
```

function [rdd,rd,r,Bdd,Bd,B,Keff,Beff,rTF,BTF,kus,V_crit,Fyf,Fyr] =
    car_sim(input_values,car_params)
%CAR_SIM Perform one time step of the simulation of the yaw model
% Detailed explanation goes here

V = input_values.V;
del = input_values.del; % Steer angle
deld = input_values.deld; % Derivative of Steer angle
r = input_values.r;
rd = input_values.rd; % Derivative of yaw rate
B = input_values.B; % Beta, body slip
Bd = input_values.Bd; % Derivative of body slip
dt = input_values.dt; % time step

caf = car_params.caf; % front axle cornering stiffness
car = car_params.car; % rear axle cornering stiffness
Iz = car_params.Iz;
a = car_params.a;
b = car_params.b;
m = car_params.m;
L = car_params.L;
use_duggoff = car_params.use_duggoff;
use_tire_relaxation = car_params.use_tire_relaxation;

% Vy = V*sin(B);
% Vx = V*cos(B);
Vy = V*B;
Vx = V;

alphaf = (Vy + a*r)./Vx - del;
alphan = (Vy - b*r)./Vx;
Wf = car_params.Wf;
Wr = car_params.Wr;

% Adjust cornering stiffness according to duggoff tire model
if use_duggoff == 1
    mu = car_params.mu;
    f_lamb_f = f_lambda(alphaf,caf,Wf,mu);
    f_lamb_r = f_lambda(alphan,car,Wr,mu);
    caf = caf*f_lamb_f;
    car = car*f_lamb_r;
end

% Making the transfer functions from del to r and B
c0 = caf + car;
c1 = a*caf - b*car;
c2 = (a^2)*caf + (b^2)*car;

% Creating coefficients of the transfer functions

```

```

A_r = (a*caf);
B_r = (a*caf*c0 - c1*caf)/(m*V);

A_B = (caf*Iz)/(m*V);
B_B = (caf*c2)/(m*V^2) - (a*caf*c1)/(m*V^2) - a*caf;

Meff = Iz;
Beff = (c0*Iz + m*c2)/(m*V);
Keff = (c0*c2)/(m*V^2) - c1 - (c1^2/(m*V^2));

denominator = [Meff, Beff, Keff];

rTF = [0, A_r, B_r; denominator];
BTF = [0, A_B, B_B; denominator];

% actually calculating rd, r, Vyd, Vy, and B

if use_tire_relaxation == 1
    Fyf = input_values.Fyf;
    Fyr = input_values.Fyr;

    tire_relaxation_length = car_params.tire_relaxation_length;
    Fyf_dot = (-alphaf*caf - Fyf)/(tire_relaxation_length/V);
    Fyf = Fyf + Fyf_dot*dt;
    Fyr_dot = (-alphar*car - Fyr)/(tire_relaxation_length/V);
    Fyr = Fyr + Fyr_dot*dt;

else
    Fyf = -caf*alphaf;
    Fyr = -car*alphar;
end

% rd => Derivative of r
% rdd => Derivative of rd

rdd = [];
rd = (a*Fyf - b*Fyr)/Iz;
% rdd = (A_r*deld + B_r*del - Beff*rd - Keff*r)/Meff;
% rd = rd + rdd*dt;
r = r + rd*dt;

% Vyd => Derivative of Vy

% Vyd = (-caf*((Vy + a*r)/Vx - del) - car*((Vy - b*r)/Vx) - m*V*r)/m;
Vyd = (Fyf + Fyr)/m - V * r;
Vy = Vy + Vyd*dt;

Bdd = [];
Bd = [];

```

```
% B = tan(Vy/Vx);
B = Vy/Vx;

% Bdd = (A_B*deld + B_B*del - Beff*Bd - Keff*B)/Meff;
% Bd = Bd + Bdd*dt;
% B = B + Bd*dt;

kus = Wf/(caf) - Wr/(car);    % rad/g
V_crit = sqrt(-L/kus);
end

Not enough input arguments.

Error in car_sim (line 5)
V = input_values.V;
```

Published with MATLAB® R2022a