MBD Project

WEEK 3

MATLAB & Simulink Implementation of a Complex Anti-Lock Braking System (ABS)

Venkata Diwakar Reddy Kashireddy

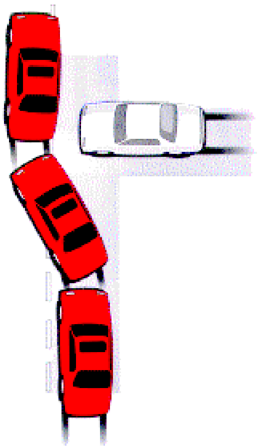
2005674

ABSTRACT

Abs is an abbreviation for antilock braking system it was designed to help the driver maintains some steering ability and avoid skidding while braking

Abs was introduced in the mid-1980s and has become “standard” equipment on the majority of vehicles sold in Canada. ABS in cars and most multi-purpose vehicles (mpv’s) works on all four wheels. This promotes directional stability and allows steering while maximizing braking.

The abs in most pick-up trucks works only on the rear wheels, which promotes directional stability only. However, there are some available with abs on all four wheels.



Control under heavy breaking

ABS allow you to maintain control of the vehicle. Since four-wheel ABS prevents all wheels from skidding, it allows you to steer the vehicle and still maintain braking.

Where is it located?

The ABS is a three-channel system. That means that the abs hydraulic unit has a separate control valve for each front wheel, and a third control valve that is shared by both back wheels. Because of that, when one rear wheel starts to lose traction, braking has to be reduced to both rear wheels.

This is a simple model of an Anti-Lock Braking System (ABS). It simulates the dynamic behaviour 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. This must be visible in the Simulink output graphs.

INTRODUCTION

Abs uses wheel speed sensors to determine if one or more wheels are trying to lock up during braking. if a wheel tries to lock up, a series of hydraulic valves limit or reduce the braking on that wheel. This prevents skidding and allows you to maintain steering control there is a toothed ring that spins at each wheel next to a magnetic hall affect sensor. As the wheel turns, the sensor sends out a pulse to the abs controller as each tooth passes by it. By measuring the frequency of the pulses coming in, the controller can determine how fast the wheel is turning. Note that it can't tell which direction the wheel is turning; the pulses are the same either way. it assumes the wheel is rolling forward. By comparing the speed difference of each wheel, it can detect when one or more wheels are slowing down faster than the car, indicating an Impending loss of traction. The abs controller then commands the abs hydraulic unit to release the pressure on that wheel's brake. It then reapplies brake pressure as soon as it senses that the wheel has sped back up. This happens rapidly over and over (about eight times a second) so that there is a perceived pulsing or buzzing sensation. By adjusting the braking this way, this wheel's tire is held right at its maximum traction limit.

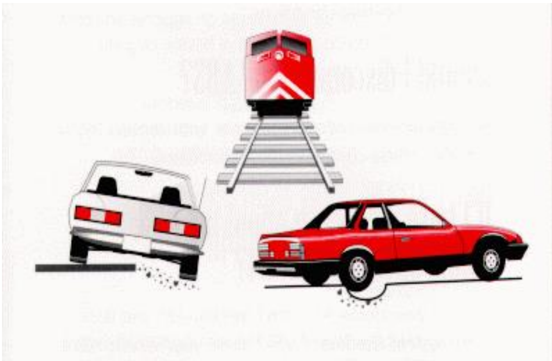
The abs control firmware takes into account not just the difference in speed between each wheel, but also a maximum deceleration rate (incase the system misses and a wheel does actually lockup), as well as compensation for cornering (the outside wheels in a corner need to spin faster than the inside wheels. So, it checks to see if the difference between the inside and outside at the front is similar to the difference between the inside and outside at the rear) another important compensation that abs performs is for "split-mu" surfaces. an example of this is when the two wheels on one side of the car are on the road (lots of traction), and the wheels on the other side are off the road (less traction). If the system just adjusted to each wheel's maximum braking force (which is what some less advanced abs systems do), the tyres on the side of the car on the pavement would apply a force that would make the car tend to spin out. The split-mu detection algorithm will reduce the overall braking force just enough to prevent this from happening.

Abs has three significant advantages: it reduces the need for driver skill during panic situations. It can separately control braking thresholds. Most important: it allows the driver to steer while braking at the limit.

Since the abs will not allow the tyre to stop rotating, you can brake and steer at the same time. Braking and steering ability of the vehicle is limited by the amount of traction the tyre can generate. Situations where, abs would give assistance, a tyre for all practical purposes has a fixed amount of traction in any directions (accelerating, braking and cornering).when steering and braking at the same time, this traction has to be shared between the two functions. When braking in a straight line, traction of all the the tyres can be used for braking. When cornering at the limit, the tyre has no available traction for braking. Between these two extremes, the traction can be shared. ABS automatically adjusts the brakes for the traction that’s left over after the cornering force. If you demand steering while braking, the 100% of traction that the tire can generate will be divided between both tasks. For example, if you require 50% for steering then there is 50% of available traction left for braking. If you require 10% for steering then there is 90% left for braking. Be aware that 100% traction on a dry road is a great deal more traction than 100% traction on ice! Therefore, your vehicle is unable to steer and brake as well on a slippery surface as it can on a dry road.

Road surfaces and ABS

Road hazards that will cause ABS to function unexpectedly are gravel, sand, ice, snow, mud, railway tracks, potholes, manhole covers and even road markings when it is raining.



The ABS cannot make up for road conditions or bad judgment. It is still the driver’s responsibility to drive at reasonable speeds for weather and traffic conditions. Always leave a margin of safety.

2.5 Instructions while driving with abs

We need to take some necessary tips for operation of vehicle possessing abs

Do keep your foot on the brake.

Do allow enough distance to stop.

Do practice driving with abs.

Do consult the vehicle’s owner’s manual.

Do know the difference between four wheel and rear wheel abs.

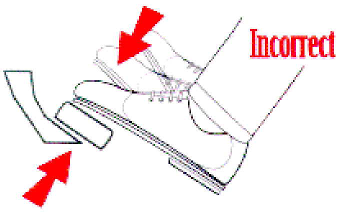
Don’t drive an abs equipped vehicle more aggressively.

Don’t pump the brakes.

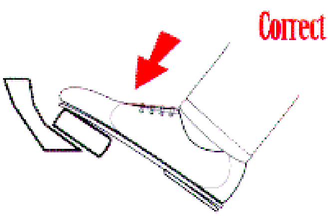
Don’t forget to steer.

HOW DO I USE ABS

Apply steady and constant pressure- don’t take your foot off the brake pedal until the vehicle is stopped and don’t pump the brake.



6 Disadvantages of abs



Abs has a couple of disadvantages: in deep snow or gravel it’s actually better for the wheels to lock up. On completely glare ice, locked wheels will often stop a car faster because even though the sliding friction is less than non-sliding friction, it is applied 100%.

Modeling

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.

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

Equation 1

$$\omega_v = \frac{V_v}{R_r}$$

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

$\omega_v =$ vehicle speed divided by wheel radius

$V_v =$ vehicle linear velocity

$R_r =$ wheel radius

$\omega_w =$ 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.

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 (see note below.).

- Note: In an actual vehicle, the slip cannot be measured directly, so this control algorithm is not practical. It is used in this example to illustrate the conceptual construction of such a simulation model. The real engineering value of a simulation like this is to show the potential of the control concept prior to addressing the specific issues of implementation.

The initial conditions are:

g = 32.18; gravitational pull

v0 = 88;

Rr = 15/12; wheel radius

Kf = 1; Force & torque

m = 50; weight

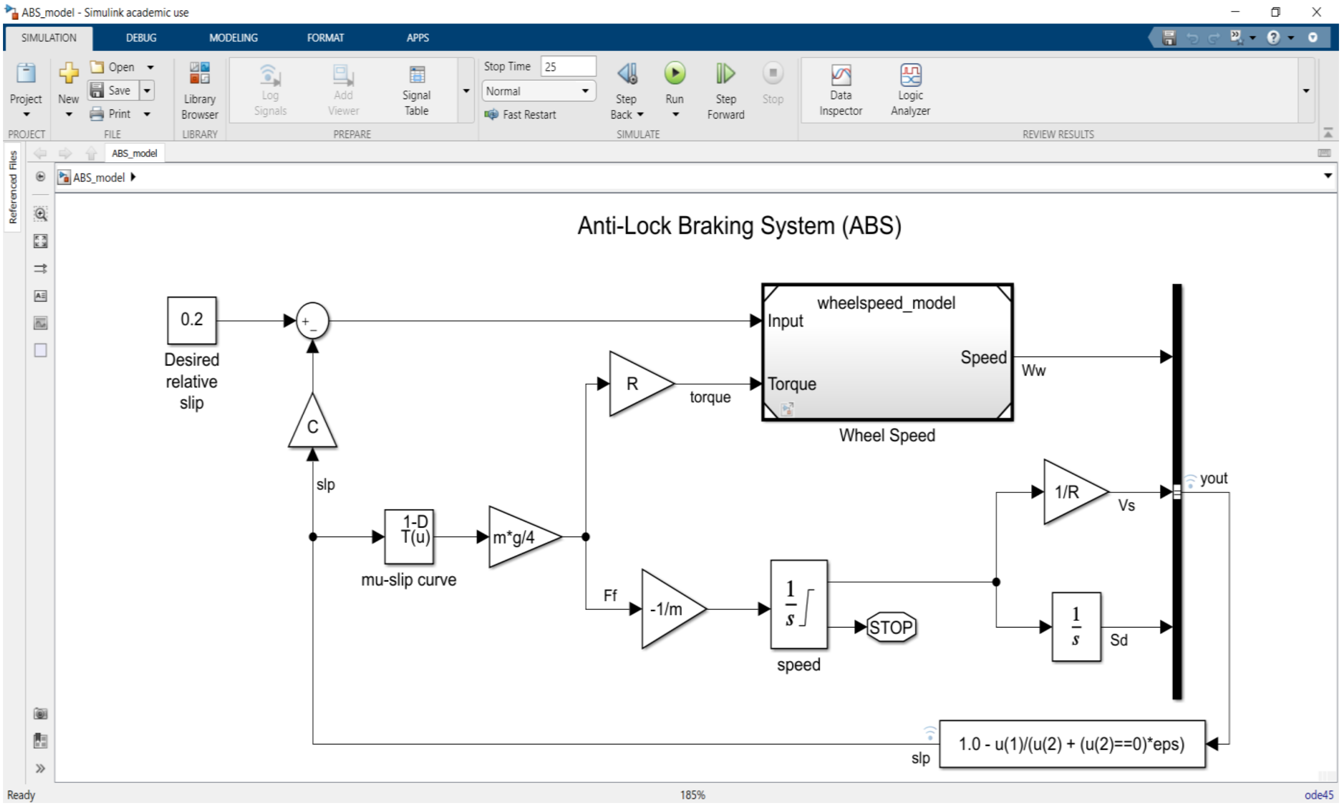
PBmax = 1500; %Upper limit of integrator for Brake Pressure

TB = 0.01;

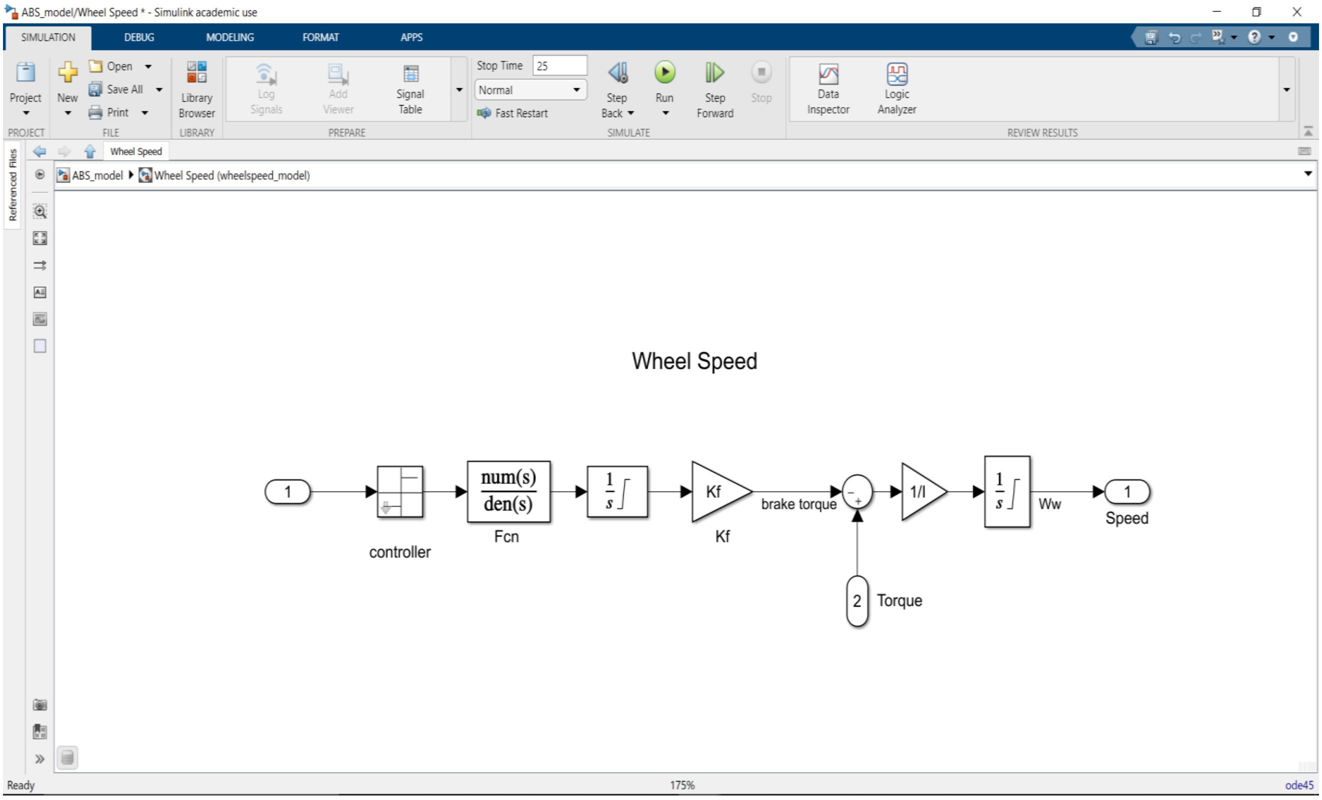
I = 5;

Simulink Model:

1. ABS Model



2. Wheel Speed Model



Variables in Workspace:

Workspace

Name	Value
absbrake_out...	*1x1 Dataset*
C	1
ctrl	1
g	32.1800
I	5
Kf	1
m	50
mu	*1x21 double*
PBmax	1500
R	1.2500
Rr	1.2500
slip	*1x21 double*
TB	0.0100
v0	88

Callbacks:

Callbacks(3)

PreLoadFcn*

```
% Load initialization data


g = 32.18;
v0 = 88;
Rr = 15/12;  % Wheel radius
Kf = 1;
m = 50;
PBmax = 1500;
TB = 0.01;
I = 5;
```

Callbacks(3)

StopFcn*

```
if exist ('absbrake_output')
  absbrakeplots;
end
```

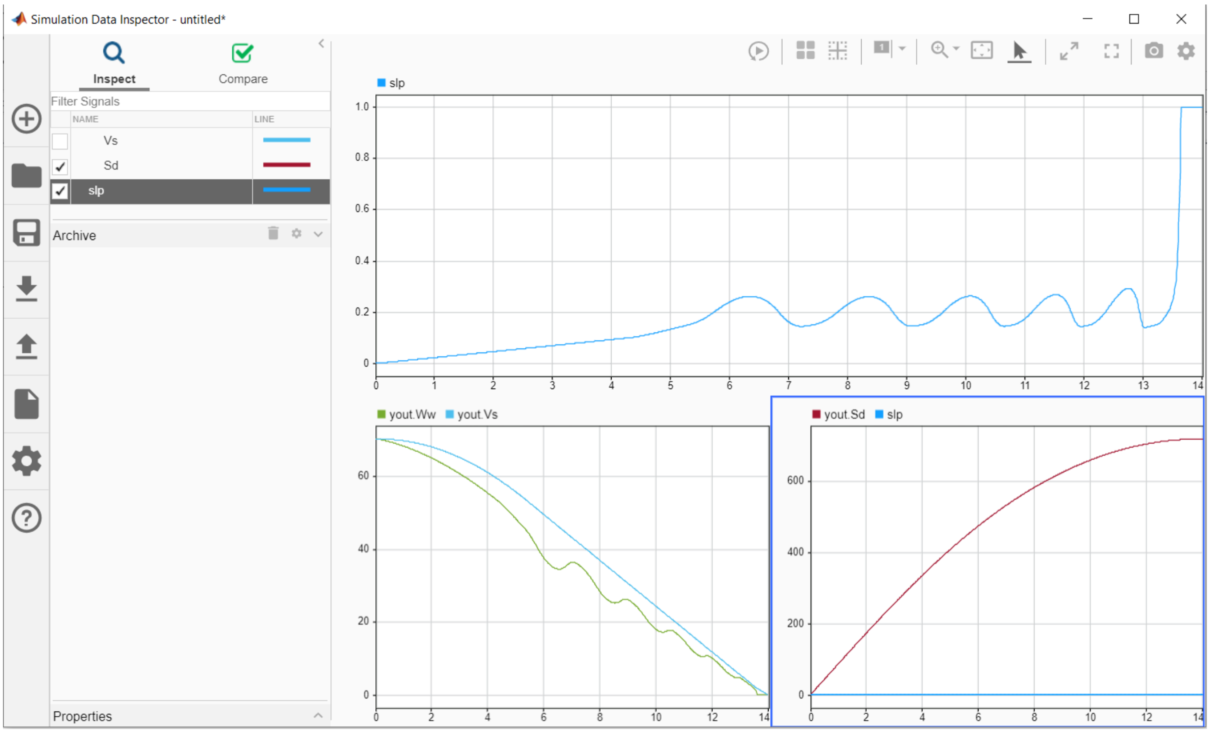
Data Inspector in the SIMULINK model

You will use the Simulation Data Inspector to simulate the data you generate during the design process. Simulation data that you log on to the Simulation Data Inspector in the Simulink Model Logs. You may also import test data and other reported data to the Simulation Data Inspector to review and analyse it alongside logged simulation data. The Simulation Data Inspector provides many types of plots that help you to quickly construct complex visualizations of your data.

View Logged Data

Logged signals as well as outputs and states logged using the Dataset format automatically log to the Simulation Data Inspector when you simulate a model. You can also record other kinds of simulation data so the data appears in the Simulation Data Inspector at the end of the simulation. To see states and output data logged using a format other than Dataset in the Simulation Data Inspector, in the Model Configuration

Parameters Data Import/Export pane, select the Record logged workspace data in Simulation Data Inspector option.



Data Inspector output in SIMULINK

Solver Selection Strategy

ODE23 and ODE45 are MATLAB's ordinary differential equation solver functions.

ODE23 is based on the integration method, Runge Kutta23, and ODE45 is based on the integration method, Runge Kutta45. The way that ODE23 and ODE45 utilize these methods is by selecting a point, taking the derivative of the function at that point, checking to see if the value is greater than or less than the tolerance, and altering the step size accordingly.

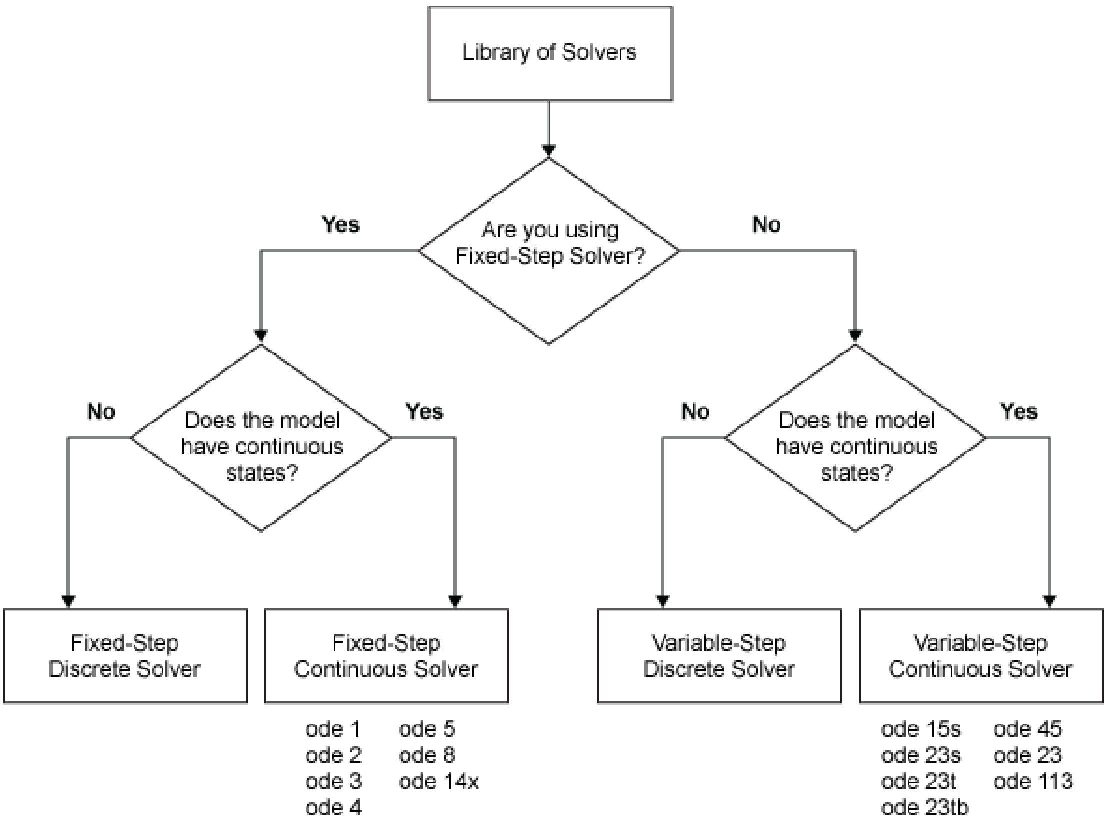
This integration approaches do not lend themselves to set phase sizes. Using an algorithm that uses a fixed phase size is risky, since you may skip points where the frequency of the signal is greater than the frequency of the solver. Using a variable step means that a broad step size is used for low frequencies and that a small step size is used for high frequencies. ODE23/ODE45 was designed for variable phase, run faster with variable step duration, and the results are obviously more reliable.

If one wishes to obtain only those values at a certain fixed increment, do the following:

- Use ODE45 to solve the differential equation.
- Better to use with variable step solver

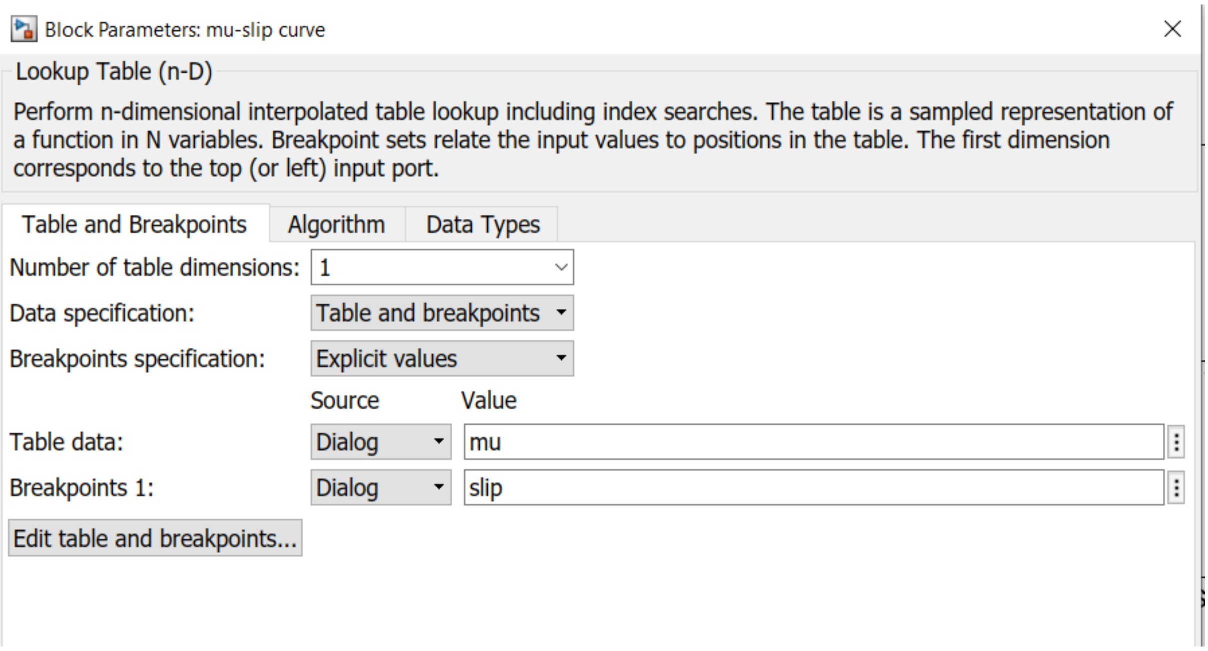
For this simulation, ODE45 is chosen.

This chart provides a broad classification of solvers in the Simulink library.



Look – up Table:

A look up table is created mu and slip.

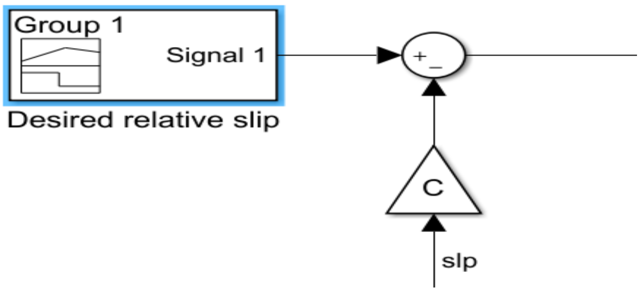


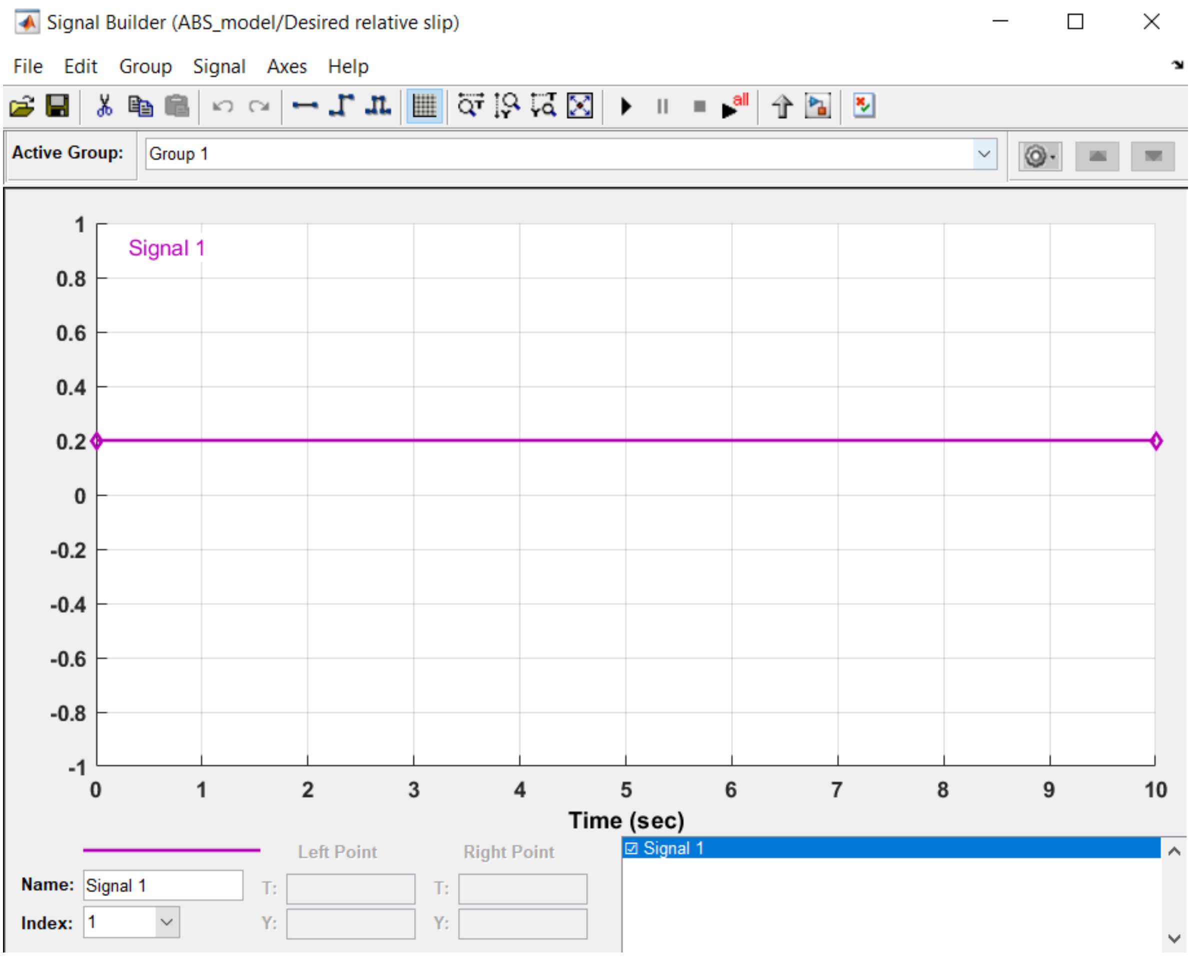
Signal Builder

Signal builders are used for creating tests in Simulink. Let's say you've created a low-pass filter and want to make sure that you didn't make any mistakes while developing it. You may want to input a signal to your low-pass filter such as different kind of sine waves with different frequencies and look at your filter's behaviour.

A Signal Builder also allows you to construct different kinds of test cases so that you are running all of them to study your model's behaviours.

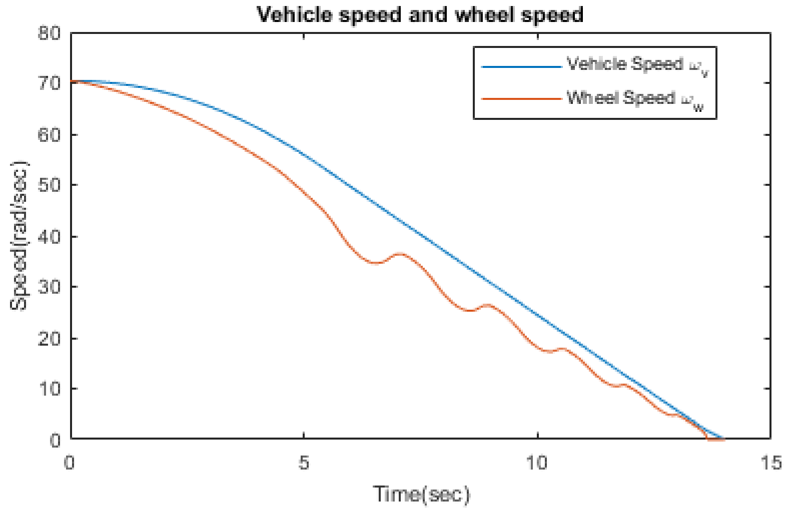
You can find Simulink blocks in the Simulink library at "Simulink/Sources." Here's what the Signal Builder block looks like:

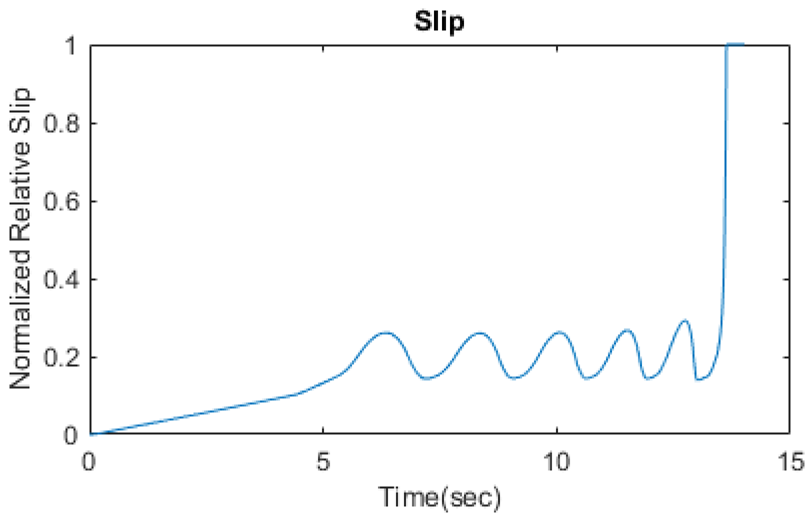



Signal Builder (ABS_model/Desired relative slip)
File Edit Group Signal Axes Help
Active Group:
Group 1
Signal 1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0 1 2 3 4 5 6 7 8 9 10
Time (sec)
Left Point
Right Point
Name: Signal 1
Index: 1
T:
Y:
T:
Y:
Signal 1

Outputs:

Running the Simulation in ABS Mode



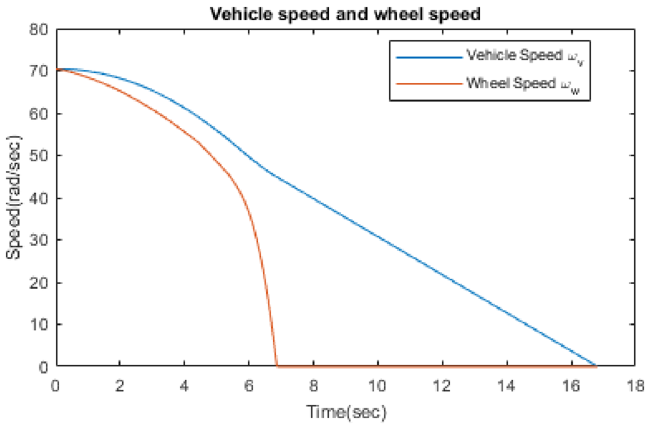


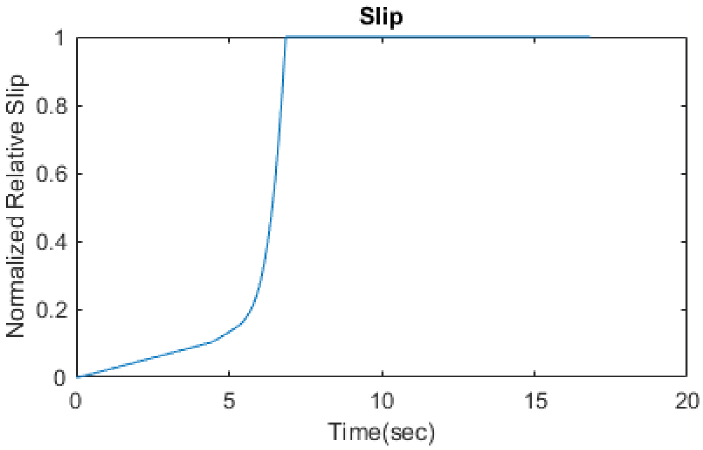
The plots above show the ABS simulation results. The first plot shows the wheel angular velocity and corresponding vehicle angular velocity. This plot shows that the wheel speed stays below vehicle speed without locking up, with vehicle speed going to zero in less than 15 seconds.

Running the Simulation Without ABS

For more meaningful results, consider the vehicle behavior without ABS. At the MATLAB command line, set the model variable ctrl = 0. This disconnects the slip feedback from the controller, resulting in maximum braking.

```
ctrl = 0;
```

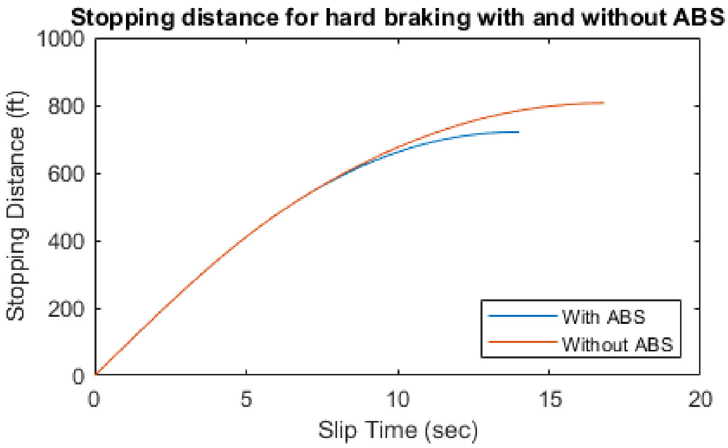




Braking with ABS Versus Braking Without ABS

In the plot showing vehicle speed and wheel speed, observe that the wheel locks up in about seven seconds. The braking, from that point on, is applied in a less-than-optimal part of the slip curve. That is, when slip = 1, as the slip plot shows, the tire is skidding so much on the pavement that the friction force has dropped off.

This is, perhaps, more meaningful in terms of the comparison shown below. The distance traveled by the vehicle is plotted for the two cases. Without ABS, the vehicle skids about an extra 100 feet, taking about three seconds longer to come to a stop.



Conclusion

With such sophisticated technology which has made driving simpler than ever before, there is an immense demand for these systems becoming more and more common in the vehicles to come in future, though some auto enthusiasts don’t prefer to drive with the ABS on as it takes away some of the driving skill from the driver, but this is true only in the case of serious auto enthusiasts. But overall, the system works towards the safety of the driver, and it has got praises from worldwide market. So, the technology just goes on growing and working toward more and more safety.

This model shows how you can use Simulink to simulate a braking system under the action of an ABS controller. The controller in this example 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® Coder™ 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.