ECE 5553 Final Project

Brian Lesko

Kevin lastname

Ovee Dankithcar

Contents

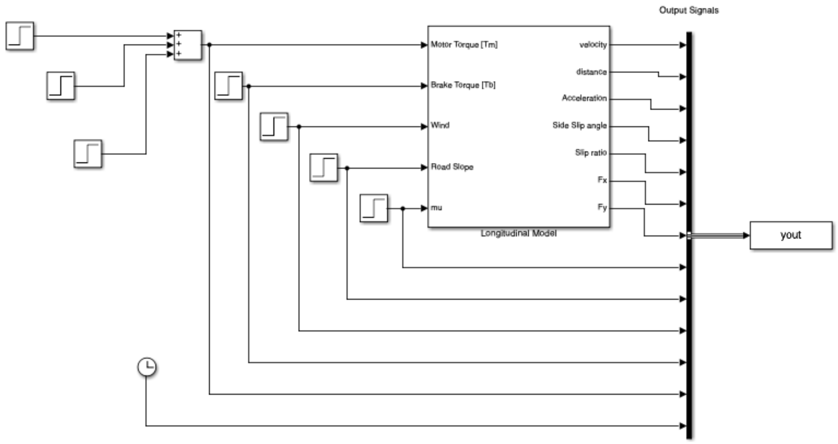
Longitudinal dynamics parameters

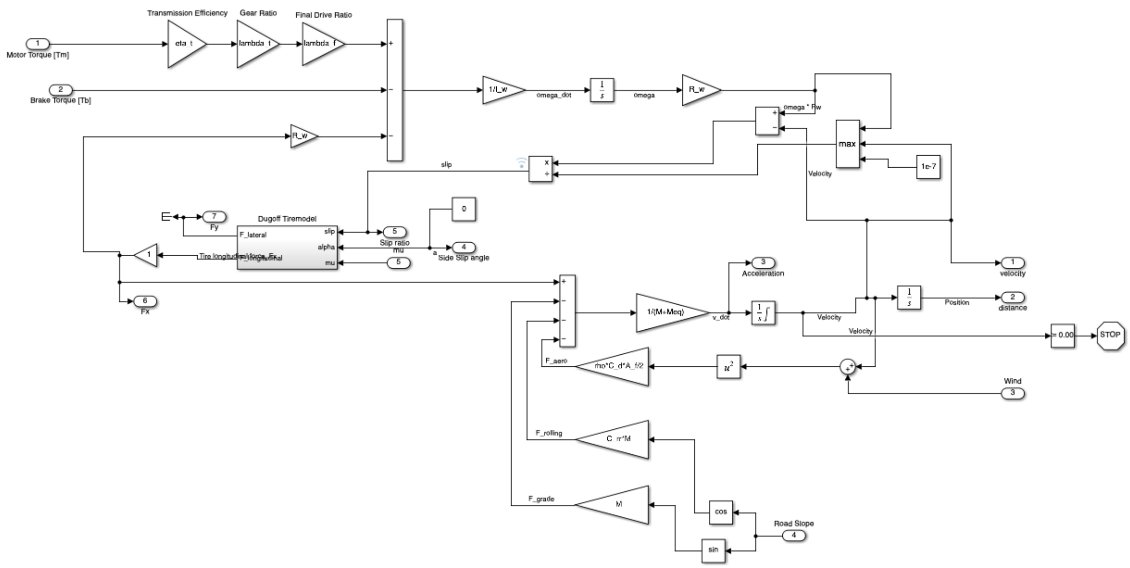
Deriving more accurate parameters by modifying parameters of a sedan, used in a previous project, below:

```
clc; clear; close all;

% longitudinal dynamics parameters for a sedan
    % M=2000; % Mass of the vehicle[Kg]
    % eta_t=0.9; % Transmission Efficiency
    % lambda_t=1.0; % Gear Ratio
    % lambda_f=4.1; % Final Drive Ratio
    % I_w=1; % Inertia of the Wheel[kgm2]
    % R_w=0.3; % Wheel Radius[m]
    % Meq=0.1*M; % Equivalent Mass Factor
    % C_d=0.29; % Drag coefficient
    % rho=1.225; % Air density[kg/m3]
    % A_f=2.8;% Frontal area    [m2]
    % g=9.81;% [N/m2]
    % C_rr=0.015;%Rolling resistance coefficient
    % C_x=3e5; %Longitudinal Stiffness[N]
    % C_alpha = 1.5e5; % Cornering Stiffness for 1 tire [N/rad]
    % L=2.85;% Wheelbase[m]
    % l_f=1.3;% Distance from the center of gravity of the vehicle (CG) to
the front axle [m]
```

```
    % l_r=1.55;% Distance from the center of gravity of the vehicle (CG) to
the rear axle [m]
    % C_s=1.5e5;% Cornering Stiffness of Front and Rear Tires [N/rad]
    % I_z=3700;% Inertia moment around z axis J or Iz [kg/m2]
    % mu=0.7; % Road friction coefficient

% for the shuttle
    M=3500; % Mass of the vehicle[Kg] based on Tesla Model X size and
weight, the shuttle will be 2k kg heavier
    eta_t=0.8; % Transmission Efficiency, less efficient than a sedan
    lambda_t=1.0; % Gear Ratio, the same as the sedan
    lambda_f=4; % Final Drive Ratio, same as the sedan
    I_w=1; % Inertia of the Wheel[kgm2], same as the sedan
    R_w=0.3; % Wheel Radius[m], same as the sedan
    Meq=0.1*M; % Equivalent Mass Factor, same as the sedan
    C_d=0.4; % Drag coefficient, Worse drag than the sedan
    rho=1.225; % Air density[kg/m3], same as the sedan (obviously)
    A_f=6;% Frontal area [m2], more frontal area than the sedan
    g=9.81;% [N/m2], same as the sedan
    C_rr=0.015;%Rolling resistance coefficient, same as the sedan

    C_x=3e5; %Longitudinal Stiffness[N], same as the sedan
    C_alpha = 1.5e5; % Cornering Stiffness for 1 tire [N/rad], same as the
sedan

    L=4.75;% Wheelbase[m], longer by 2 meters than the sedan
    l_f=2.25;% Distance from the center of gravity of the vehicle (CG) to
the front axle [m]
    l_r=2.5;% Distance from the center of gravity of the vehicle (CG) to
the rear axle [m]
    C_s=1.5e5;% Cornering Stiffness of Front and Rear Tires [N/rad]
    I_z=4000;% Inertia moment around z axis J or Iz [kg/m2] Higher
    mu=0.7; % Road friction coefficient same

[M, eta_t, lambda_t, lambda_f, I_w, R_w, Meq, C_d, rho, A_f, g, C_rr, C_x,
C_alpha,L, l_f, l_r, C_s, I_z, mu] = getLogitudinalParameters();
```

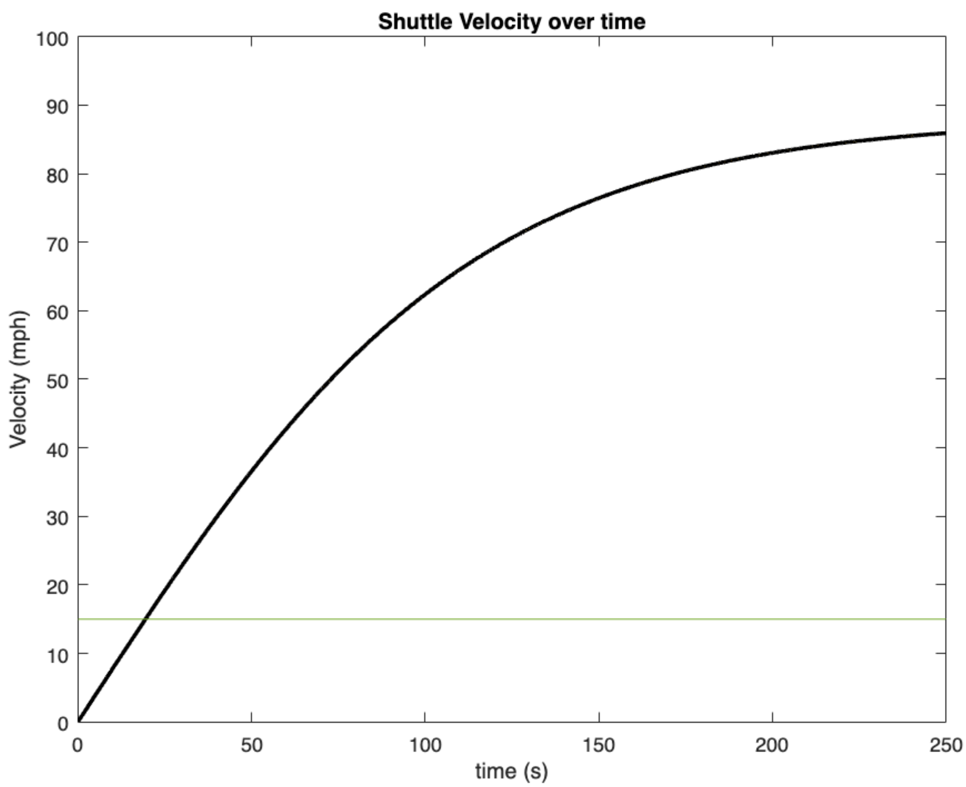
Longitudinal Dynamics via Duggoff model






```
% unchanging Sim parameters for pure acceleration
[motorStart,motorStartTime,motorIncrease,motorIncreaseTime,motorDecrease,mot
orDecreaseTime,brakeTime,breakStrength,headWindTime,headWind,roadSlopeTime,r
oadSlope] = setLongitudinalInputsToZero();
[M, eta_t, lambda_t, lambda_f, I_w, R_w, Meq, C_d, rho, A_f, g, C_rr, C_x,
C_alpha,L, l_f, l_r, C_s, I_z, mu] = getLogitudinalParameters();

motorStart = 65;
addpath("Simulink_Models/")
sim("longitudinal_modelV2.slx")
```

```
ynames = ["V","X","A","Side Slip Angle","Slip Ratio","Tire force
Fx","Tire Force Fy","Motor Torque","Brake Torque","Wind Force","Road
Slope","Mu","time"];
sim1 = yout;
time1 = sim1(:,length(ynames));
plot(time1,2.23694*sim1(:,1),'k','Linewidth',2), ylabel('Velocity (mph)'),
xlabel('time (s)'), title('Shuttle Velocity over time'), hold on
axis([0 250 0 100])
% horizontal line
line_x = linspace(0,250,4); line_y = 15 * ones(1,length(line_x));
plot(line_x,line_y)
```

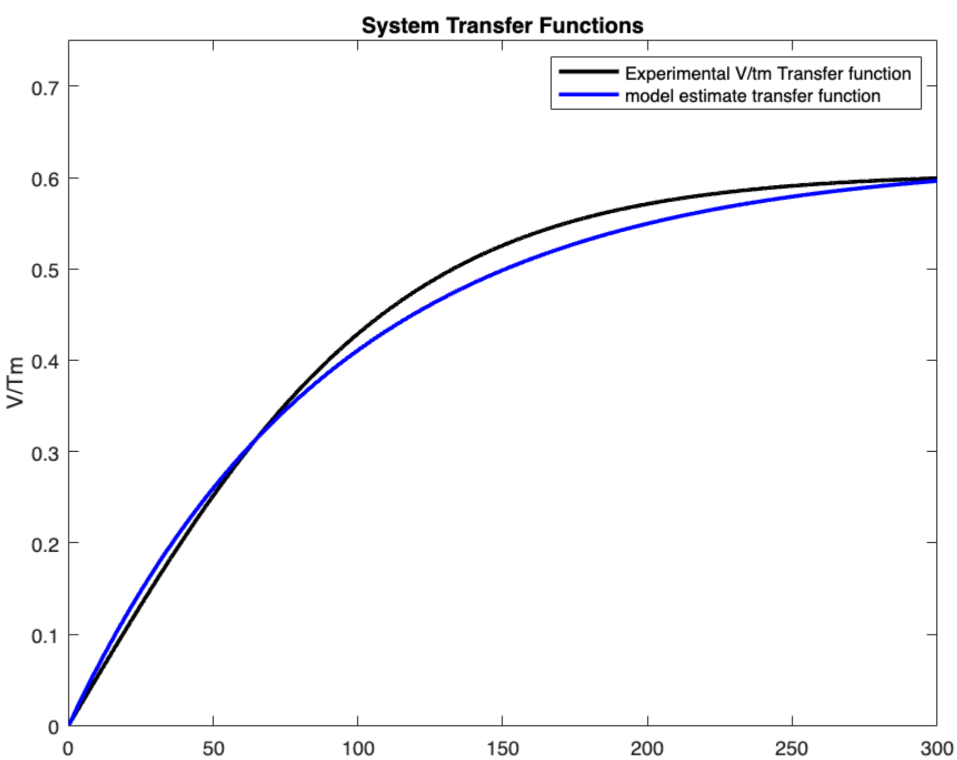


The longitudinal dynamics model used to make this curve originates from the duggoff model which uses the parameters chosen in the previous section. With the desired cruising velocity of 15 mph, the shuttle can reach this speed after ~ 20 seconds, which is reasonable for the low speed application of this shuttle. Overally, this is likely a reasonable velocity time curve for standard use. It is however built off the nonlinear duggoff tire model and is higher fidelity than needed in ongoing steps of the project. Therefor, in the next sectioon this curve's transfer function will be described and approximated using a simple linear system.

LTI System Approximation

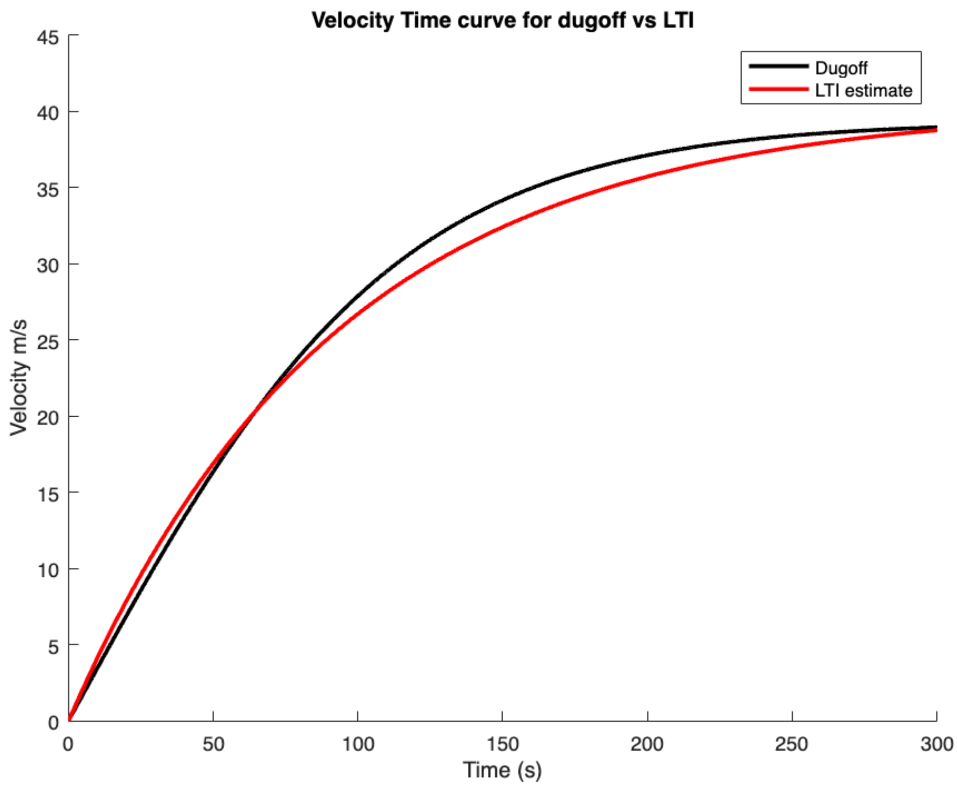
The system that creates the transfer function above can be described as a linear single input single output transfer function with Velocity as the output and Motor torque as the input. Therefor, the simulated transfer function can be displayed by plotting the simulation Velocity values divided by the motor torque, which we chose as constant in the last section.

```
motor_torque = motorStart;
velocity = sim1(:,1);

figure()
plot(time1,velocity/motor_torque,'k','Linewidth',2), ylabel('System'),
xlabel(''), title('Shuttle Velocity / Motor Torque'), hold on

K = 0.62;
T = 92;

s = tf('s');
V_tf = K/ (T*s + 1);

[h,g] = step(V_tf);
plot(g, squeeze(h), 'b','LineWidth',2)

legend('Experimental V/tm Transfer function','model estimate transfer
function')
title('System Transfer Functions')
ylabel('V/Tm')
axis([0 300 0 .75])
```



The LTI system approximates the duggoff model around our motor torque of interest of 65 Nm, lets reproduce the velocity time curve from this LTI system and compare it to the simulated duggoff model

```
figure(), hold on
plot(time1,sim1(:,1),'k','Linewidth',2)
plot(g,h*motor_torque,'Linewidth',2,'Color',[1 0 0])
xlabel('Time (s)'), ylabel('Velocity m/s'), title('Velocity Time curve for
dugoff vs LTI'), legend('Dugoff','LTI estimate'), axis([0 300 0 45])
```



This curve is just a scaled version of the previous one.

Lateral Dynamics Parameters

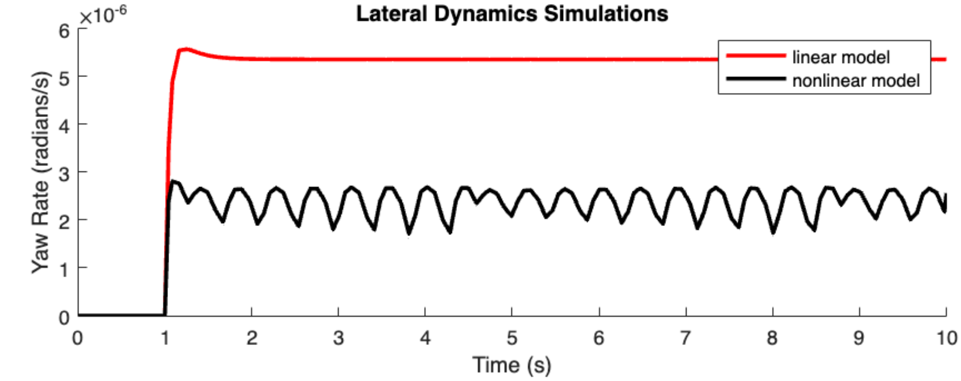
```
% These Lateral parameters have overlap with the longitudinal values
% chosen, and have been set to match

    % L=4.75; % Distance between the axles [m], matched with first set
    % g=9.81; %
    % Lr = 2.5;% Distance from the center of gravity of the vehicle (CG) to
the rear axle, matched with first set
    % Lf = 2.25; % Distance from the center of gravity of the vehicle (CG)
to the front axle, matched with first set
    % Cf = 1.5e5; % Cornering Stiffness of Front Tires x2, matched with
first set
```

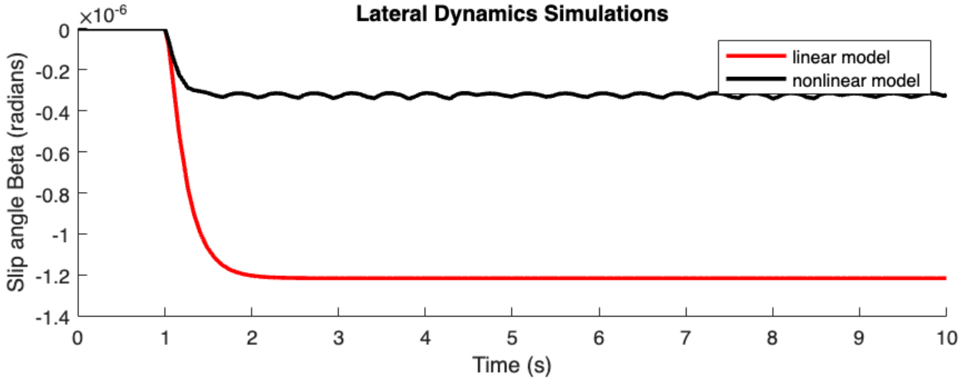
```
    % Cr = 1.5e5; % Cornering Stiffness of Rear Tires x2, matched with
first set
    % Cs = 1.5e5; % Cornering Stiffness, matched with first set
    % m = 3500; %Mass of the vehicle [kg], matched with first set
    % J = 4000; %Yaw moment of Inertia, matched with first set
    % mu = 0.7; %Dry coefficient of Friction, matched with first set
    % R = 0.3; % Wheel radius, matched with first set
    % Vref = 20; % Constant vehicle velocity.
    % alphaf = 0.1; % Steering wheel angle rad/sec
    %
    % %Linear Parameters Calculation
    % a11 = -(Cr+Cf)/(mVref);
    % a12 = -1-((CfLf-CrLr)/(mVref^2));
    % a21 = (LrCr-LfCf)/J;
    % a22 = -((CfLf^2)+(CrLr^2))/(VrefJ);
    % b11 = Cf/(mVref);
    % b12 = Cr/(mVref); %delta_r parameter
    % b21 = CfLf/J;
    % b22 = Cr*Lr/J; %delta_r parameter
    % e2 = 1/J; % For yaw moment term

[L,g,Lr,Lf,Cf,Cr,Cs,m,J,mu,R,Vref,alphaf,a11,a12,a21,a22,b11,b12,b21,b22,e2]
 = getLateralParameters();
```

Lateral Dynamics Simulation

Lets use the lateral model comparison simulink from an earlier project

```
sim('lateral_models')
figure
subplot(2,1,1), hold on
plot(time,yawRateLin,'r','linewidth',2)
plot(time,yawRateNonLin,'k','linewidth',2)
legend('linear model','nonlinear model'), title('Lateral Dynamics
Simulations'),xlabel('Time (s)'), ylabel('Yaw Rate (radians/s)')
subplot(2,1,2), hold on
plot(time,betaLin,'r','linewidth',2)
plot(time,betaNonLin,'k','linewidth',2)
legend('linear model','nonlinear model'), title('Lateral Dynamics
Simulations'),xlabel('Time (s)'), ylabel('Slip angle Beta (radians)')
```





These figures showcase the difference between the bicycle model with linear tires vs nonlinear dugoff tire models. The linear tires overestimate both the yaw rate and side slip angle, but resemble the same shape.

Path Following model

Functions

Fetch longitudinal parameters

```
function [M, eta_t, lambda_t, lambda_f, I_w, R_w, Meq, C_d, rho, A_f, g,
C_rr, C_x, C_alpha,L, l_f, l_r, C_s, I_z, mu] = getLogitudinalParameters()
    M=2000; % Mass of the vehicle[Kg]
    eta_t=0.9; % Transmission Efficiency
    lambda_t=1.0; % Gear Ratio
    lambda_f=4.1; % Final Drive Ratio
    I_w=1; % Inertia of the Wheel[kgm2]
    R_w=0.3; % Wheel Radius[m]
    Meq=0.1*M; % Equivalent Mass Factor
    C_d=0.29; % Drag coefficient
    rho=1.225; % Air density[kg/m3]
    A_f=2.8;% Frontal area    [m2]
    g=9.81;% [N/m2]
    C_rr=0.015;%Rolling resistance coefficient
```

```
    C_x=3e5; %Longitudinal Stiffness[N]
    C_alpha = 1.5e5; % Cornering Stiffness for 1 tire [N/rad]
    L=2.85;% Wheelbase[m]
    l_f=1.3;% Distance from the center of gravity of the vehicle (CG) to
the front axle [m]
    l_r=1.55;% Distance from the center of gravity of the vehicle (CG) to
the rear axle [m]
    C_s=1.5e5;% Cornering Stiffness of Front and Rear Tires [N/rad]
    I_z=3700;% Inertia moment around z axis J or Iz [kg/m2]
    mu=0.7; % Road friction coefficient
end
function
[motorStart,motorStartTime,motorIncrease,motorIncreaseTime,motorDecrease,mot
orDecreaseTime,brakeTime,breakStrength,headWindTime,headWind,roadSlopeTime,r
oadSlope] = setLongitudinalInputsToZero()
    motorStart = 0; motorStartTime = 0;
    motorIncrease = 0;motorIncreaseTime = 0;
    motorDecrease = 0;motorDecreaseTime = 0;
    brakeTime = 0;breakStrength = 0;
    headWindTime = 0;headWind = 0;
    roadSlopeTime = 0;roadSlope = 0;
end
```

Sample a vector evenly to make it length b

```
function y = sampVec(x, b)
    % x: input vector
    % b: length of output vector
    a = length(x);
    y = linspace(1, a, b)';
    y = round(y);
    y = x(y);
end
```

Fetch stored Lateral Parameters

```
function
[L,g,Lr,Lf,Cf,Cr,Cs,m,J,mu,R,Vref,alphaf,a11,a12,a21,a22,b11,b12,b21,b22,e2]
 = getLateralParameters()
    L=4.75; % Distance between the axles [m], matched with first set
    g=9.81; %
    Lr = 2.5;% Distance from the center of gravity of the vehicle (CG) to
the rear axle, matched with first set
    Lf = 2.25; % Distance from the center of gravity of the vehicle (CG) to
the front axle, matched with first set
    Cf = 1.5e5; % Cornering Stiffness of Front Tires x2, matched with first
set
    Cr = 1.5e5; % Cornering Stiffness of Rear Tires x2, matched with first
set
    Cs = 1.5e5; % Cornering Stiffness, matched with first set
    m = 3500; %Mass of the vehicle [kg], matched with first set
```

```
    J = 4000; %Yaw moment of Inertia, matched with first set
    mu = 0.7; %Dry coefficient of Friction, matched with first set
    R = 0.3; % Wheel radius, matched with first set
    Vref = 20; % Constant vehicle velocity.
    alphaf = 0.1; % Steering wheel angle rad/sec

    %Linear Parameters Calculation
    a11 = -(Cr+Cf)/(m*Vref);
    a12 = -1-((Cf*Lf-Cr*Lr)/(m*Vref^2));
    a21 = (Lr*Cr-Lf*Cf)/J;
    a22 = -((Cf*Lf^2)+(Cr*Lr^2))/(Vref*J);
    b11 = Cf/(m*Vref);
    b12 = Cr/(m*Vref); %delta_r parameter
    b21 = Cf*Lf/J;
    b22 = Cr*Lr/J; %delta_r parameter

    e2 = 1/J; % For yaw moment term
end
```