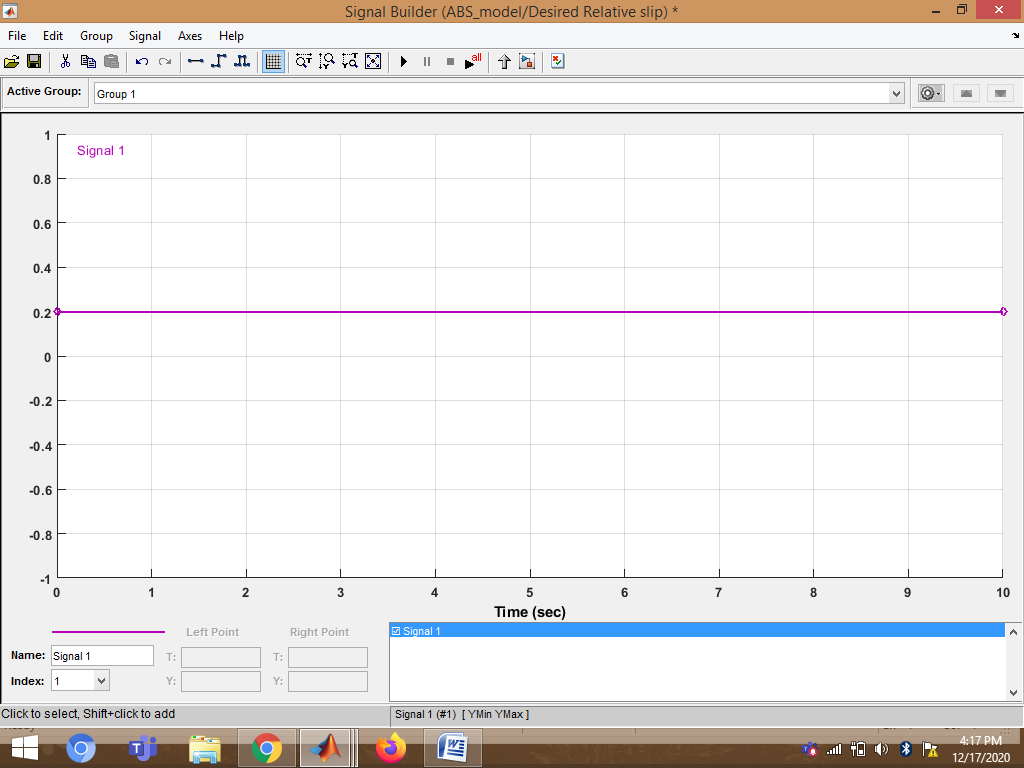
1. Look-Up Table:

A lookup table block uses an array of data to map input values to output values, approximating a mathematical function. Given input values, Simulink performs a “lookup” operation to retrieve the corresponding output values from the table In this model I have used 1-D Lookup table for mu-slip friction curve.



1. Signal Builder:

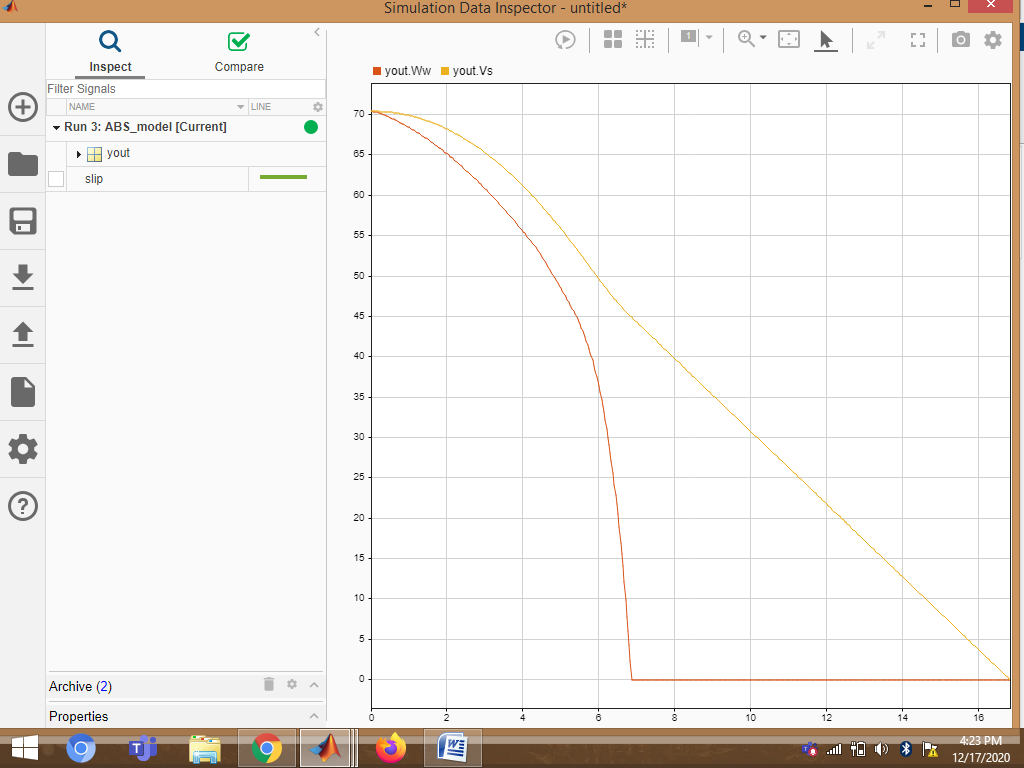
In this model, I have used signal builder to provide desired relative slip as input.



**6.Observations:**

1.Vehicle behavior without ABS:

Model variable ctrl = 0. This disconnects the slip feedback from the controller, resulting in maximum braking.



2.Vehicle behavior with ABS:

Model variable ctrl = 1. This connects the slip feedback from the controller.

