Anti-Lock Braking System (ABS)

17-12-2020
Model Based Design

Omkar Rane 2005203
KPIT-Genesis

Contents

1. Introduction

Problem statement:

ABS System Design (Anti-Lock Braking System Design)

Qn: A 4-wheeled vehicle is to decelerate from a speed of 44 m/s (V0) to standstill without wheel locking. The co-efficient friction between the tyres and road is known. The desired wheel slip is 0.2. Find the vehicle speed and slip over time, and the stopping distance.

Mathematical modeling and simulation bring significant benefits in terms of understanding the braking phenomena and the impact of different parameters on the braking performance of a vehicle. Anti-lock braking systems (ABS) are meant to control the wheel slip in order to maintain the friction coefficient close to the optimal value. Wheel slip is defined as the relative motion between a wheel (tire) and the surface of the road, during vehicle movement.

Wheel slip occurs when the angular speed of the wheel (tire) is greater or less compared to its free-rolling speed. In order to simulate the braking dynamics of a vehicle, we are going to implement simplified mathematical models (quarter-car model) for both vehicle and wheel. Also, a simplified ABS controller is going to be implemented in order to emulate the braking torque in slip conditions.

1.1 Vehicle model
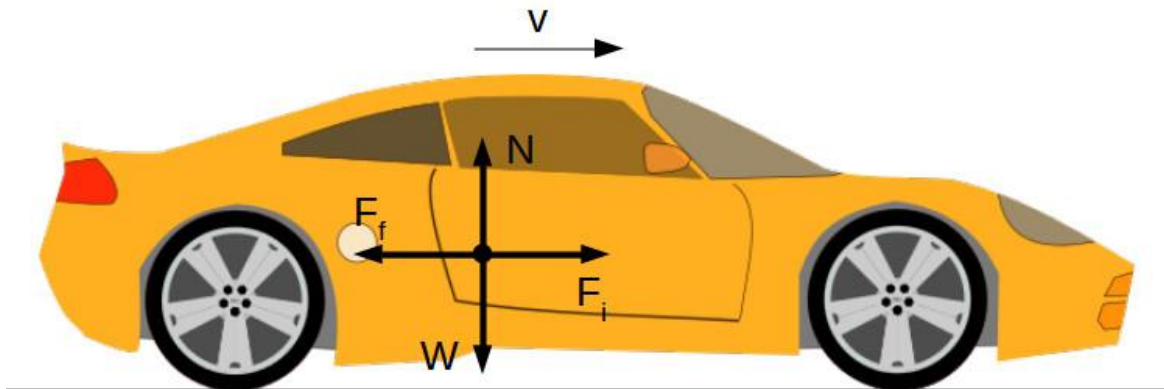
If we consider a vehicle moving in a straight direction under braking conditions, we can write the equations of equilibrium:

for horizontal direction:

$F_f = F_i$ -(1)

where:

F_f [N] – is the friction force between wheel and ground

F_i [N] – is the inertial force of the vehicle

for vertical direction:

$N = W$ -(2)

where:

N [N] – normal force (road reaction)
W [N] – vehicle weight

We can write the expressions of the friction force as:
$F_f = \mu \cdot N$ -(3)
where μ [-] the friction coefficient between wheel and road.

The vehicle's weight is:

$W = m_v \cdot g$ -(4)

Replacing (2) and (4) in (3) gives the expression of the friction force as:

$F_f = \mu \cdot m_v \cdot g$ -(5)

where:
m_v [kg] – is the total vehicle mass
g [m/s²] – is the gravitational acceleration
The inertia force is the product between the vehicle mass m_v [kg] and vehicle acceleration
a_v [m/s²]:

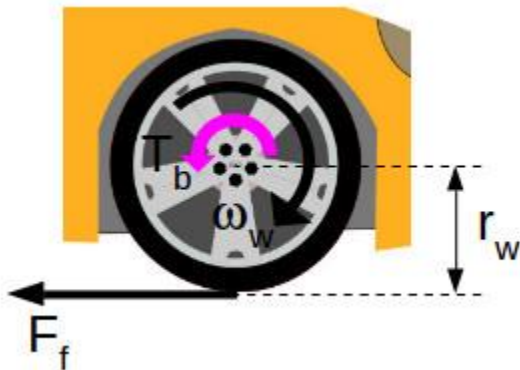$F_i = m_v \cdot a_v = m_v \cdot \dfrac{dV_v}{dt}$ -(6)

where v_v [m/s] is the vehicle speed.
From equations (1), (5) and (6) we can extract the expression of the vehicle acceleration:

$\dfrac{dV_v}{dt} = \dfrac{1}{m_v \cdot} (\mu \cdot m_v \cdot g)$ -(7)

Vehicle speed is obtained by integration of equation (7).

1.2 Wheel model

During braking, through the braking system, the driver applies a braking torque T_b [Nm] to the wheels. The friction force F_f [N] between the wheel and road creates an opposite torque with the wheel radius r_w [m].
For simplification, we are going to consider that the wheel is rigid and the normal force (road reaction) passes through the wheel hub, therefore doesn't add an additional torque.

We can write the equation of equilibrium for the wheel as:

$$T_b - F_f \cdot r_w - J_w \cdot \frac{d\omega_w}{dt} = 0 \quad -(8)$$

where:

J_w [kg·m2] – is the wheel's moment of inertia
ω_w [rad/s] – is the angular speed of the wheel
From equation (8) we can extract the expression of the **wheel acceleration**:

$$\frac{d\omega_w}{dt} = \frac{1}{J_w} \cdot (T_b - F_f \cdot r_w) \quad -(9)$$

Wheel speed is obtained by integration of equation (9).
Wheel slip

The ABS system has to control the wheel slip s [-] around an optimal target. The **wheel slip** is calculated as:

$$s = 1 - \frac{\omega_w}{\omega_v} \quad -(10)$$

where ω_v [rad/s] is the **equivalent angular speed of the vehicle**, equal with:

$$\omega_v = \frac{v_v}{r_w} \quad -(11)$$

where v_v [m/s] is the vehicle speed.
Friction coefficient

The friction coefficient between wheel and road depends on several factors, like:

- wheel slip
- vehicle speed
- the type of the road surface
- environmental conditions (humidity, temperature, etc.)

For our simulation purpose, we are going to take into account only the variation of the **friction coefficient** function on the **longitudinal wheel slip**.
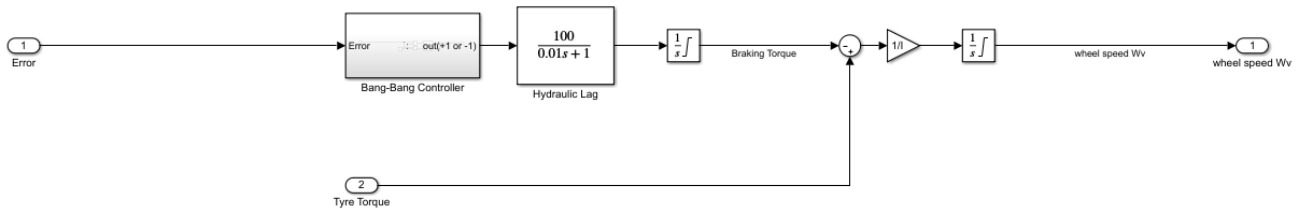
2. Model Based Design in Simulink

ABS System Design (Anti-Lock Braking System Design) (Design a complex automotive system)
Qn: A 4-wheeled vehicle is to decelerate from a speed of 44 m/s (V0) to standstill without wheel locking. The co-efficient friction between the tyres and road is known. The desired wheel slip is 0.2. Find the vehicle speed and slip over time , and the stopping distance.
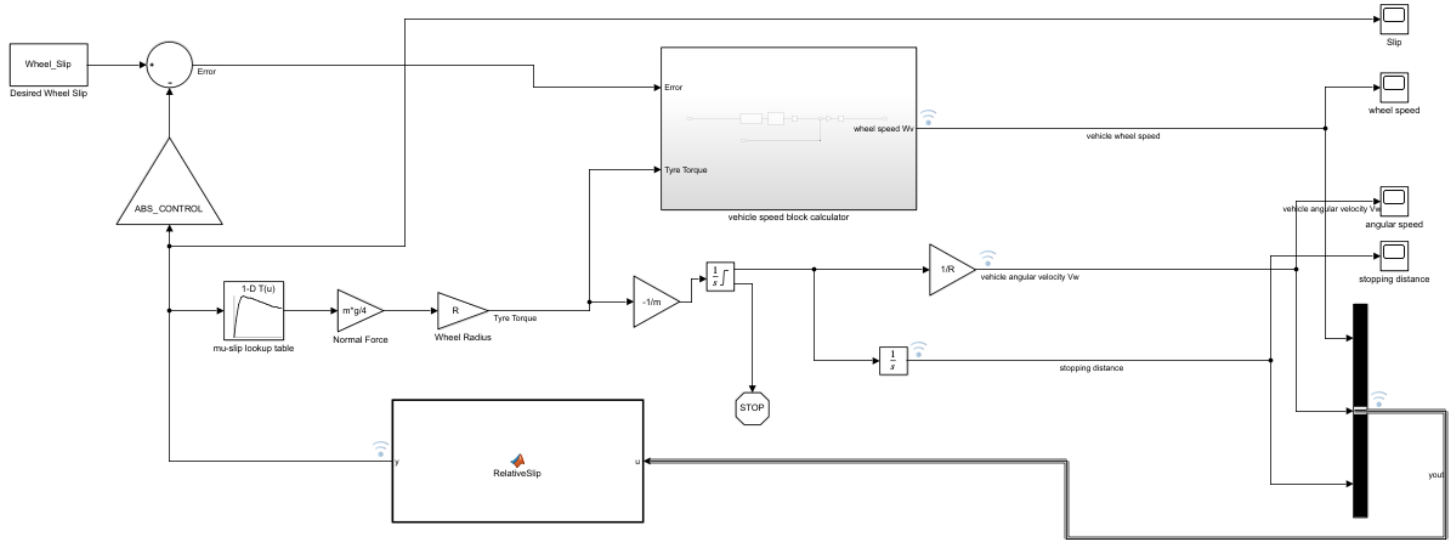
Vehicle speed block controller

Bang-Bang Controller design

Week3_problem3 ▸ vehicle speed block calculator ▸ Bang-Bang Controller

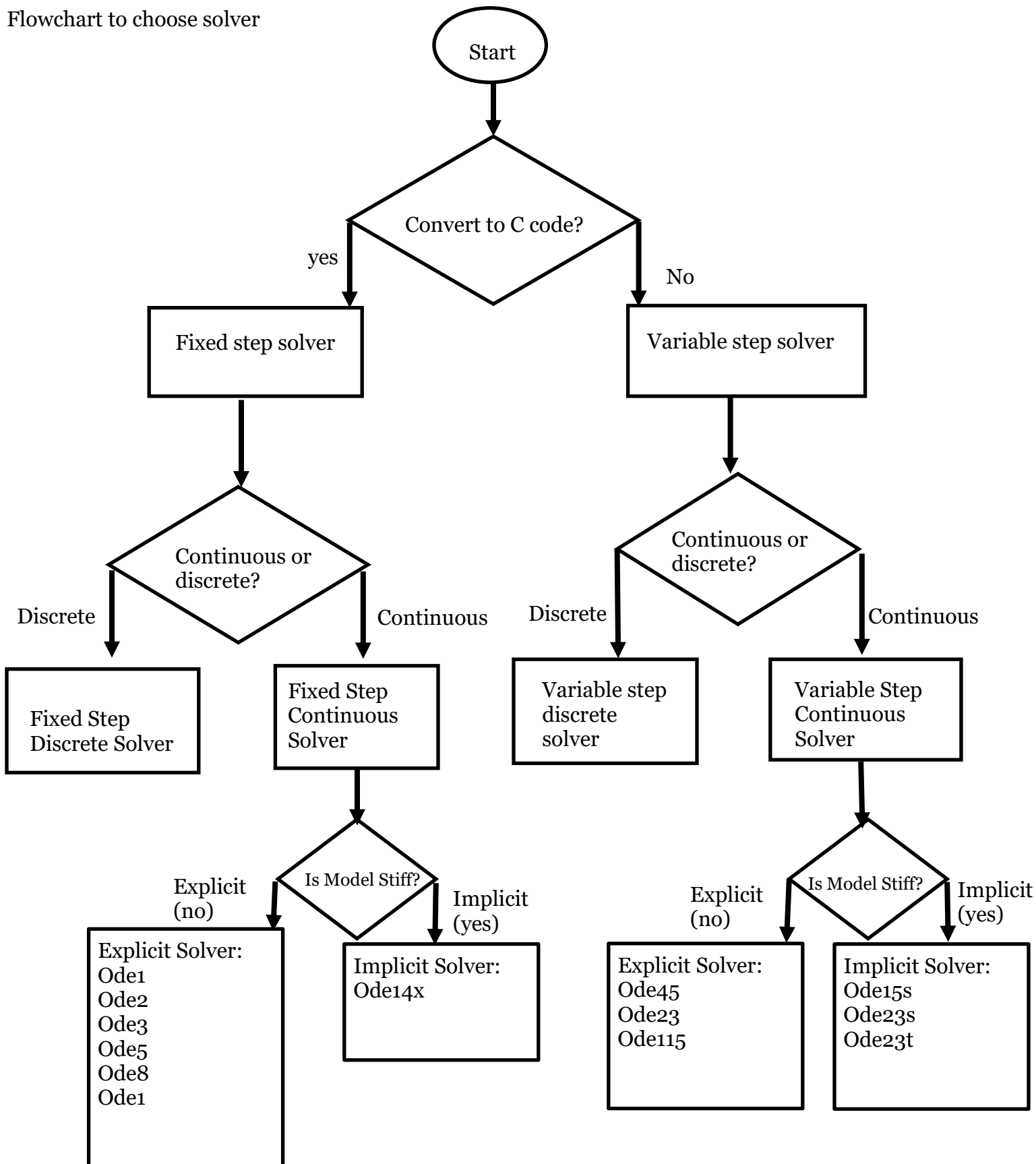# Testing ABS system with signal builder block

3. Function Call backs

To initialize value in workspace before simulating actual model. We are intializing values like vehicle mass,speed,radius of wheel ,moment of inertia,wheel slip.
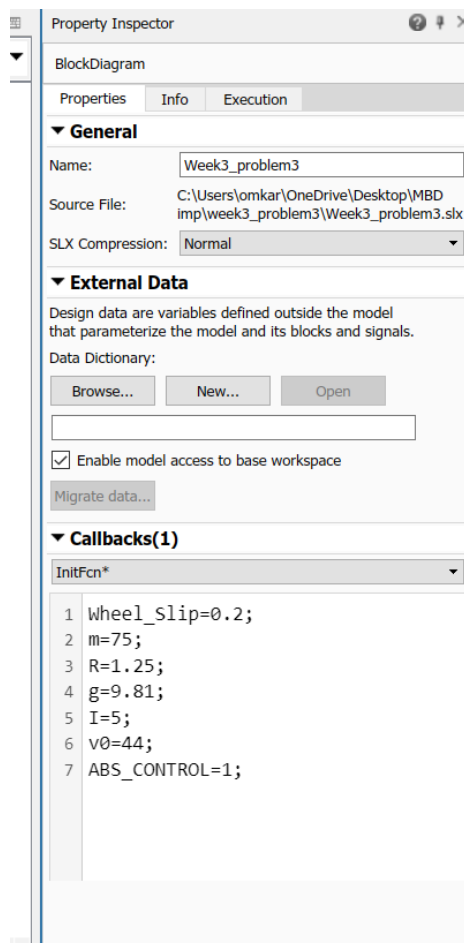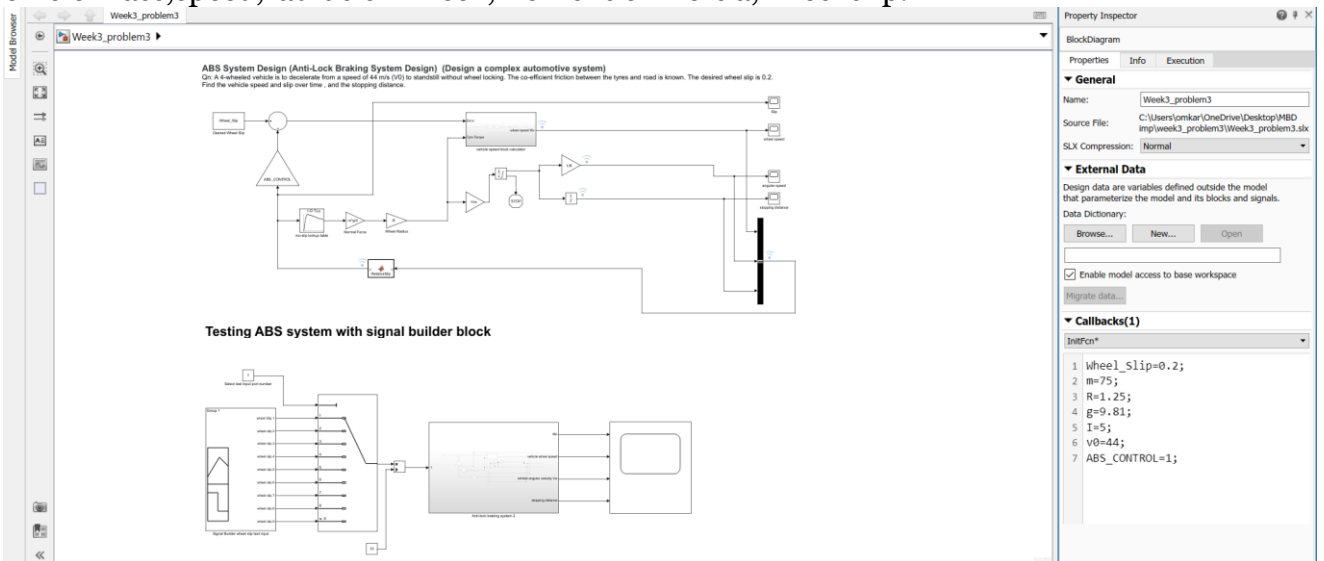
4. Solver Strategy

Flowchart to choose solver

```
                              ┌─────────┐
                              │  Start  │
                              └─────────┘
                                   │
                                   ▼
                          ◇ Convert to C code? ◇
                   yes ↙                        ↘ No
        ┌──────────────────┐            ┌──────────────────┐
        │ Fixed step solver│            │Variable step solver│
        └──────────────────┘            └──────────────────┘
                 │                                │
                 ▼                                ▼
        ◇ Continuous or discrete? ◇      ◇ Continuous or discrete? ◇
   Discrete ↙          ↘ Continuous   Discrete ↙          ↘ Continuous
```

Fixed Step Discrete Solver	**Fixed Step Continuous Solver**

◇ Is Model Stiff? ◇

Explicit (no) / Implicit (yes)

Explicit Solver:
Ode1
Ode2
Ode3
Ode5
Ode8
Ode1

Implicit Solver:
Ode14x

Variable step discrete solver	**Variable Step Continuous Solver**

◇ Is Model Stiff? ◇

Explicit (no) / Implicit (yes)

Explicit Solver:
Ode45
Ode23
Ode115

Implicit Solver:
Ode15s
Ode23s
Ode23t

Solver strategy of ABS system is as follows:

1) Based on above mentioned flowchart we are not required to generate any C code at present so I am proceeding with variable step solver.
2) MATLAB solver are numerical solver based on numerical methods.
3) Initially to determine best solver method we can select type as 'Variable step' and Solver as 'Auto' MATLAB will select best suitable solver as per requirements.
4) Stiffness of model can be determined by amount of time required for calculation. Based on message given by MATLAB whether model is stiff we select solver from above list.
5) In our case model is not stiff so we are selecting ODE45 based on dormand-prince solver selected by auto-solver.

The system has / The solver type is	Only discrete states	Continuous states	
		The system is an ordinary differential equation (ODE)	The system has differential algebraic equations (DAE)
Fixed-Step	Fixed StepDiscrete	ode3	ode14x
Variable-Step	Variable StepDiscrete	The model is stiff: ode15s / The model is non-stiff: ode45	ode23t

5. Data Inspector

Using data inspector we can compare graphs with various conditions and we can archive our results.
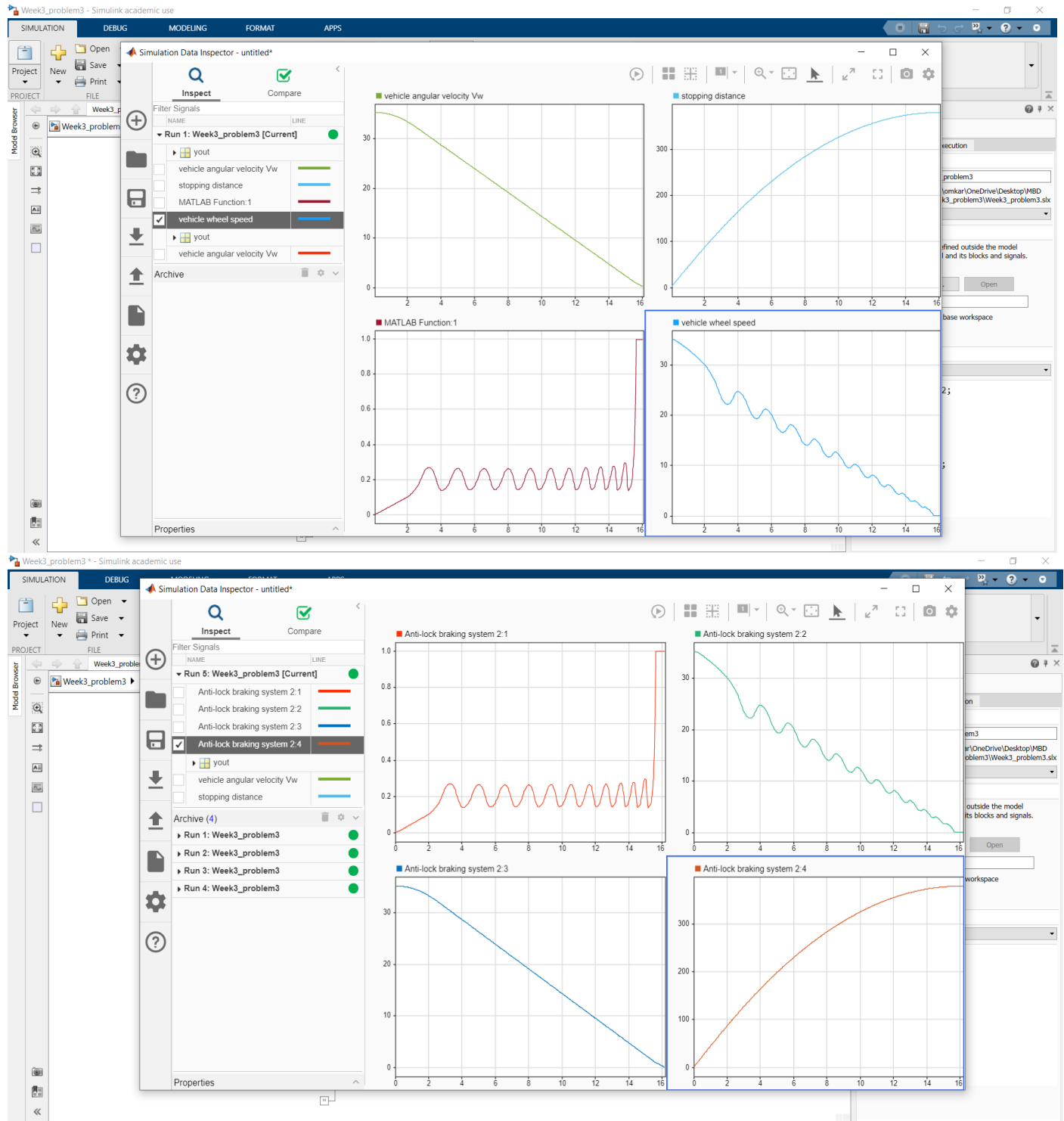
6. MATLAB Function Block

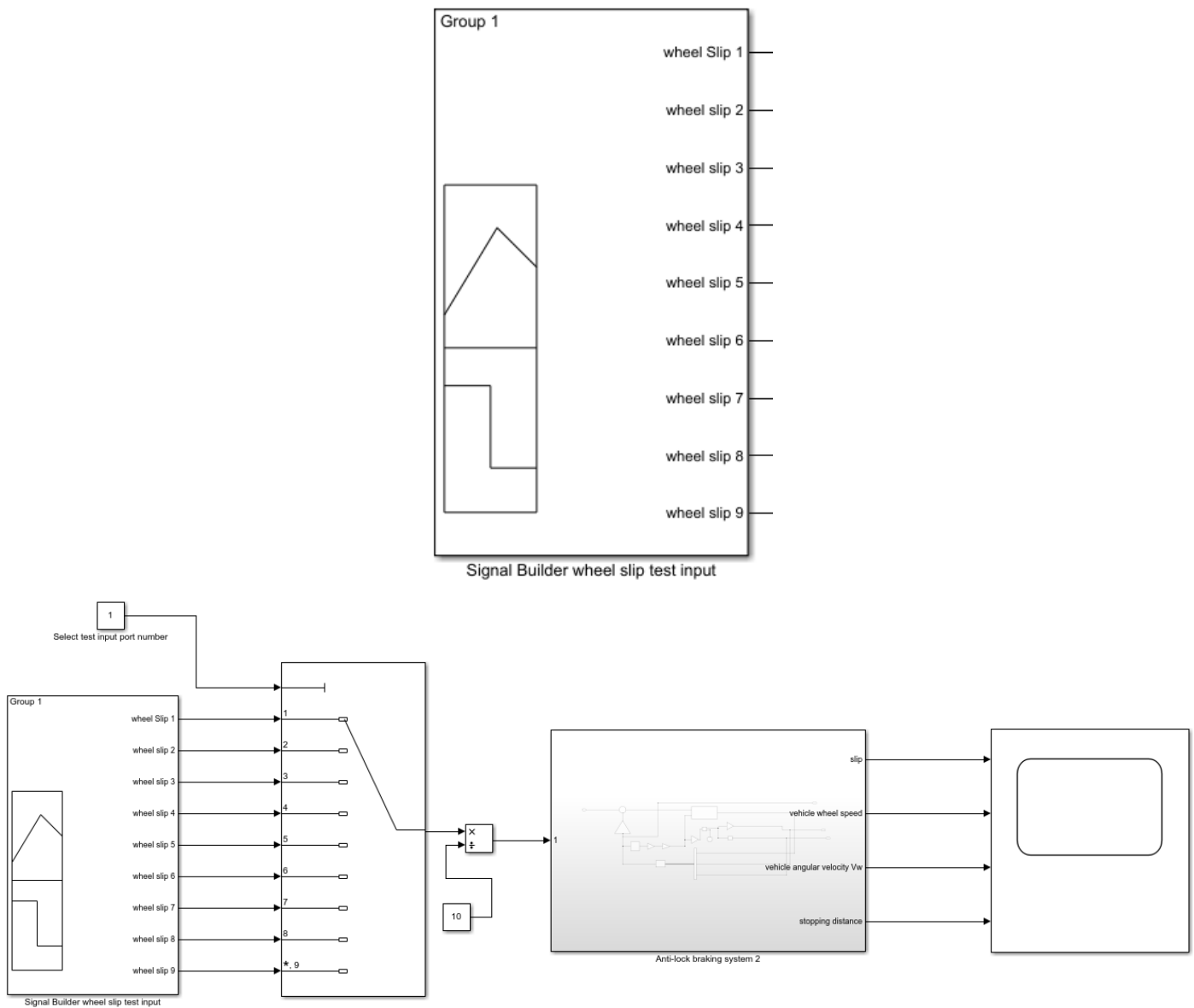In our model we are using matlab function block to calculate relative slip and provide a feedback to the system.
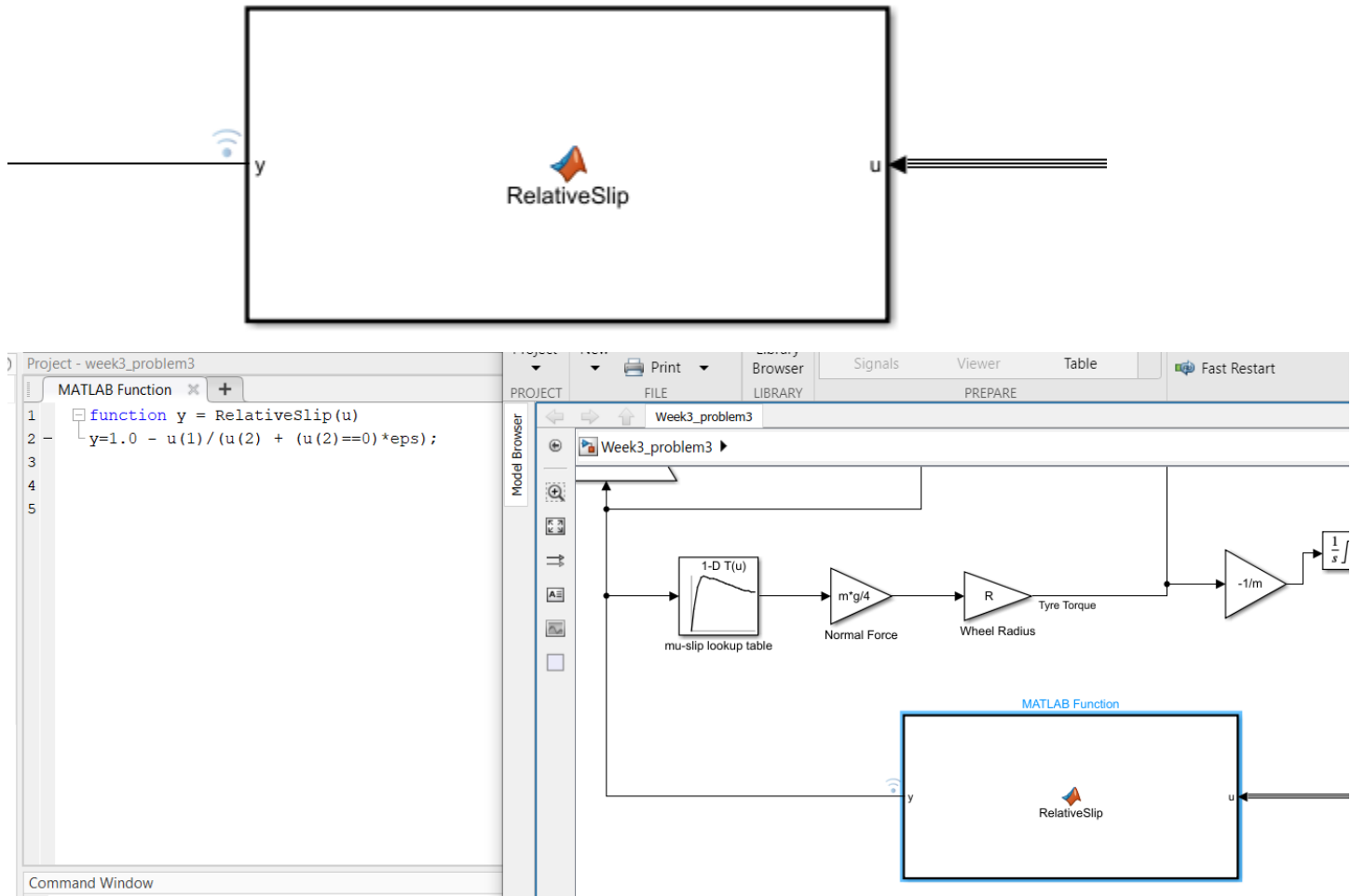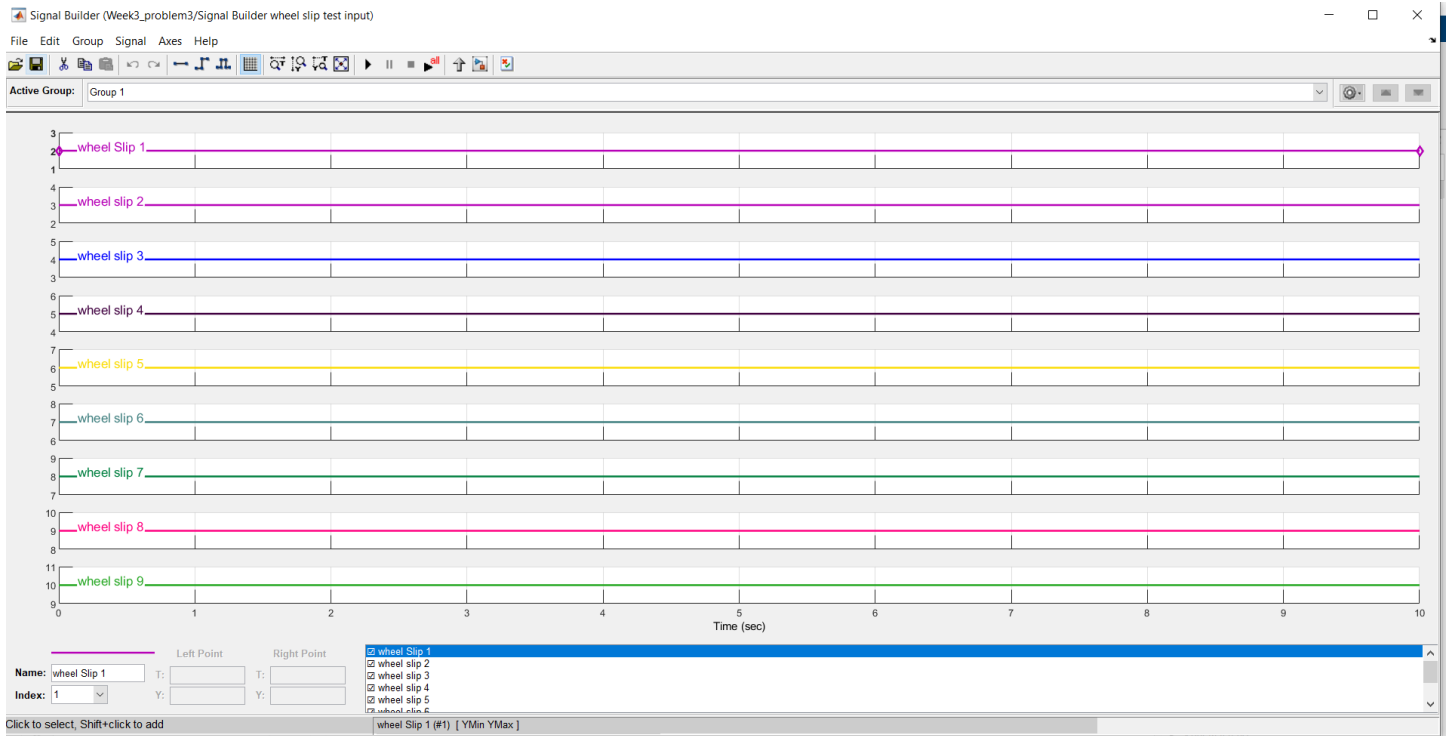
7. Look-up Table

1D lookup Table is used to get appropriate value of coefficient friction mapped to desired slip values. It's a graph of coefficient of friction vs Slip.
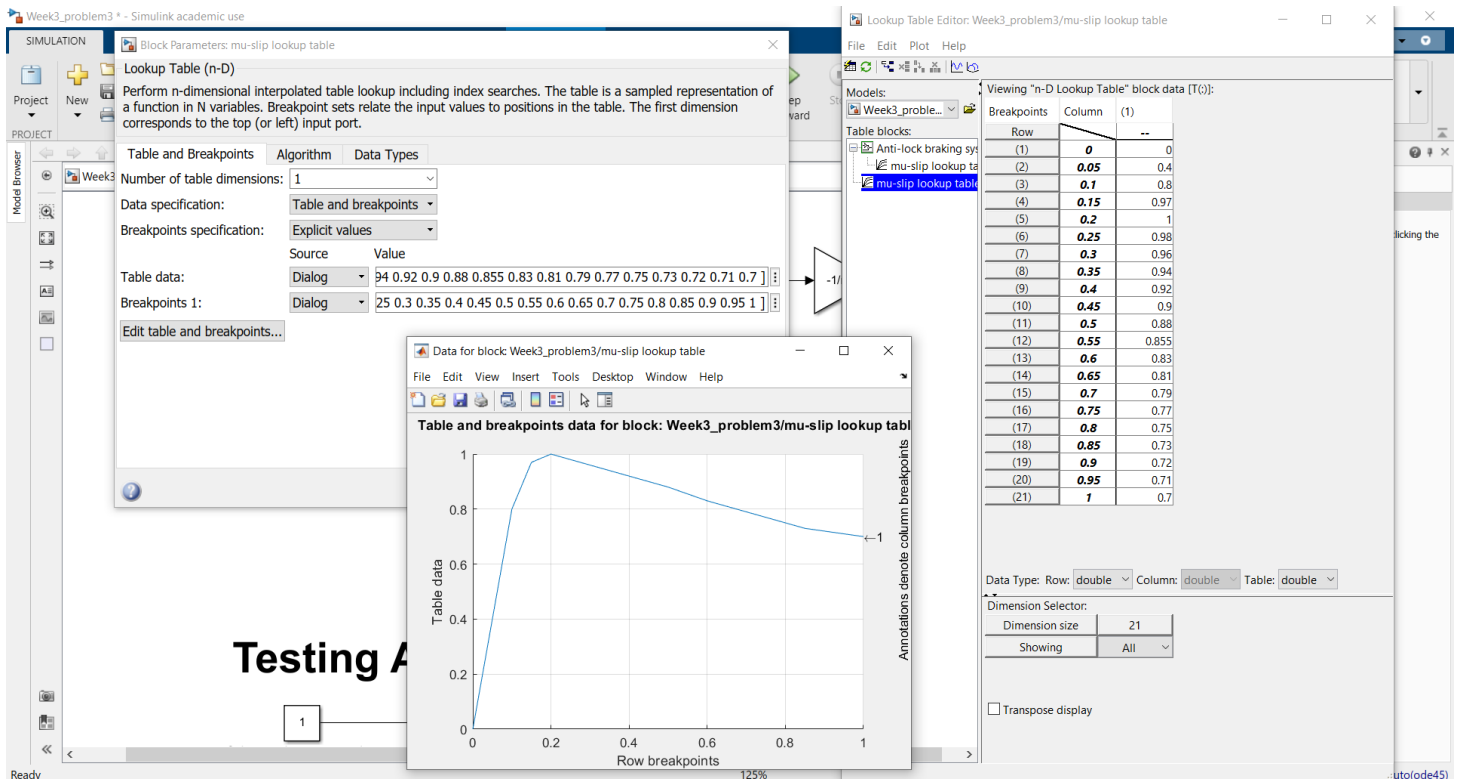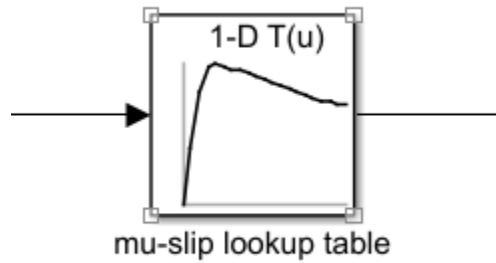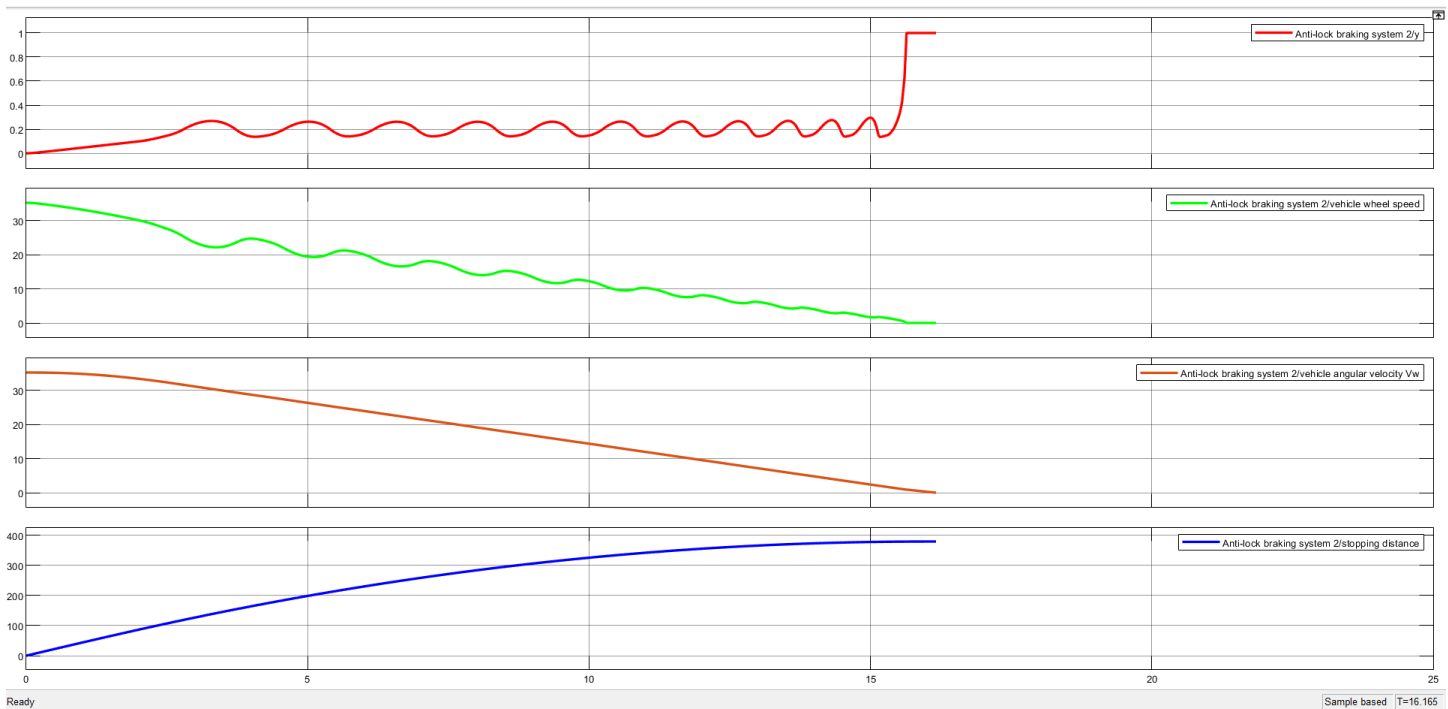
8. Signal Builder to generate test signals.

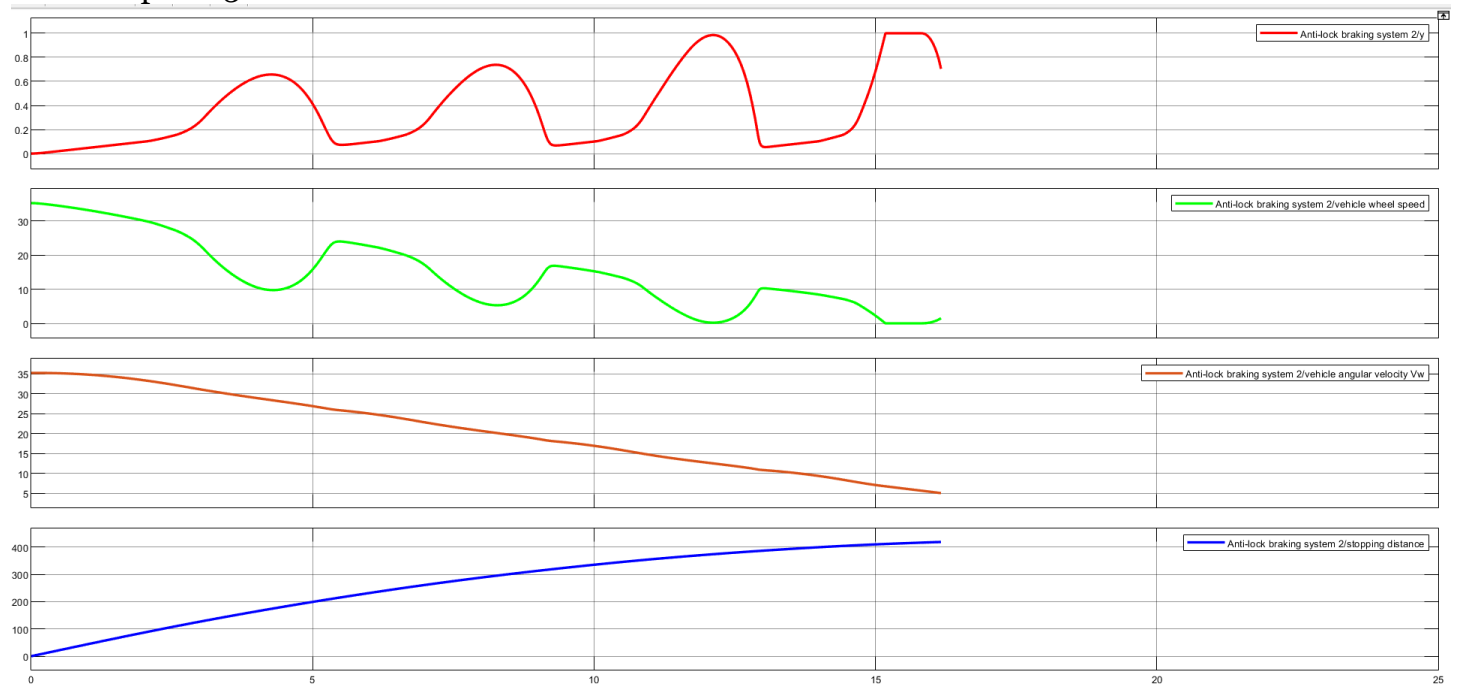To test various signal input values for wheel slip. We can use signal builder block for generating various test inputs.
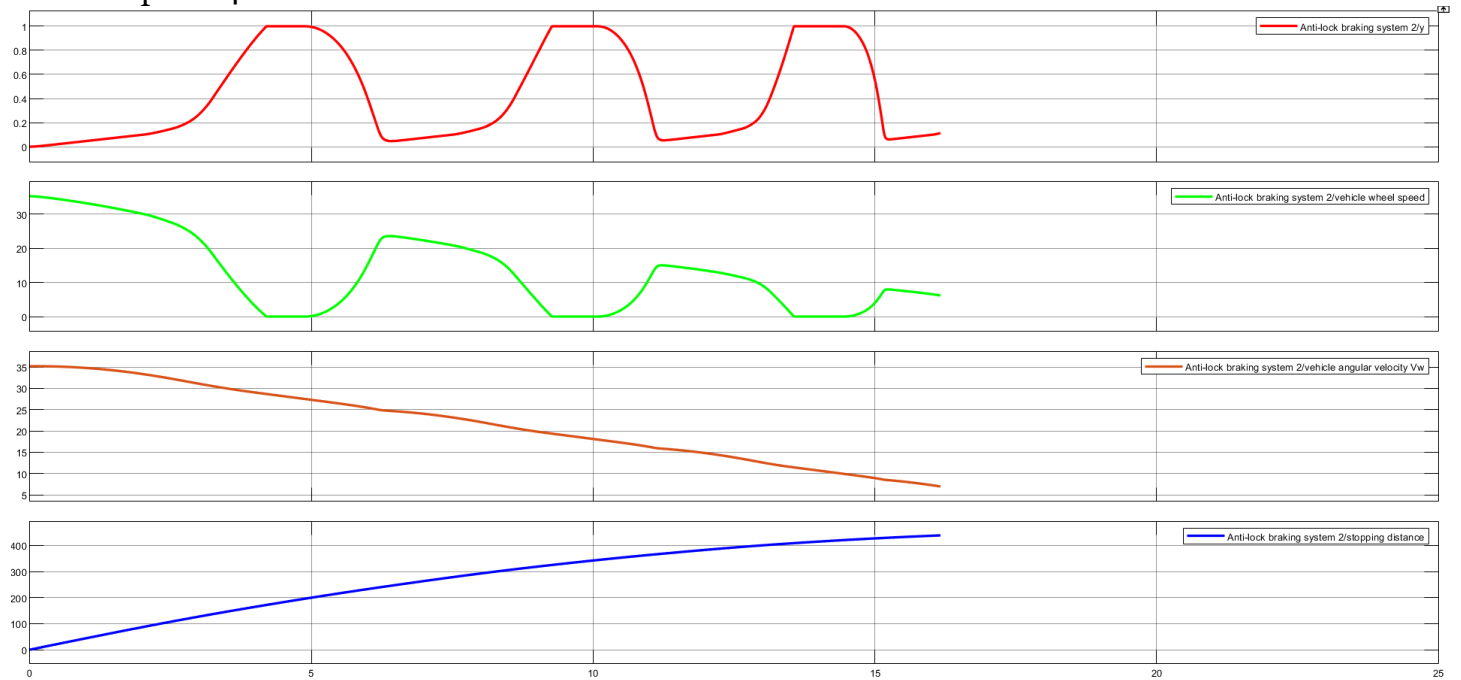

Signal Builder wheel slip test input

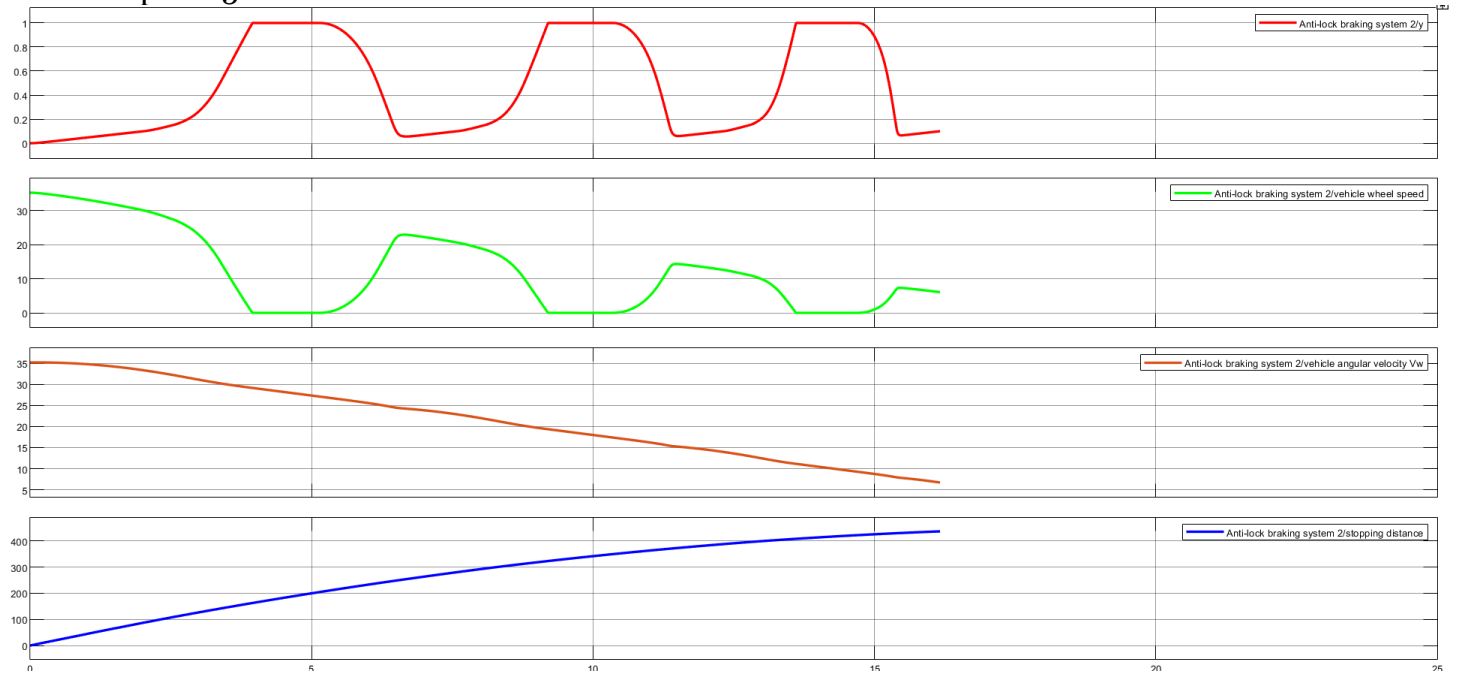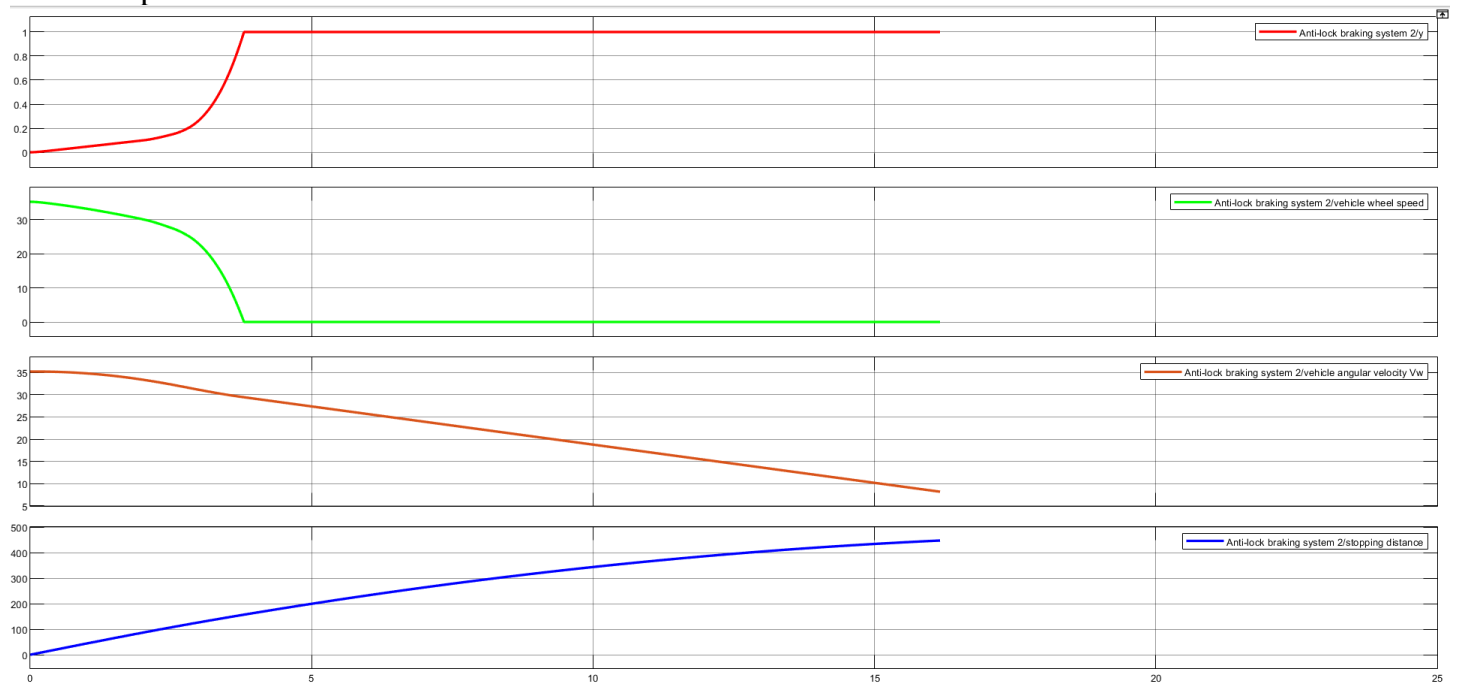Result:
Wheel slip = 0.2

Wheel slip = 0.3

Legend (top to bottom):
- Anti-lock braking system 2/y
- Anti-lock braking system 2/vehicle wheel speed
- Anti-lock braking system 2/vehicle angular velocity Vw
- Anti-lock braking system 2/stopping distance

Wheel slip = 0.4

Legend (top to bottom):
- Anti-lock braking system 2/y
- Anti-lock braking system 2/vehicle wheel speed
- Anti-lock braking system 2/vehicle angular velocity Vw
- Anti-lock braking system 2/stopping distance

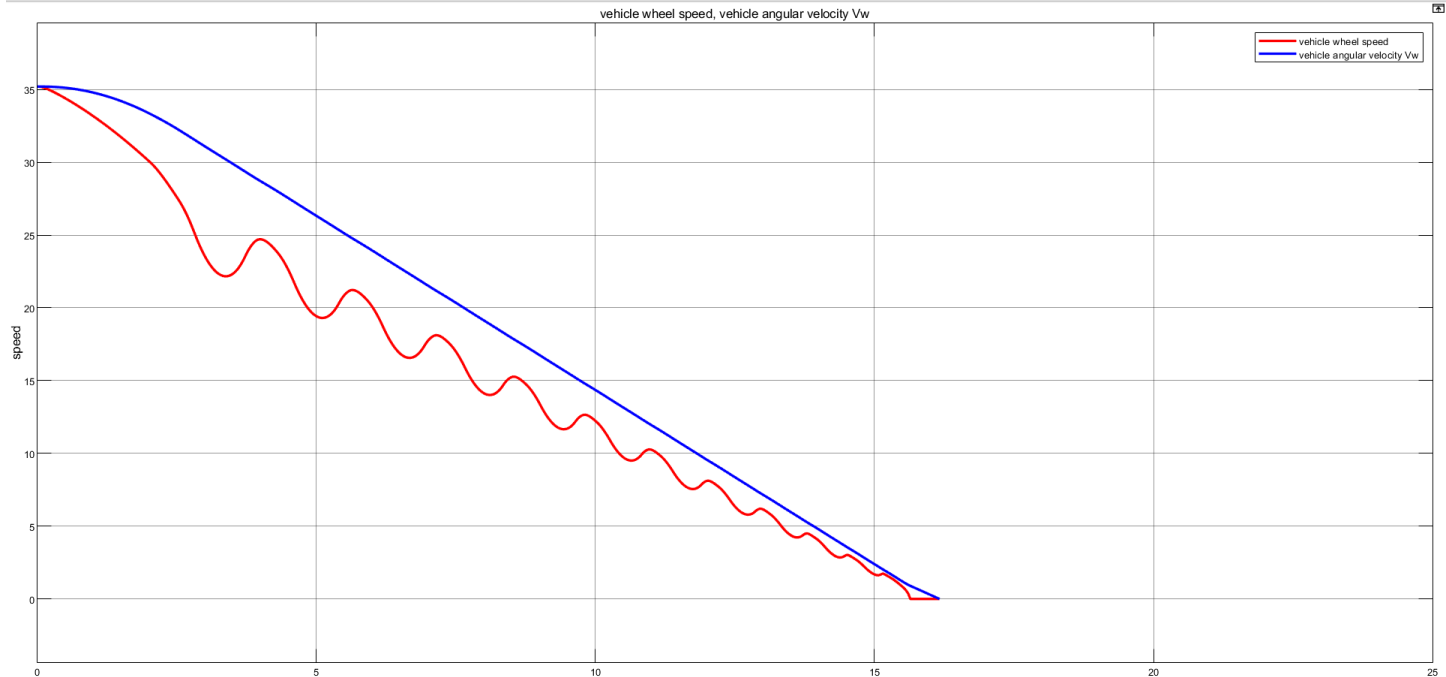## Wheel slip = 0.5

Wheel slip = 1

when slip = 1, as the slip plot shows, the tire is skidding so much on the pavement that the friction force has dropped off.
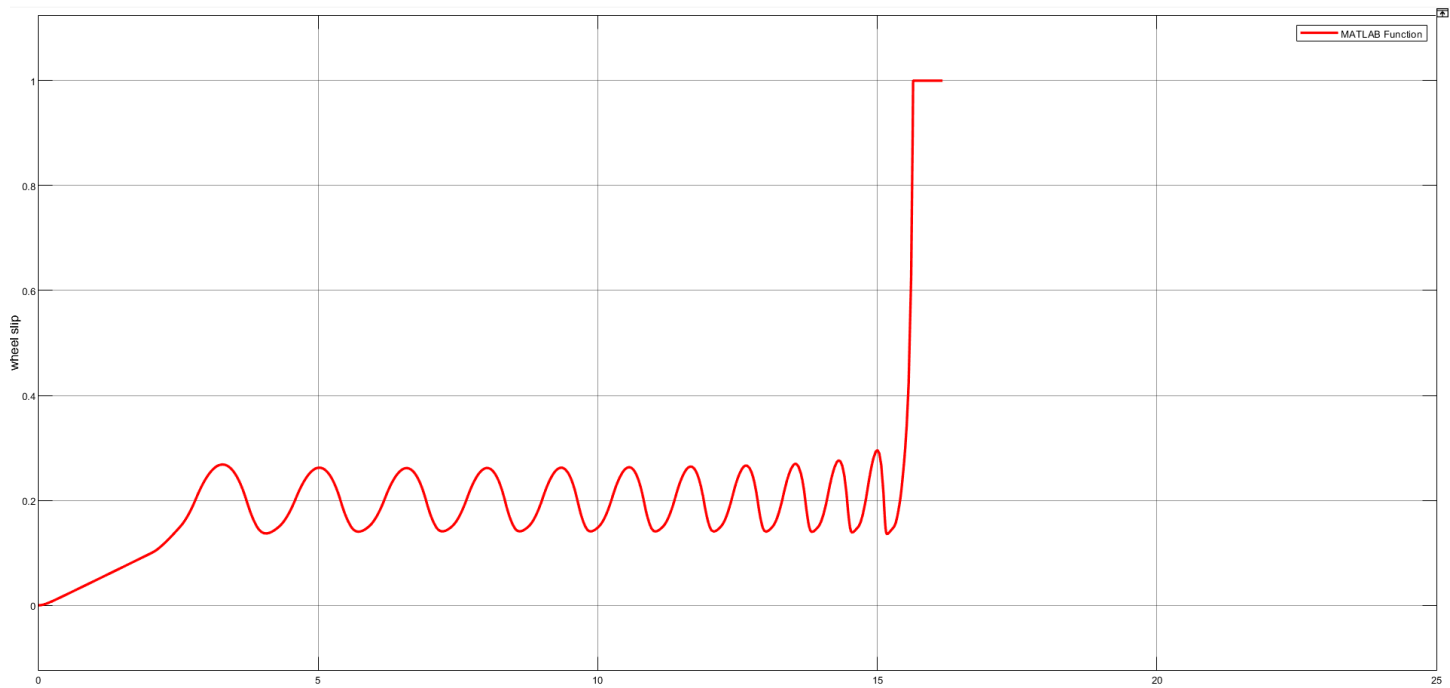
System Result with ABS :

ABS_CONTROL=1   this will enable ABS system.
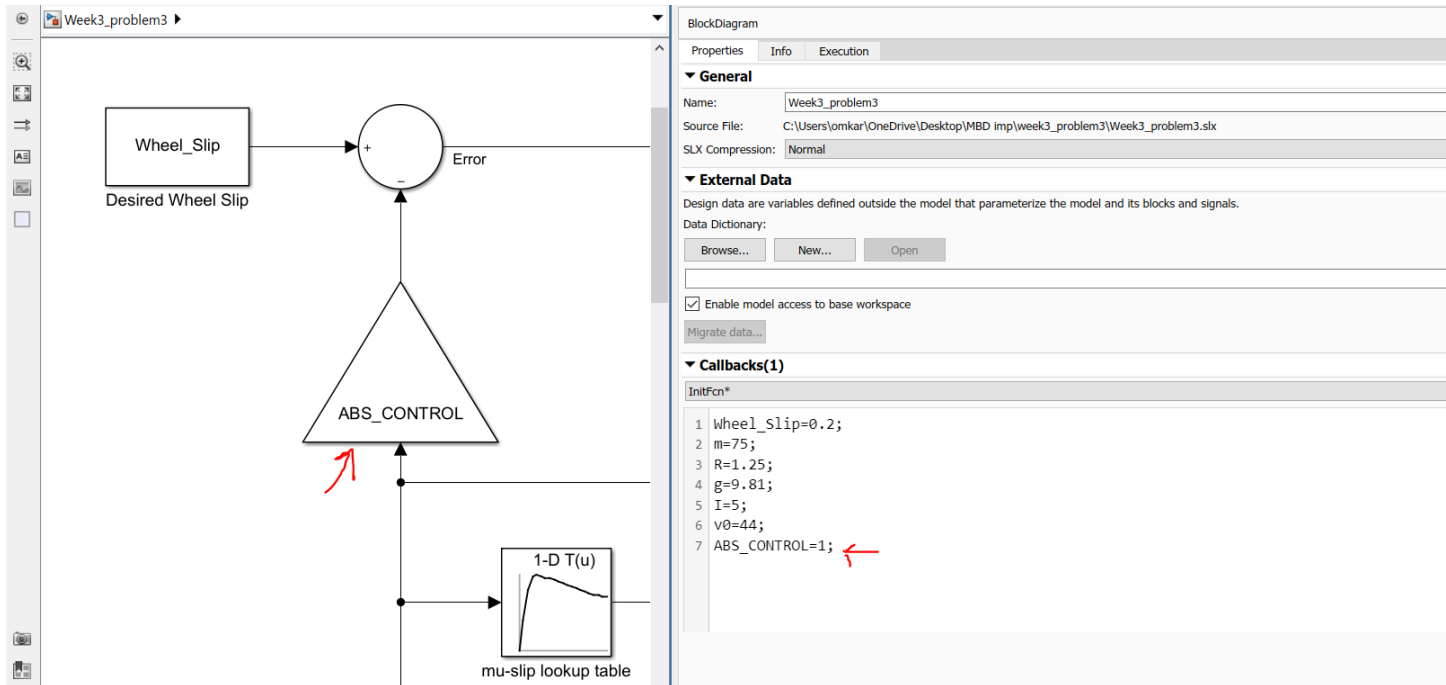
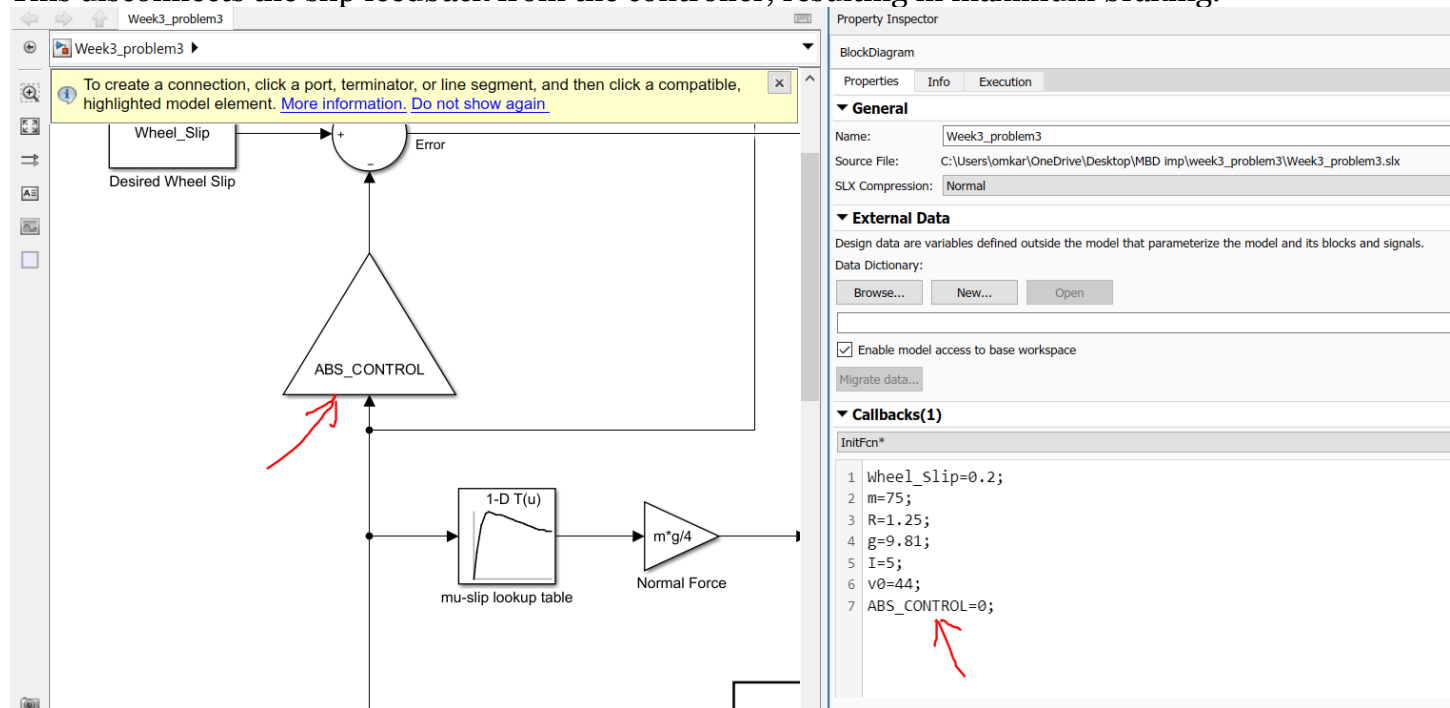Comparison of vehicle speed and wheel speed with ABS enabled

Wheel slip when ABS is enabled.
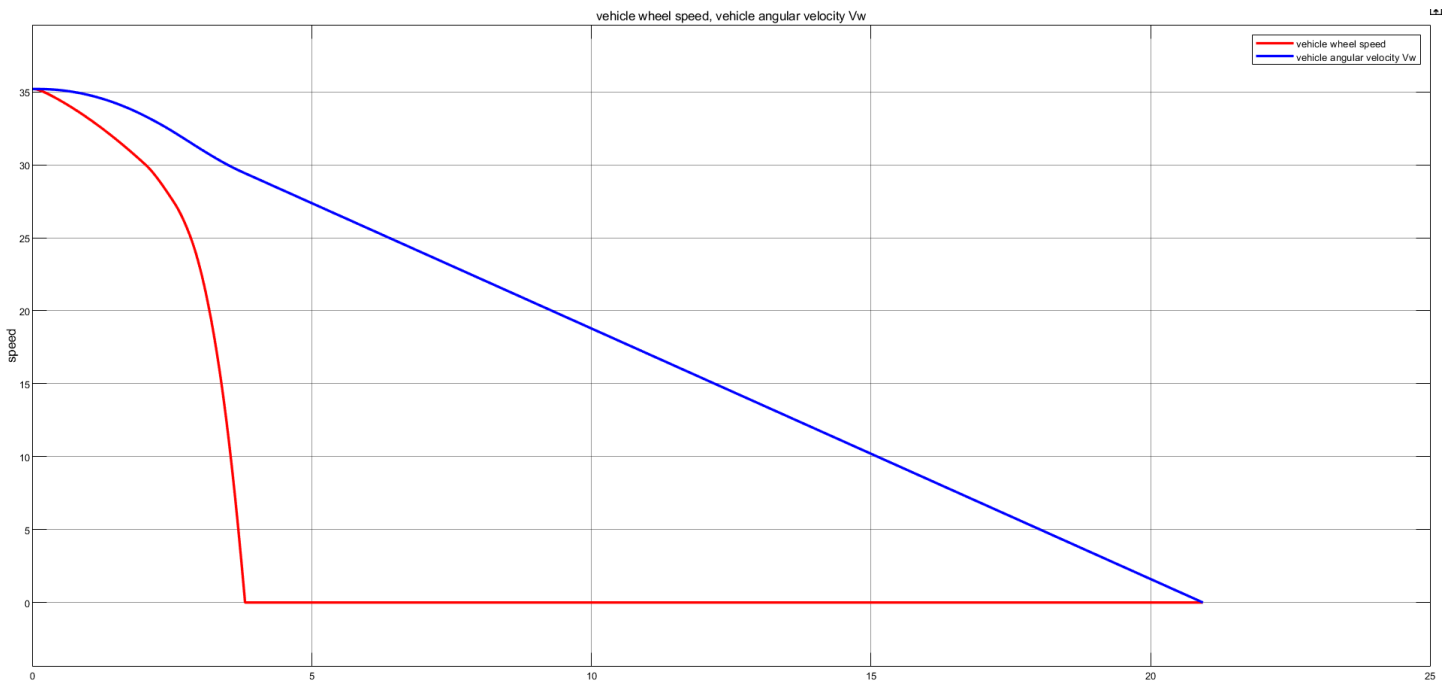

System Result without ABS :

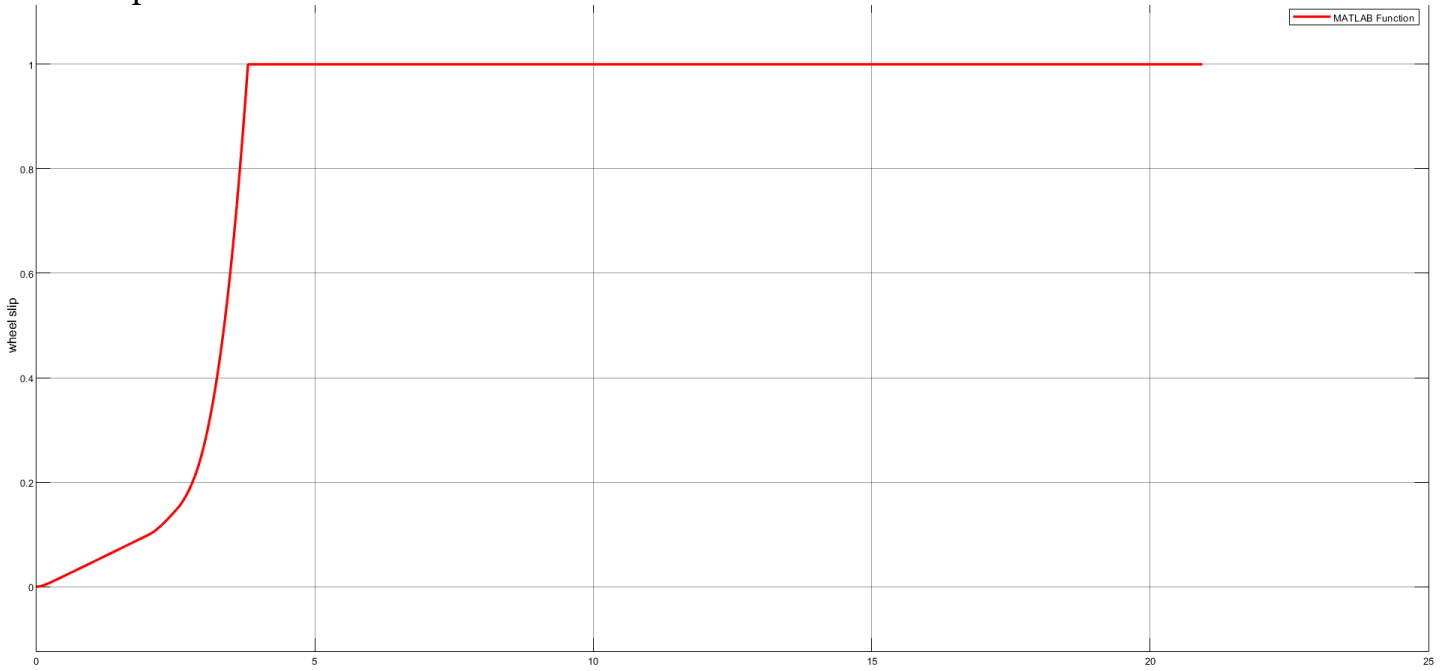ABS_CONTROL=0   this will disable ABS system.

This disconnects the slip feedback from the controller, resulting in maximum braking.
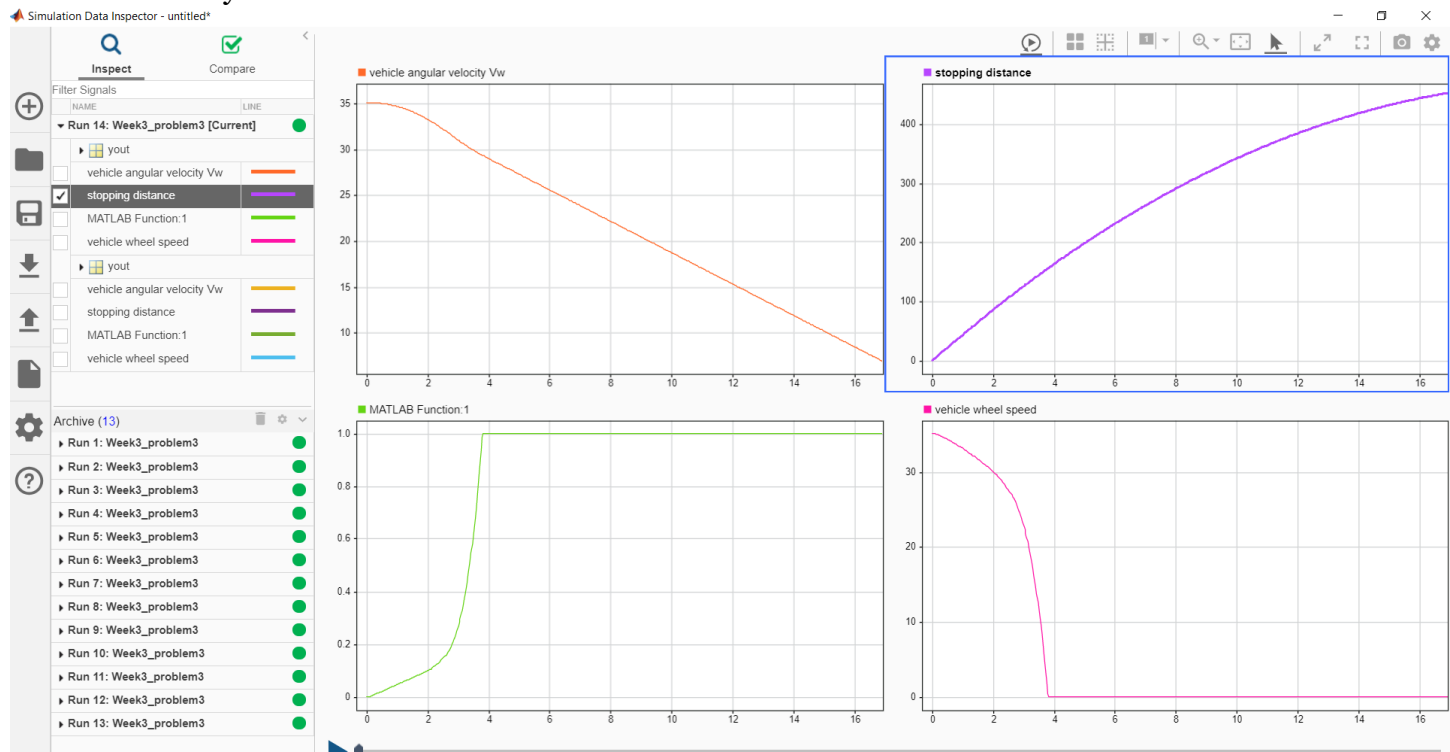
Comparsion of vehicle speed and wheel speed when ABS is disabled.
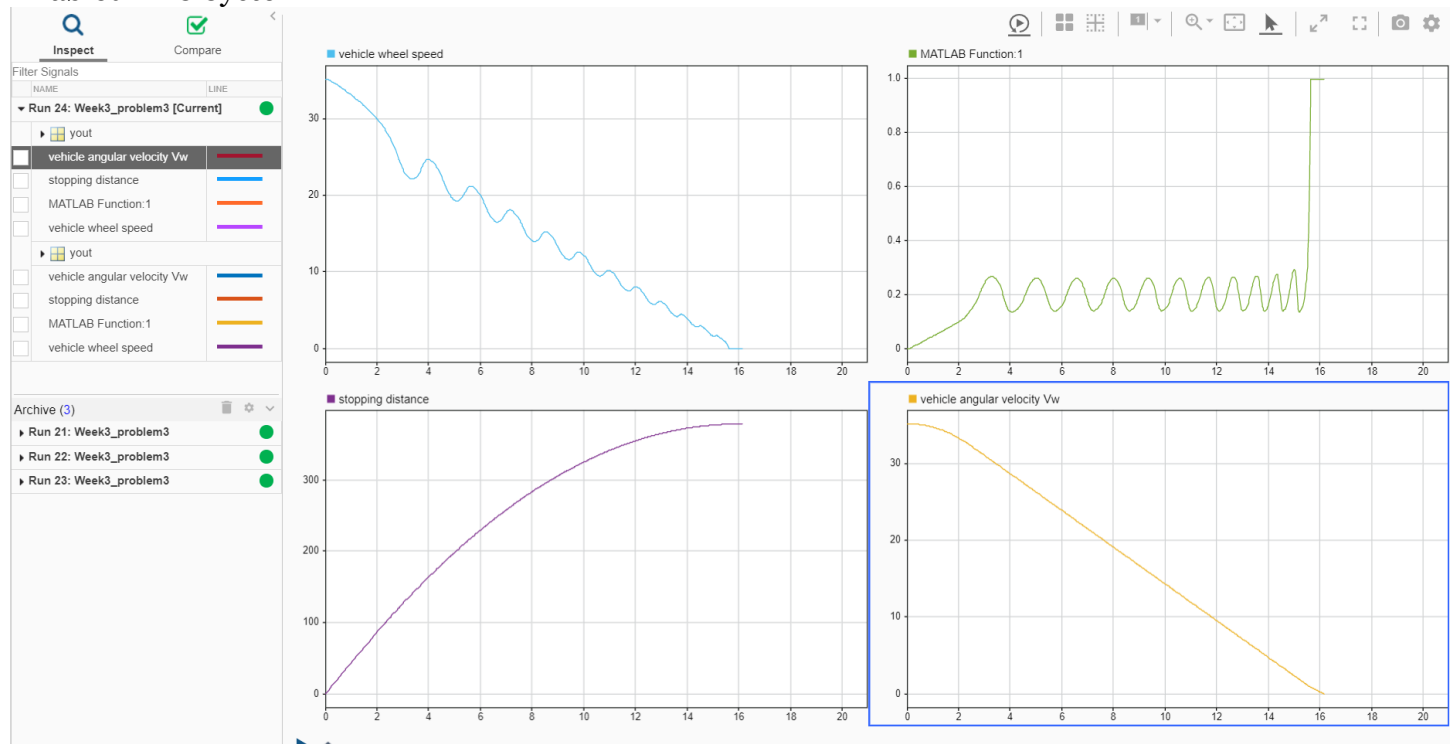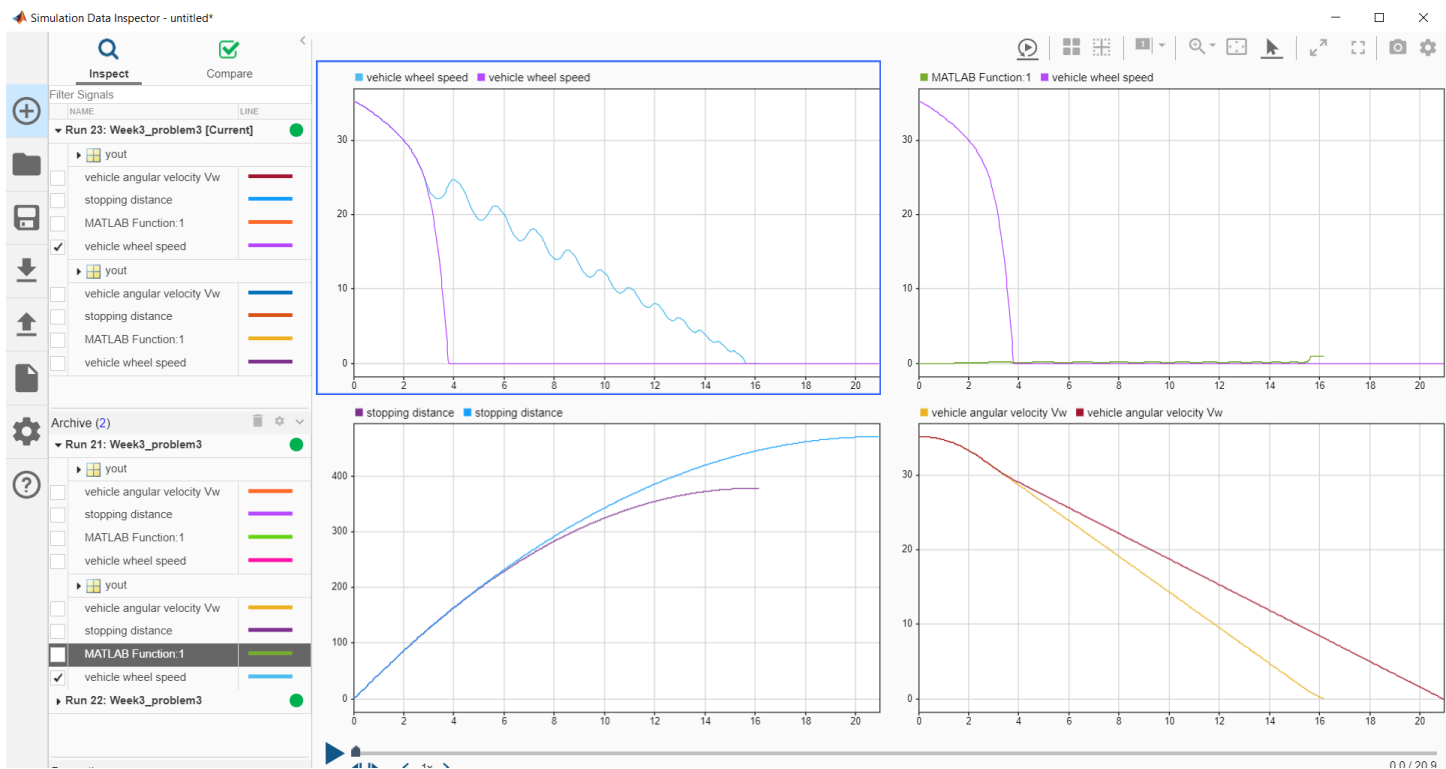

Wheel slip when ABS is disabled.

Disabled ABS system

Enabled ABS System

Data inspector comparison for with disabled and enabled ABS system

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:

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. 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 mode.

9. Reference

[1]   https://in.mathworks.com/help/matlab/math/choose-an-ode-solver.html

[2]   https://in.mathworks.com/help/simulink/callback-functions.html

[3]   https://in.mathworks.com/help/simulink/what-is-a-matlab-function-block.html

[4]   https://in.mathworks.com/help/simulink/ug/choose-a-solver.html

[5]   https://in.mathworks.com/help/simulink/ug/use-auto-solver-to-select-a-solver-1.html

[6]   https://in.mathworks.com/help/physmod/simscape/ug/setting-up-solvers-for-physical-models.html

[7]   https://x-engineer.org/projects/anti-lock-braking-system-abs-modeling-simulation-xcos/

[8]   https://a-lab.ee/edu/sites/default/files/ABS%20System%20Control.pdf