

```

% Nettoyer l'environnement de travail
clc; % Effacer la fenêtre de commande
clear; % Supprimer toutes les variables de l'espace de travail
close all; % Fermer toutes les figures

% Définir l'intervalle de temps et les conditions initiales pour le problème de Cauchy
tspan = [0 10]; % Intervalle de temps de la simulation
y0 = [0 0 0 0]; % Conditions initiales : [vitesse de lacet, vitesse latérale, angle de lacet, position latérale]

% Résoudre le système d'équations différentielles ordinaires avec la fonction Model_ML
[t,y] = ode45(@Model_ML, tspan, y0); % Appel à la fonction ode45 pour résoudre le système d'équations différentielles

% Tracer les résultats
figure(1);
plot(t,y(:,1),'r') % Tracer la vitesse de lacet en fonction du temps
title('Vitesse de lacet(°)'); % Titre du graphique
xlabel('Temps (s)'); % Axe des abscisses
ylabel('Vitesse de lacet (°)'); % Axe des ordonnées
grid on

figure(2);
plot(t,y(:,2),'b') % Tracer la vitesse latérale en fonction du temps
title('Vitesse latérale(m/s)'); % Titre du graphique
xlabel('Temps (s)'); % Axe des abscisses
ylabel('Vitesse latérale (m/s)'); % Axe des ordonnées
grid on

figure(3);
plot(t,y(:,3),'g') % Tracer l'angle de lacet en fonction du temps
title('Psi (angle de lacet)'); % Titre du graphique
xlabel('Temps (s)'); % Axe des abscisses
ylabel('Angle de lacet (rad)'); % Axe des ordonnées
grid on

figure(4);
plot(t,y(:,4),'m') % Tracer la position latérale en fonction du temps
title('Yg (position latérale (m))') % Légende du graphique
xlabel('Temps (s)'); % Axe des abscisses
ylabel('Position latérale (m)'); % Axe des ordonnées
grid on

% Définition de la fonction Model_ML
function dydt = Model_ML(t, y)
    % y(1) = phi. = vitesse de lacet
    % y(2) = vy = vitesse latérale
    % y(3) = psi = angle de lacet
    % y(4) = Yg = position latérale

    % Définir les paramètres du véhicule
    Mt=1759; % Masse totale du véhicule

```

```

Mf=1319;    % Masse avant du véhicule
Mr=440;     % Masse arrière du véhicule
Iz=2638.5;  % Moment d'inertie
Lf=0.71;    % Distance entre le centre de gravité et l'essieu avant
Lr=2.13;    % Distance entre le centre de gravité et l'essieu arrière
cyf=94446;  % Coefficient de raideur latérale de l'essieu avant
cyr=48699;  % Coefficient de raideur latérale de l'essieu arrière

% Convertir la vitesse longitudinale de km/h en m/s
Vxt = 90/3.6;
Vx0 = Vxt;

% Calcul de thetav (angle de volant) en fonction du temps
f = 1/8;
thetav = 2.2*sin(2*pi*f*(t-0.25));
for i=1:length(t)
    if t(i)<0.25
        thetav(i) = 0;
    elseif t(i)>8.25
        thetav(i) = 0;
    end
end

% Calcul de Bvt (angle de braquage des roues avant) en fonction de thetav[1] ✓
[1]
lambda = 16;% Constante lambda
Bvt = thetav/lambda;

% Calculer les dérivées des variables d'état
dydt1 = 2*(Lf *cyf*Bvt)/Iz + 2*(-Lf*cyf + Lr*cyr)*y(2)/(Vx0*Iz) - 2*((Lf^2)*cyf ✓
+ (Lr^2)*cyr)*y(1)/(Vx0*Iz);
dydt2 = 2*cyf*Bvt/Mt - 2*(cyf + cyr)*y(2)/(Mt*Vx0) + (2*((-cyf*Lf + cyr*Lr)/ ✓
(Mt*Vx0))-Vx0)*y(1);
dydt3 = y(1);
dydt4 = Vx0*deg2rad(y(3)) + deg2rad (y(2));

% Retourner les dérivées dans un vecteur colonne
dydt = [dydt1 ;dydt2 ;dydt3 ;dydt4];
end

```