Anti-lock Braking System (ABS) Simulation using Matlab and Simulink Modeling

Anti-lock Braking System (ABS) is a system that automatically adjusts the braking force of the wheels when the vehicle is braking. It can prevent the wheels from locking to obtain the best braking effect and improve the directional stability of the automobile. In this study, the ABS system model based on Matlab/Simulink will be introduced, including single wheel vehicle model, tire model and vehicle braking system model.

Anti-lock braking systems (ABS) are used to control the wheel slip to keep 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. To simulate the braking dynamics of a vehicle, we are going to simulate simplified mathematical models (quarter-car model) for both vehicle and wheel. Also, a simplified ABS controller is going to be implemented to emulate the braking torque in slip conditions.

1. Vehicle Model

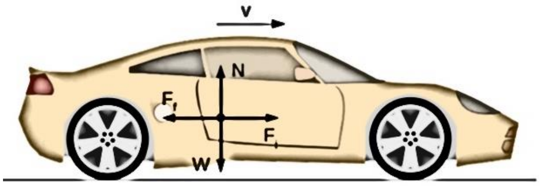

Figure 1: Vehicle under braking conditions [1].

If a vehicle moving in the forward direction, under the braking conditions, the equilibrium equations can be written as [1]:

$$F_f + F_i = 0 \quad \rightarrow \quad F_f = -F_i \tag{1}$$

where F_f is the friction force between the wheel and ground and F_i is the inertial force of the vehicle. For the vertical direction:

$$N + W = 0 \quad \rightarrow \quad N = -W \tag{2}$$

where N is the normal force (road reaction) and W is the vehicle weight. The friction force can be expressed as:

$$F_f = \mu \cdot N \tag{3}$$

where μ is the friction coefficient between the wheel and road. The vehicle's weight is:

$$W = m_v \cdot g \tag{4}$$

Substituting equations (2) and (4) in (3) yields the expression of the friction force as:

$$F_f = -\mu \cdot m_v \cdot g \tag{5}$$

where m_v is the total vehicle mass in kg and g is the gravitational acceleration in m/s^2. The inertia force is the product between the vehicle mass m_v in kg and vehicle acceleration a_v in m/s^2:

$$F_i = m_v \cdot a_v = m_v \cdot \dot{v}_v \tag{6}$$

where v_v is the vehicle speed in m/s. From equations (1), (5) and (6) the vehicle acceleration will be extracted as:

$$\dot{v}_v = \frac{1}{m_v} \cdot (-\mu \cdot m_v \cdot g) \tag{7}$$

Vehicle speed can be obtained by integration of equation (7).

1.1. Wheel Model

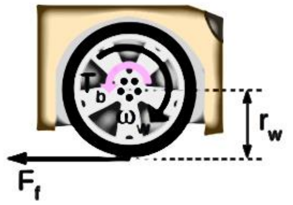

Figure 2: Acting forces during wheel braking [1].

During braking, the driver applies a braking torque T_b in [Nm] to the wheels. The friction force F_f in [N] between the wheel and road creates an opposite torque with the wheel radius r_w in m. For simplification, we can consider the wheel as 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:

$$J_w \cdot \dot{\omega}_w = F_i \cdot r_w - T_b \tag{8}$$

where J_w [kg·m^2] is the wheel's moment of inertia and ω_w [rad/s] is the angular speed of the wheel.
From equation (8) we can extract the expression of the wheel acceleration:

$$\dot{\omega}_w = \frac{1}{J_w} \cdot (F_i \cdot r_w - T_b) \tag{9}$$

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

1.2. Wheel Slip

The ABS system must control the wheel slip λ around an optimal target. The wheel slip is calculated as [1]:

$$\lambda = \frac{v_v - r_w \cdot \omega_v}{v_v} \tag{10}$$

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

$$\omega_v = \frac{v_v}{r_w} \tag{11}$$

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

2. Friction Coefficients Calculation

The friction coefficient between wheel and road depends on several factors, like wheel slip, vehicle speed, and the type of the road surface. For our simulation purpose, we are going to consider only the variation of the friction coefficient function on the longitudinal wheel slip**.** During braking, if the wheel slip is 100 %, the wheel is locked but the vehicle is still moving. At 0 % slip, the wheel and vehicle have the same speed [1]. The optimum friction coefficient (highest value) is obtained when the wheel slip is around 20 %. If the wheel slip enters the unstable area, the friction coefficient will decrease, and the wheel will lock causing skid and vehicle instability. For this example, the ABS systems will have to keep the wheel slip around 20 %, where friction coefficient has the highest values.

Several tire friction models describing the nonlinear behavior are reported in the literature. There are static models as well as dynamic models. The most reputed tire models are by Layne [3] and by Pacejka [4], also known as magic formula and it is derived heuristically from experimental data. In this study, the Burckhardt model [5] will be used, as it is particularly suitable for analytical purpose while retaining a good degree of accuracy in the description of the friction coefficient. The friction coefficients can be expressed as an empirical function, where the wheel slip is a function argument:

$$\mu(\lambda, v) = \left[A\left(1 - e^{-B\cdot\lambda}\right) - C\cdot\lambda\right]e^{-D\cdot\lambda v} \tag{12}$$

where λ is the wheel slip and A, B, C, D are the empirical coefficients. The Simulink model for the friction coefficients versus the wheel slip is shown in Figure 3.

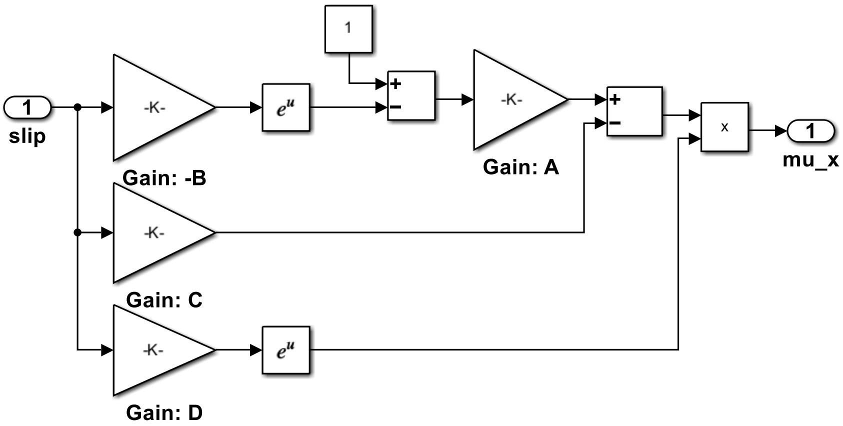

Figure 3: The friction coefficient as a function of the wheel slip

Depending on the value of the coefficients A, B, C and D, the empirical formula (12) can be used to represent the friction coefficient for different road types/states [3],[5].

Surface conditions	A	B	C	D
Asphalt, dry	1.2801	23.99	0.523	0.03
Asphalt, wet	0.857	33.822	0.347	0.03
Concrete, dry	1.1973	25.168	0.5373	0.03
Cobblestones, dry	1.3713	6.4565	0.6691	0.03
Cobblestones, wet	0.4004	33.708	0.1204	0.03
Snow	0.1946	94.129	0.0646	0.03
Ice	0.05	306.39	0	0.03

Using a Matlab script (friction_slip_curves.m), for $v = 30\ m/s$, we can plot the variation of the friction coefficient function of slip, for different road conditions. From Figure 4, we can see that the maximum value of the friction coefficient decreases sharply for a road covered by snow or ice. Even if the value of the friction coefficient is not significantly lower for 100 % slip, preventing wheel lock improves vehicle maneuverability (steering).

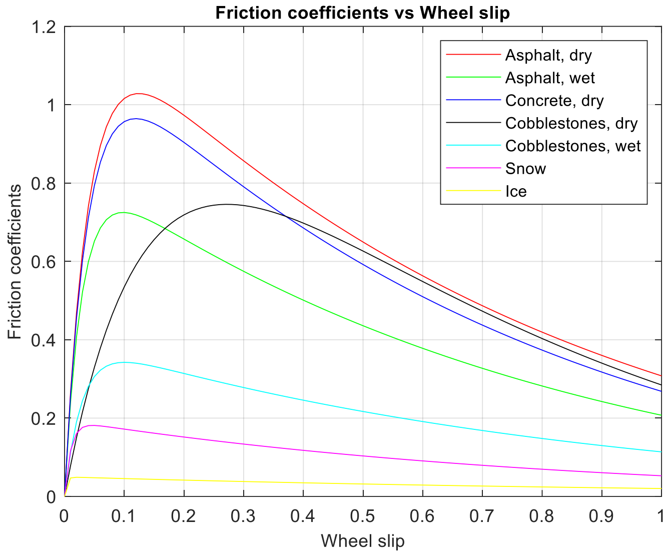

Figure 4: Friction coefficients for different road types

3. Quarter Car Model

The vehicle and wheel model used for the simulation is known in the literature as a quarter-car model. This means that a quarter of the vehicle mass is considered with only one wheel. Also, only the longitudinal vehicle dynamics is considered, disregarding the impact of the suspension system. This example uses a standard set of equations for the dynamics of a quarter car. It contains two continuous time states and is described by the set of non-linear equations (7) to (9). The implementation of quarter car equations is shown in Figure 5.

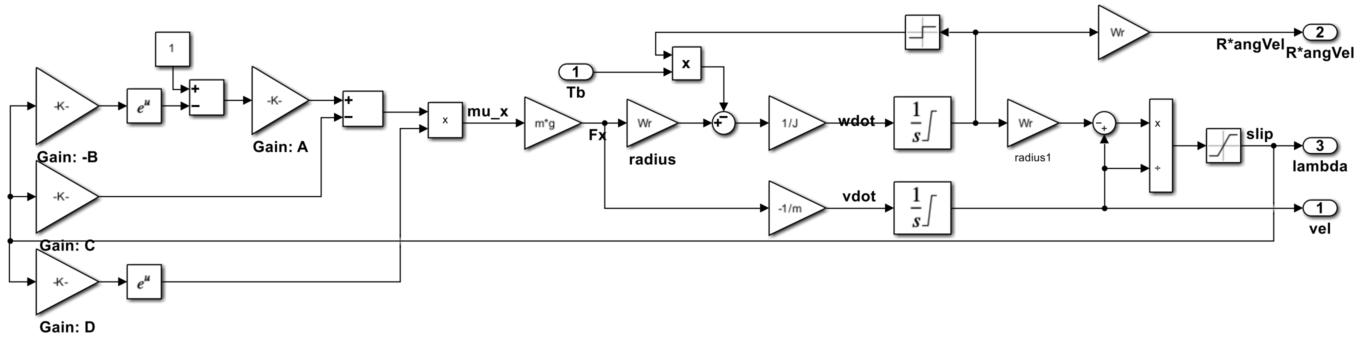

Figure 5: Quarter Car Model Subsystem.

3.1. Actuator/Controller Model

Actuator dynamics, and particular time delays, are often critical to the design of a sufficiently accurate control algorithm. This example uses a simple first order lag in series with a time delay to model the actuator. In practice a second order model is almost always required, and often actuators have different responses when they are opening and closing, and hence need to be modeled in considerably more detail than is done here. The model for the actuator is given by Equation 13:

$$T_b = e^{-\tau s}\frac{a}{s+a}T_{ref} \tag{13}$$

subject to the constraint that $0 < T_b < T_{b\_sat}$.

The first order lag transfer function has been implemented using an integrator, gain and summation/negation block rather than a Transfer Function Block (from the *Continuous* library). This has been done as the Transfer Function block does not allow vector signals as an input, but the current implementation does. Hence the model being developed can more easily be expanded to allow for 4-channels, i.e., one for each wheel on a four wheeled vehicle.

There are many different potential implementations for the controller. Here a simple PI (proportional–integral) controller has been shown to be adequate. The subsystem has been implemented and given a discrete sample rate of T_s = 5ms. Controller gains that have been determined to work reasonably well for the configuration chosen here are K_p = 1200 and K_i = 100000. The implementation of actuator/controller is shown in Figure 6.

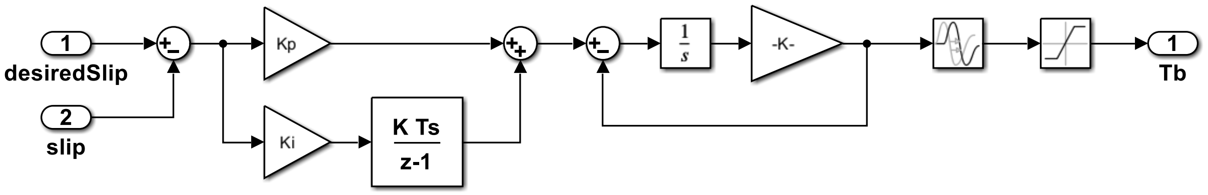

Figure 6: The Actuator/Controller Subsystem.

4. The Complete Model

The control loop developed in this study follows a very standard form. The controller, actuator and quarter car models are all in the feedforward path. The calculated wheel slip (which is to be controlled) is fed back and compared to a desired slip value, with the error fed into the controller. This is shown in Figure 7.

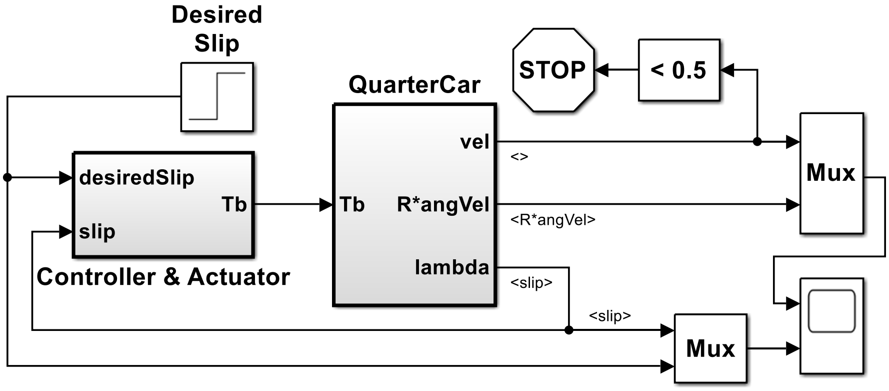

Figure 7: Quarter Car Slip Control Loop.

As shown in Figure 7, the model is fully parameterized and all the parameters are created in the MATLAB workspace by executing the m-file QuarterCarParams.m which is done automatically when the model is opened, and this is achieved by specifying a PreLoadFcn for the model.

5. Simulation Results

Once the model components have been implemented and connected, and the parameters defined, then the model may be simulated. For our simulation example, we are going to use a vehicle with the following parameters: Inertia=1 kgm^2, wheel radius=0.32m, quarter vehicle mass=450kg, and the gravitational force=9.81 m/s^2. The results for the current configuration are shown in Figure 8.

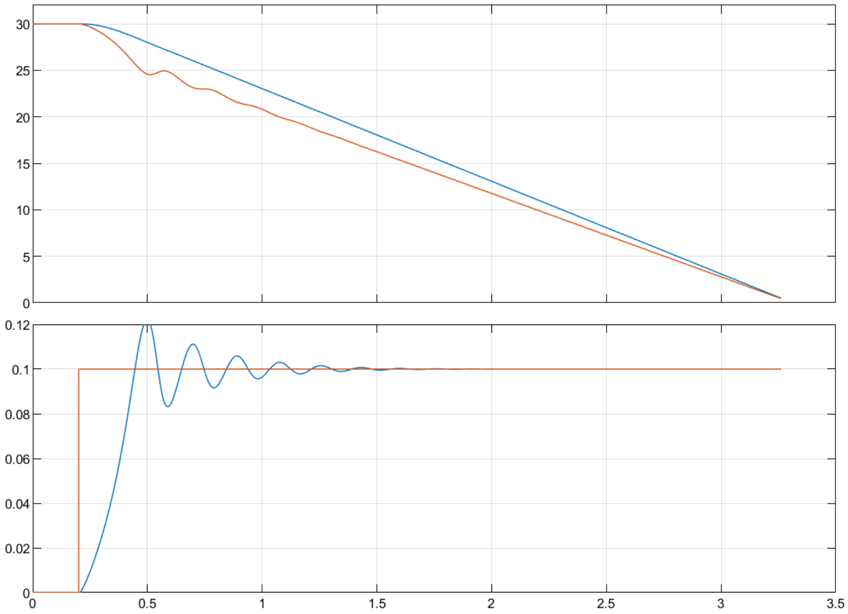

Figure 8: The Simulation Results.

The blue line in Figure 8 shows the vehicle longitudinal velocity and the red line shows the wheel's linear velocity (calculated as the wheel radius multiplied by the angular velocity). When there is no wheel slip, the two lines are the same. When the red line is below the blue line there is wheel slip. Figure 8 shows that the vehicle is initially moving forward at 30m/s with no wheel slip. At T=0.2s a slip of $\lambda = 0.1$ is demanded, which is achieved by applying the brake to the wheel. Consequently, the vehicle decelerates reducing its speed down towards zero.

References:

[1] Anti-lock braking system (ABS) modeling and simulation (Xcos), https://x-engineer.org/anti-lock-braking-system-abs-modeling-simulation-xcos/

[2] Simulink – Slip Control of a Quarter Model, https://www.goddardconsulting.ca/simulink-quarter-car-model.html

[3] J. R. Layne, K. M. Passino, and S. Yurkovich, Fuzzy learning control for antiskid braking systems, *IEEE Trans. Contr. Syst. Technol.*, Volume. 1, pp. 122-129.

[4] Pacejka, H. B. and R. S. Sharp, Shear force developments by pneumatic tires in steady-state conditions: A review of modeling aspects. *Vehicle Systems Dynamics* 29, 409-422

[5] Burckhardt, M, Fahrwerktechnik: Radschlupf-Regelsysteme. Würzburg: Vogel Verlag.