**ANTI – LOCK BREAKING SYSTEM**

****

**CONTENTS**

1. Introduction to ABS
2. Operation of ABS
3. Simulink Model developed
4. Sub-system Model
   * 1. Callback Function
     2. MATLAB Function
     3. 1-D Lookup table
     4. Data Inspector
     5. Solver
     6. Signal Builder
5. Sources

**Anti-Lock Braking System**

**Introduction to ABS**



**Operation of ABS**

The anti-lock brake controller is also known as the CAB (Controller Anti-lock Brake).

Typically ABS includes a central [electronic control unit](https://en.wikipedia.org/wiki/Electronic_control_unit) (ECU), four [wheel speed sensors](https://en.wikipedia.org/wiki/Wheel_speed_sensor), and at least two hydraulic valves within the brake [hydraulics](https://en.wikipedia.org/wiki/Hydraulic_brake). The ECU constantly monitors the [rotational speed](https://en.wikipedia.org/wiki/Rotational_speed) of each wheel; if it detects the wheel rotating significantly slower than the speed of the vehicle, a condition indicative of impending wheel lock, it actuates the valves to reduce hydraulic pressure to the brake at the affected wheel, thus reducing the braking force on that wheel; the wheel then turns faster. Conversely, if the ECU detects a wheel turning significantly faster than the others, brake hydraulic pressure to the wheel is increased so the braking force is reapplied, slowing down the wheel. This process is repeated continuously and can be detected by the driver via brake pedal pulsation. Some anti-lock systems can apply or release braking pressure 15 times per second. Because of this, the wheels of cars equipped with ABS are practically impossible to lock even during panic braking in extreme conditions.

The ECU is programmed to disregard differences in wheel rotative speed below a critical threshold because when the car is turning, the two wheels towards the centre of the curve turn slower than the outer two. For this same reason, a [differential](https://en.wikipedia.org/wiki/Differential_(mechanics)) is used in virtually all roadgoing vehicles.

If a fault develops in any part of the ABS, a warning light will usually be illuminated on the vehicle instrument panel, and the ABS will be disabled until the fault is rectified.

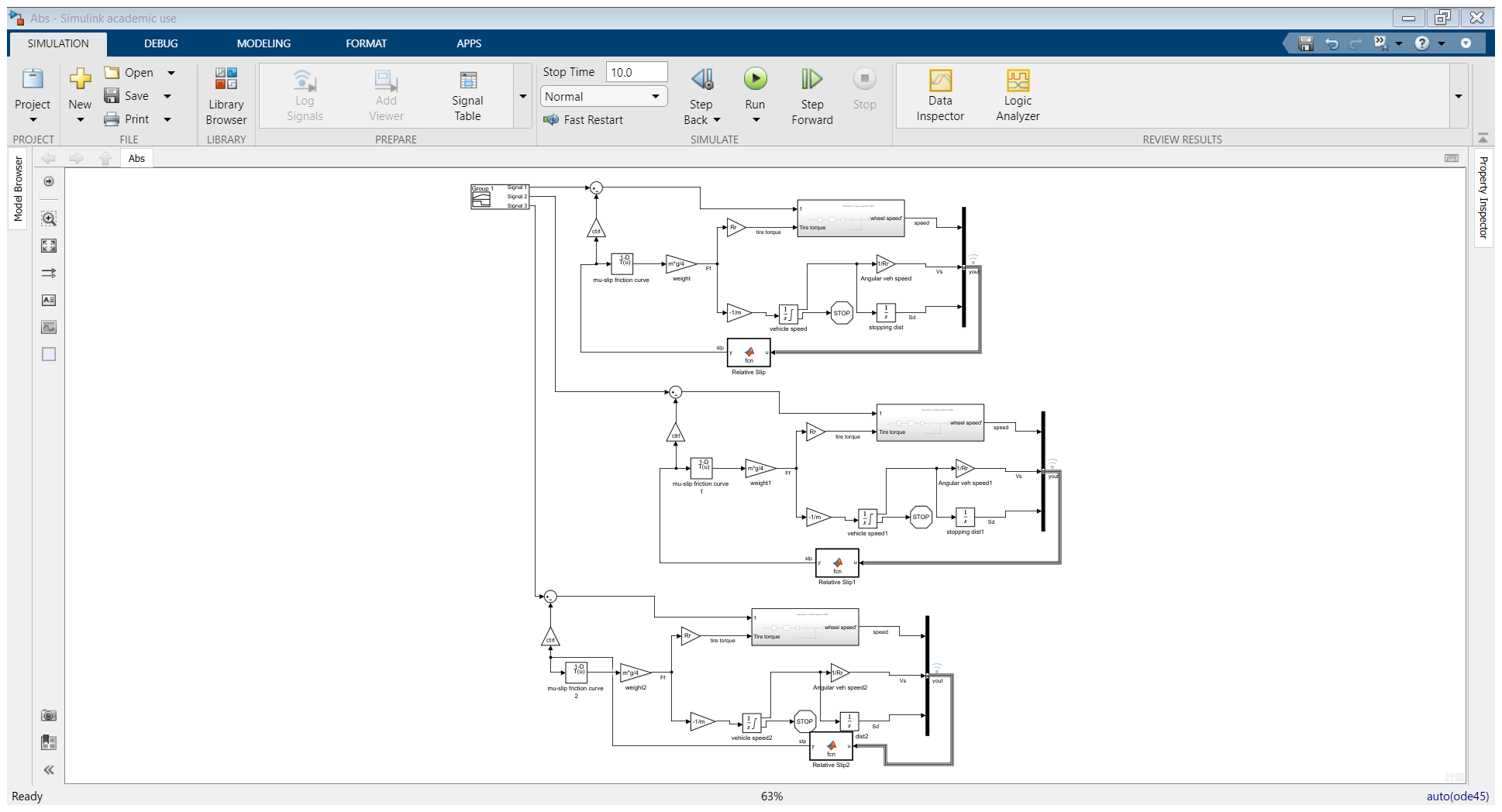
Modern ABS apply individual brake pressure to all four wheels through a control system of hub-mounted sensors and a dedicated [micro-controller](https://en.wikipedia.org/wiki/Micro-controller). ABS is offered or comes standard on most road vehicles produced today and is the foundation for electronic stability control systems, which are rapidly increasing in popularity due to the vast reduction in the price of vehicle electronics over the years.[[27]](https://en.wikipedia.org/wiki/Anti-lock_braking_system#cite_note-absfaq-27)

Modern electronic stability control systems are an evolution of the ABS concept. Here, a minimum of two additional sensors are added to help the system work: these are a [steering wheel](https://en.wikipedia.org/wiki/Steering_wheel) angle sensor and a [gyroscopic](https://en.wikipedia.org/wiki/Gyroscope) sensor. The theory of operation is simple: when the gyroscopic sensor detects that the direction taken by the car does not coincide with what the steering wheel sensor reports, the ESC software will brake the necessary individual wheel(s) (up to three with the most sophisticated systems), so that the vehicle goes the way the driver intends. The steering wheel sensor also helps in the operation of [Cornering Brake Control](https://en.wikipedia.org/wiki/Cornering_Brake_Control) (CBC), since this will tell the ABS that wheels on the inside of the curve should brake more than wheels on the outside, and by how much.

ABS equipment may also be used to implement a [traction control system](https://en.wikipedia.org/wiki/Traction_control_system) (TCS) on the acceleration of the vehicle. If, when accelerating, the tire loses traction, the ABS controller can detect the situation and take suitable action so that traction is regained. More sophisticated versions of this can also control throttle levels and brakes simultaneously.

The speed sensors of ABS are sometimes used in indirect [tire pressure monitoring system](https://en.wikipedia.org/wiki/Tire_pressure_monitoring_system) (TPMS), which can detect under-inflation of the tire(s) by the difference in the rotational speed of wheels.

**SIMULINK Model Developed**

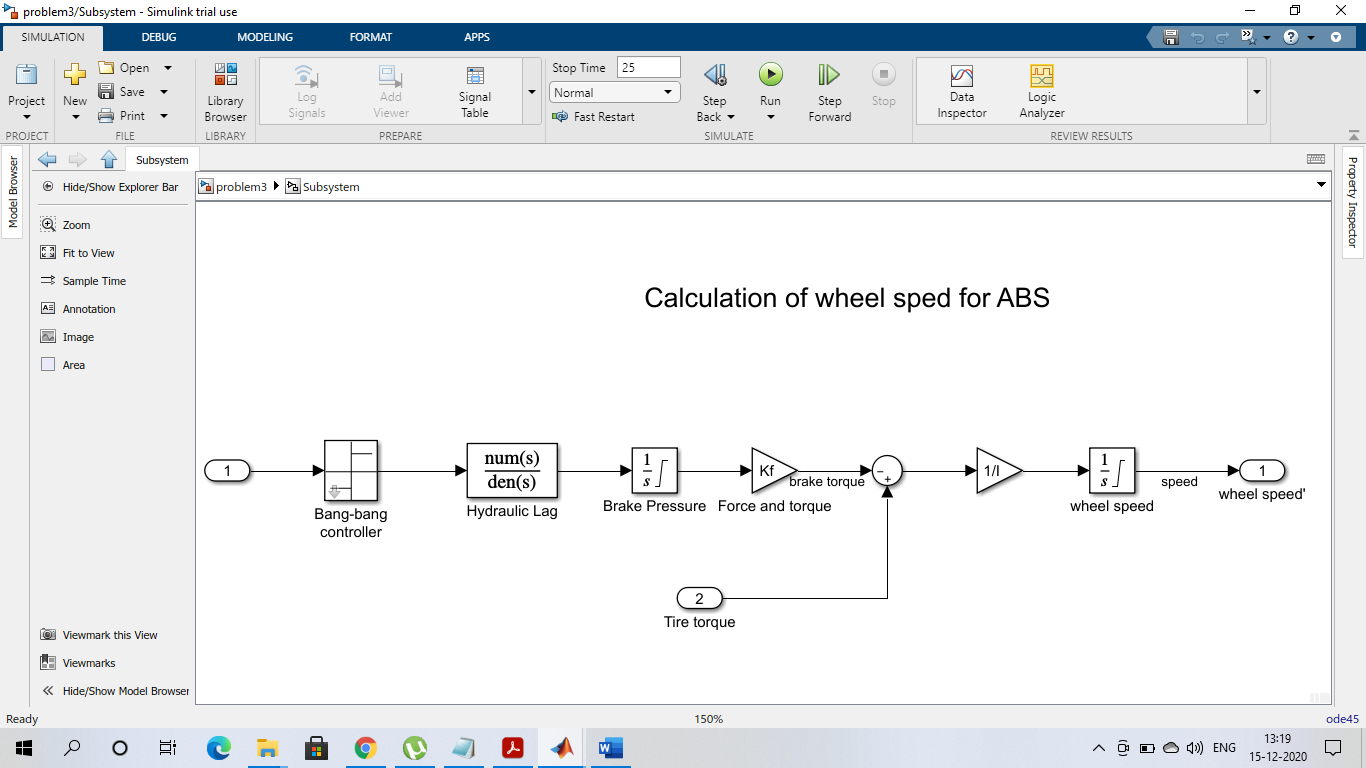


The blocks used in the system model are:

* Gain block
* Integrator
* Signal Generator
* Sum
* 1-D Lookup Table
* Stop Simulation
* MATLAB Function
* Bus
* A Sub-system

**Sub-system model:**

Used for calculation of wheel speed looks as follows:



The blocks used in this sub-system model are:

* Inport
* Sign mask
* Transfer Function
* Integrator
* Gain
* Sum
* Outport

**IMPORTANT FUNCTIONS**

**Callback Function:**

It is a powerful way used to customize the Simulink Model. It is executed when we perform actions on our model. Model callbacks execute at specified action points, for example after you load or save the model. Model call-back functions used are:

Model Pre-load function (PreLoadFcn) callback that created variables in the MATLAB workspace with the name **variables**.

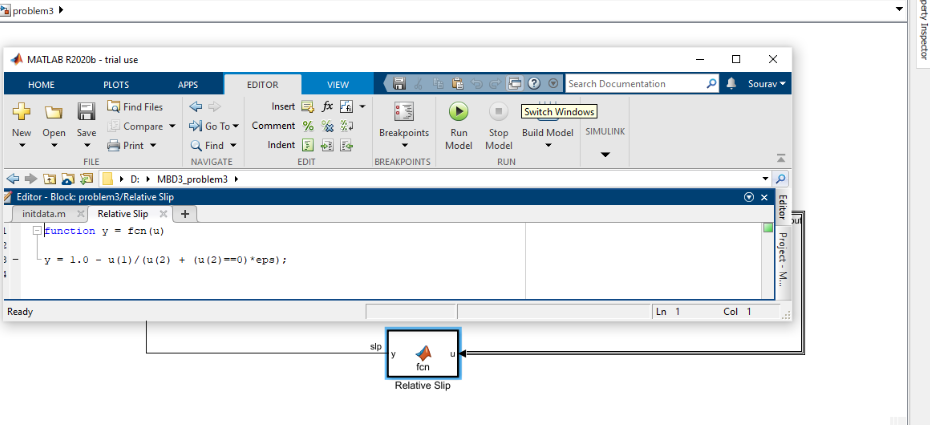
Model close function (CloseFcn) callback that clears the MATLAB workspace. Has not been built or is out of date.

Simulation Stop function (StopFcn) callback checks for the output and then graph is plotted. After that the model function stops.

Instead of using a CloseFcn callback for model , we can use a StopFcn callback in model to clear the variables used by the model from the MATLAB workspace.

**MATLAB FUNCTION:** Use of **MATLAB Function** in the model helps to define custom functionality in Simulink models by using the MATLAB language. In the model developed, MATLAB function block is used to calculate Relative Slip.

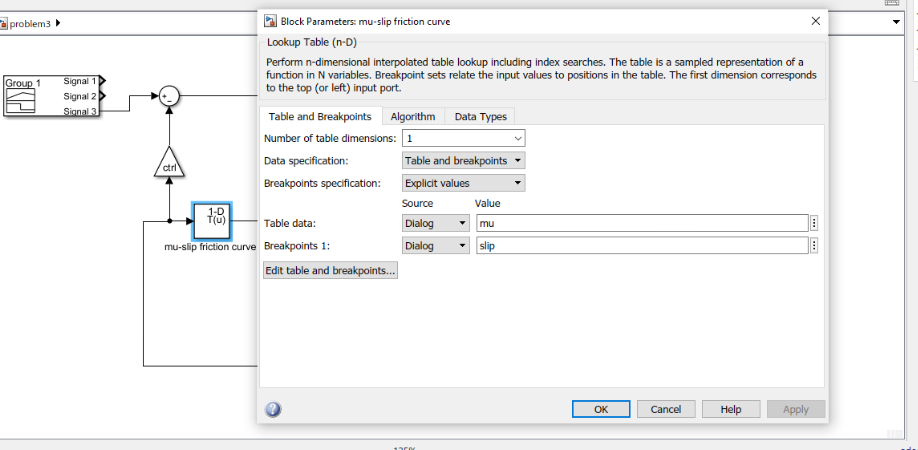
It takes the yout i.e. the output of the bus as the input to block and then the MATLAB code is written into it. And the output od the block helps to control relative slip. That’s the beauty of this block that we can make use of MATLAB editor code for function in the Simulink.



Screenshot of MATLAB Function

**1-D Lookup Table** :

It is used in the model to store the value of friction curve. Slip is taken as an input of the table and the corresponding mu value is stored in the table. These values of mu and slip are stored and provides desired curve.

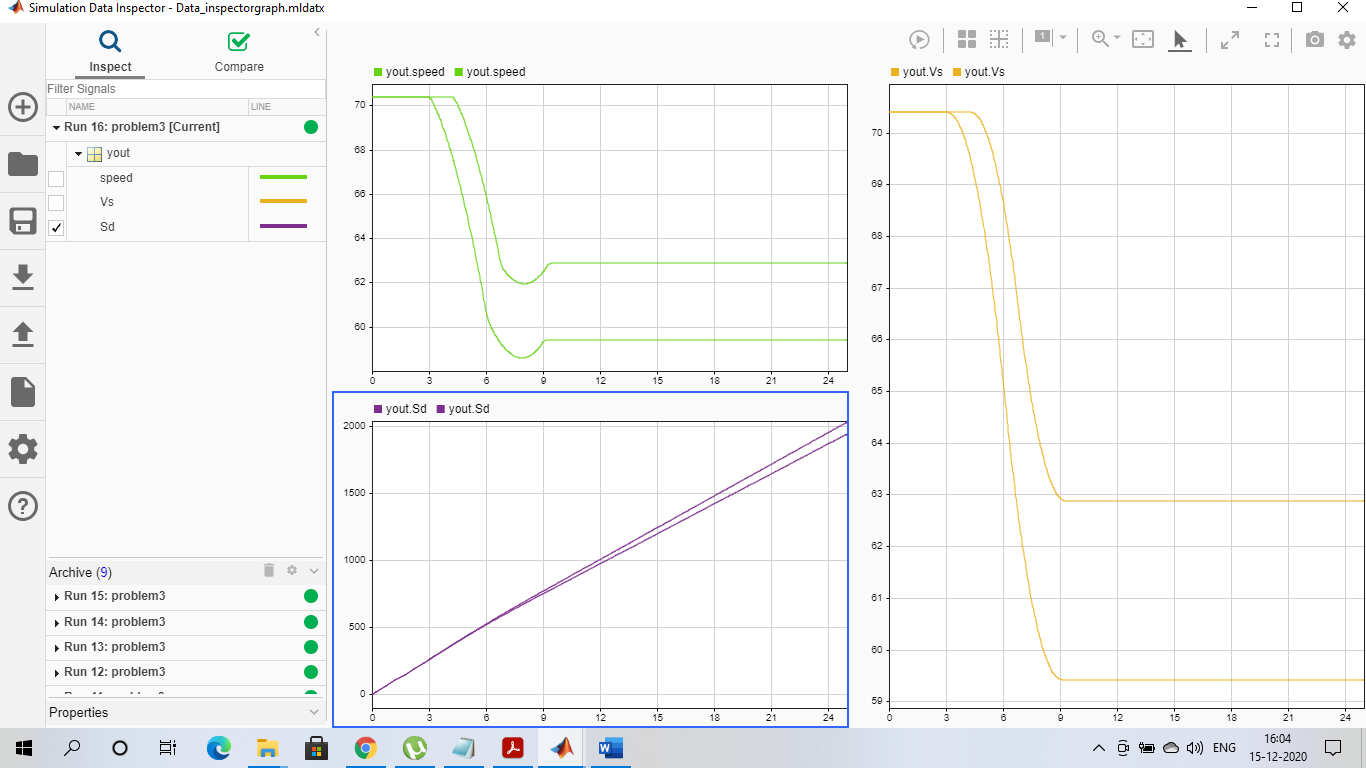


Screenshot of lookup table used in model

**Data Inspector**:

It is used in the model to save the curves of various logged signals for every time any changes are made in the model. All the curves will be saved in the data inspector.

Logged signals as well as outputs and states logged using the Dataset format automatically log to the Simulation Data Inspector when we simulate a model. Graphs are stored as Run1, Run2…. and so on depending on the number of times model is simulated after making changes.



Screenshot of the data inspector

**Solver**:

It is selected from the solver settings and the type of solver selected depends on the requirements. In my model, the output is checked for various type of solvers. An observation I made is that when ode45 solver is selected, the initial disturbances were not there in any of the curves and the simulation time taken was also not bad.

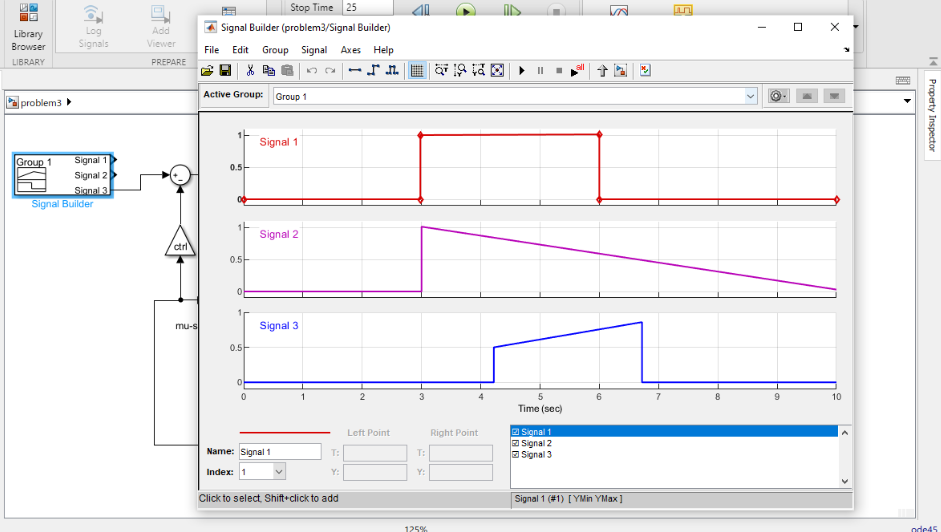
But when solver odeN was selected, simulation time was fast but there was large initial disturbance in the curves. I also checked with ode15, the output was similar to that of ode45 but time taken was less.

So, I went on with **ode45 solver** as there were no disturbances with not so much time taken for simulation.

**Signal Builder:**

This block was taken from library browser. It helped to create interchangeable groups of piecewise linear signal sources and use them in a mode. We can quickly switch the signal groups into and out of a model to facilitate testing. In the Signal Builder window, create signals and define the output waveforms.

Multiple test signals were generated inside the Signal Builder of different amplitude and size. All these signals were used for simulation and output curves were seen for all test cases signals in the data inspector.



Screenshot of test cases signals inside signal builder

So, these are some of the important parts or components I have used in my model which I have tried to mention in the report. Along with the explained methods, there were many other important blocks or functions which I have used and are listed above.

**SOURCES**

1. <https://in.mathworks.com/help/>
2. <https://en.wikipedia.org/wiki/Anti-lock_braking_system#Operation>