

```
% Technische Universität Berlin
% ILS Fachgebiet Kraftfahrzeuge
% Fahrzeugregelung II
% Projektarbeit
% Sommersemester 2019
% Andreas Hartmann, M. Sc.
```

```
% -----
% Fahrzeug
% Gruppe 10
% Yuheng Wu
% Jinxiao Yang
% Jingsheng Lyu
% -----
```

```
%%
clc;
clear all;
```

```
%% Gegebene Größen
```

```
vehicle_mass = 1500; % [kg] Gesamtmasse des Fahrzeugs
in.m_v = 750.8 / 1500 * vehicle_mass; % [kg] Achslast vorn
in.m_h = 749.2 / 1500 * vehicle_mass; % [kg] Achslast hinten
in.m = vehicle_mass; % [kg] Gesamtmasse des Fahrzeugs
in.T_z = 2081.2; % [kg*m] Trägheitsmoment um z-Achse
in.L = 2.72; % [m] Radstand
in.EG = 0.25 * pi / 180; % [Grad*s*m] Eigenlenkgradient
in.SG = 0.30 * pi / 180; % [Grad*s*m] Schwimmwinkelgradient
in.mu = 0.9; % [-] Reibungskoeffizient
in.C = 1.3; % [-] Formfaktor f0 Magic Formula
in.E = 0.85; % [-] Krümmungsfaktor f0 Magic Formula
in.g = 9.81; % [m/s^2] Erdbeschleunigung
in.rho = 1.2; % [kg/m^2] Dichte der Luft
in.c_w = 0.4; % [-] Luftwiderstandsbeiwert
in.A = 1.5; % [m^2] Stirnfläche des Fahrzeugs
```

```
%% Berechnungen
```

```
% Achslastverteilung aus Momentenbilanz
```

```
in.l_h = (in.m_v * in.L) / in.m; % [m] Achsabstand Schwerpunkt zu Hinterachse
in.l_v = (in.m_h * in.L) / in.m; % [m] Achsabstand Schwerpunkt zu Vorderachse
```

```
% Linearisierte Schräglaufsteifigkeit hinten - hergeleitet anhand der Definition des
% Schwimmwinkelgradienten SG
```

```
in.c_h = ( (in.m * in.l_v) / (in.SG * in.L) ); % [N/rad]
```

```
% Linearisierte Schräglaufsteifigkeit vorne - hergeleitet anhand der Definition des
% Eigenlenkgradienten EG
```

```
in.c_v = in.l_h / ( ((in.EG * in.L) / in.m) + in.l_v / in.c_h ); % [N/rad]
```

```
%% Berechnung der fehlenden Faktoren f0 die Magic Formula
```

```
% Amplitudenfaktor D (abhängig von Achslasten "vertical load")
```

```
in.D_v = in.mu * in.m_v * in.g; % [N]
in.D_h = in.mu * in.m_h * in.g; % [N]
```

```

% Steifigkeitsfaktor B
in.B_v = in.c_v / (in.C * in.D_v);    % [1/rad]
in.B_h = in.c_h / (in.C * in.D_h);    % [1/rad]

%%
% -----
% ManÖÖver
% -----

crgmainpath =
'/Users/eason/Downloads/Fahrzeugregelung/FaRe_II/Projekt/Projektdateien/OpenCRG.1.0.6'; % Hier
vollstÖÖndigen Pfad zu: ...\\OpenCRG.1.0.6\\matlab\\lib Verzeichnis angeben
addpath(genpath(crgmainpath));

n=2;% Stecke auswÖÖhlen
switch n
    case 1
        data = crg_read('Spurwechsel_ab30m.crg'); %Spurwechsel
        param.course='Spurwechsel_ab30m.crg';
    case 2
        data = crg_read('Slalom_3x18m.crg'); % Slalom
        param.course='Slalom_3x18m.crg';
end
data=rmfield(data,{'ct','struct'}); % Struct Elemente IÖÖschen damit embedded Matlab Function diese
verarbeiten kann
crg_show_refline_map(data);

%% Initial Simulinkmodell
K = [0,0,0,0];          % Regler
K10 = [0,0,0,0];        % Regler
k3_condi = 1;           % k3
delta_ff = 0;           % Vorsteuerung
steig = 0.03;           % Steigung

%% Aufgabe 1.2
delta_condi = 1;        % Ramp mit Steigung = 0.03
v_condi = 1;            % v = 20[m/s]
vorsteuer_condi = 1;    % Ohne Vorsteuerung

sim('Projekt_FARE2_SS19_template');
delta_a = in.L .* curvature_out ;    % Ackermannwinkel
EG = (diff(delta_out)./diff(ay_out) - diff(delta_a)./diff(ay_out));
EG = [EG;EG(end)];
EG_soll = in.EG * ones(size(ay_out,1),1);

figure
subplot(1,2,1);
plot(ay_out,delta_out,ay_out,delta_a);
title('Lenkradwinkel ueber a_y');
xlabel('Quereschleunigung a_y [m/s^2]');
ylabel('Lenkradwinkel [rad]');
legend('delta','delta_a');

```

```
grid on;
% print(gcf,'-djpeg','figure1.jpg')
```

```
subplot(1,2,2);
plot(ay_out,EG,ay_out,EG_soll)
title('Eigengradient');
xlabel('Querschleunigung a_y [m/s^2]');
ylabel('EG [rad*s^2/m]');
legend('EG','EG_soll');
grid on;
print(gcf,'-djpeg','figure1.jpg')
```

```
%% Aufgabe 2.1
```

```
for vx = 5:20
```

```
    A = [0,1,0,0;
        0,-(in.c_v+in.c_h)/in.m/vx, (in.c_v+in.c_h)/in.m, -(in.c_v*in.l_v-in.c_h*in.l_h)/in.m/vx;
        0,0,0,1;
        0,-(in.c_v*in.l_v-in.c_h*in.l_h)/in.T_z/vx, (in.c_v*in.l_v-in.c_h*in.l_h)/in.T_z, -
        (in.c_v*in.l_v*in.l_v+in.c_h*in.l_h*in.l_h)/in.T_z/vx];
```

```
    EW(:,vx-4) = eig(A);
```

```
end
```

```
% Analysierung der Stabilitaets
```

```
instabilSys = EW > 0;
instabilGeschwindigkeit = find (sum(instabilSys,1) > 0) + 4;
fprintf('Bei der Geschwindigkeit v = %d [m/s] ist das System nicht stabil,\n da die Eigenwerte > 0\n sind.\n',instabilGeschwindigkeit);
```

```
%% Aufgabe 2.2
```

```
delta_condi = 2;           % Durch Konstanter Regler
v_condi = 1;               % v = 20[m/s]
vorsteuer_condi = 1;       % Ohne Vorsteuerung
```

```
B1 = [0; in.c_v/in.m; 0; in.c_v * in.l_v / in.T_z];
B2 = [0; -vx-(in.c_v*in.l_v-in.c_h*in.l_h)/in.m/vx; 0; -(in.c_v*in.l_v*in.l_v+in.c_h*in.l_h*in.l_h)/in.T_z/vx];
```

```
P = [-10,-6,-4+1i,-4-1i]; % Polezuweisung  $\hat{O} \hat{\Omega} \hat{O} \hat{\Omega} - \beta \hat{O} \hat{\Omega} \hat{O} \hat{\Omega} \hat{O} \hat{\Omega} \hat{O} \hat{\Omega} \hat{O} \hat{\Omega} \sqrt{\lambda \hat{O} \hat{\Omega} \hat{\Omega} \hat{O} \hat{\Omega}}$ 
```

```
K = place(A,B1,P);
delta_max = 40;           % [Grad]
delta_max = delta_max * ones(size(tout,1),1);
```

```
% Spurwechsel
```

```
n = 1;
switch n
case 1
    data = crg_read('Spurwechsel_ab30m.crg'); %Spurwechsel
    param.course='Spurwechsel_ab30m.crg';
```

```

case 2
    data = crg_read('Slalom_3x18m.crg'); % Slalom
    param.course='Slalom_3x18m.crg';
end
data=rmfield(data,{'ct','struct'}); % Struct Elemente löschen damit embedded Matlab Function diese
verarbeiten kann

```

```

sim('Projekt_FARE2_SS19_template');
delta_out = delta_out/pi*180; % [Grad]
Spurwechsel.Pos_x = data.rx;
Spurwechsel.Pos_y = data.ry;

```

```

figure
plot(tout,delta_out,tout,delta_max);
title('Lenkradwinkel beim Spurwechsel');
xlabel('Zeit [s]');
ylabel('Lenkradwinkel [Grad]');
legend('delta','delta_max');
grid on;
print(gcf,'-djpeg','figure2.jpg')

```

```

figure
plot(x_out,y_out,Spurwechsel.Pos_x,Spurwechsel.Pos_y);
title('Spurwechsel');
xlabel('X [m]');
ylabel('Y [m]');
legend('Modell','Referenz');
grid on;
print(gcf,'-djpeg','figure3.jpg')

```

```

% Slalom
n = 2;
switch n
case 1
    data = crg_read('Spurwechsel_ab30m.crg'); %Spurwechsel
    param.course='Spurwechsel_ab30m.crg';
case 2
    data = crg_read('Slalom_3x18m.crg'); % Slalom
    param.course='Slalom_3x18m.crg';
end
data=rmfield(data,{'ct','struct'}); % Struct Elemente löschen damit embedded Matlab Function diese
verarbeiten kann

```

```

sim('Projekt_FARE2_SS19_template');
delta_out = delta_out/pi*180; % [Grad]
Slalom.Pos_x = data.rx;
Slalom.Pos_y = data.ry;

```

```

figure
plot(tout,delta_out,tout,delta_max);
title('Lenkradwinkel beim Slalom');
xlabel('Zeit [s]');
ylabel('Lenkradwinkel [Grad]');
legend('delta','delta_max');

```

```

grid on;
print(gcf,'-djpeg','figure4.jpg')

figure
plot(x_out,y_out,Slalom.Pos_x,Slalom.Pos_y);
title('Slalom');
xlabel('X [m]');
ylabel('Y [m]');
legend('Modell','Referenz');
grid on;
print(gcf,'-djpeg','figure5.jpg')

%% Aufgabe 2.3
% ohne Vorsteuerung
vx = 20;
R = 100;
psi_p_soll = vx / R;
X_ss = -inv(A - B1*K)*B2*psi_p_soll;
e1_ss = X_ss(1);
fprintf('Stationaer Fehler der Querabweichung(e1,ss) bei v = %d und r = %d ist %.3f. \n',vx,R,e1_ss);

% Simulation
delta_condi = 2;           % Durch Konstanter Regler
v_condi = 1;               % v = 20[m/s]
vorsteuer_condi = 1;       % Ohne Vorsteuerung

n = 1;                     % Spurwechsel
switch n
    case 1
        data = crg_read('Spurwechsel_ab30m.crg'); %Spurwechsel
        param.course='Spurwechsel_ab30m.crg';
    case 2
        data = crg_read('Slalom_3x18m.crg'); % Slalom
        param.course='Slalom_3x18m.crg';
end
data=rmfield(data,{'ct','struct'}); % Struct Elemente löschen damit embedded Matlab Function diese
verarbeiten kann

sim('Projekt_FARE2_SS19_template');

ohneVorsteuerungSpurwechsel.delta_out = delta_out/pi*180;
ohneVorsteuerungSpurwechsel.x_out = x_out;
ohneVorsteuerungSpurwechsel.y_out = y_out;

n = 2;                     % Slalom
switch n
    case 1
        data = crg_read('Spurwechsel_ab30m.crg'); %Spurwechsel
        param.course='Spurwechsel_ab30m.crg';
    case 2
        data = crg_read('Slalom_3x18m.crg'); % Slalom
        param.course='Slalom_3x18m.crg';
end
data=rmfield(data,{'ct','struct'}); % Struct Elemente löschen damit embedded Matlab Function diese
verarbeiten kann

```

```
sim('Projekt_FARE2_SS19_template');
```

```
ohneVorsteuerungSlalom.delta_out = delta_out/pi*180;  
ohneVorsteuerungSlalom.x_out = x_out;  
ohneVorsteuerungSlalom.y_out = y_out;
```

```
% mit Vorsteuerung
```

```
delta_ff = in.m*vx*vx/R/in.L*(in.l_h/in.c_v - in.l_v/in.c_h + in.l_v/in.c_h * K(3)) + (in.L-in.l_h*K(3))/R;  
X_ss2 = -inv(A - B1*K)*(B1*delta_ff + B2*psi_p_soll);  
e1_ss2 = X_ss2(1);  
fprintf('Stationaer Fehler der Querabweichung(e1,ss) mit Vorsteuerung bei v = %d und r = %d ist %.3f.\n',vx,R,e1_ss2);
```

```
% Simulation
```

```
delta_condi = 2;           % Durch Konstanter Regler  
v_condi = 1;               % v = 20[m/s]  
k3_condi = 1;              % Kostante Vorsteuerung  
vorsteuer_condi = 0;       % Kostante Vorsteuerung
```

```
n = 1;                     % Spurwechsel
```

```
switch n
```

```
case 1
```

```
data = crg_read('Spurwechsel_ab30m.crg'); %Spurwechsel  
param.course='Spurwechsel_ab30m.crg';
```

```
case 2
```

```
data = crg_read('Slalom_3x18m.crg'); % Slalom  
param.course='Slalom_3x18m.crg';
```

```
end
```

```
data=rmfield(data,{'ct','struct'}); % Struct Elemente löschen damit embedded Matlab Function diese verarbeiten kann
```

```
sim('Projekt_FARE2_SS19_template');
```

```
mitVorsteuerungSpurwechsel.delta_out = delta_out/pi*180;  
mitVorsteuerungSpurwechsel.x_out = x_out;  
mitVorsteuerungSpurwechsel.y_out = y_out;
```

```
n = 2;                     % Slalom
```

```
switch n
```

```
case 1
```

```
data = crg_read('Spurwechsel_ab30m.crg'); %Spurwechsel  
param.course='Spurwechsel_ab30m.crg';
```

```
case 2
```

```
data = crg_read('Slalom_3x18m.crg'); % Slalom  
param.course='Slalom_3x18m.crg';
```

```
end
```

```
data=rmfield(data,{'ct','struct'}); % Struct Elemente löschen damit embedded Matlab Function diese verarbeiten kann
```

```
sim('Projekt_FARE2_SS19_template');
```

```
mitVorsteuerungSlalom.delta_out = delta_out/pi*180;  
mitVorsteuerungSlalom.x_out = x_out;
```

```
mitVorsteuerungSlalom.y_out = y_out;
```

```
figure
plot(ohneVorsteuerungSpurwechsel.x_out,ohneVorsteuerungSpurwechsel.y_out,mitVorsteuerungSpurwechsel.x_out,mitVorsteuerungSpurwechsel.y_out,Spurwechsel.Pos_x,Spurwechsel.Pos_y);
title('Spurwechsel');
xlabel('X [m]');
ylabel('Y [m]');
legend('Ohne Vorsteuerung','Mit Vorsteuerung','Referenz');
grid on;
print(gcf,'-djpeg','figure6.jpg')
```

```
figure
plot(ohneVorsteuerungSlalom.x_out,ohneVorsteuerungSlalom.y_out,mitVorsteuerungSlalom.x_out,mitVorsteuerungSlalom.y_out,Slalom.Pos_x,Slalom.Pos_y);
title('Slalom');
xlabel('X [m]');
ylabel('Y [m]');
legend('Ohne Vorsteuerung','Mit Vorsteuerung','Referenz');
grid on;
print(gcf,'-djpeg','figure7.jpg')
```

```
figure
plot(tout,ohneVorsteuerungSpurwechsel.delta_out,tout,mitVorsteuerungSpurwechsel.delta_out,tout,delta_max);
title('Lenkradwinkel beim Spurwechsel');
xlabel('Zeit [s]');
ylabel('Lenkradwinkel [Grad]');
legend('Ohne Vorsteuerung','Mit Vorsteuerung','delta_max');
grid on;
print(gcf,'-djpeg','figure8.jpg')
```

```
figure
plot(tout,ohneVorsteuerungSlalom.delta_out,tout,mitVorsteuerungSlalom.delta_out,tout,delta_max);
title('Lenkradwinkel beim Slalom');
xlabel('Zeit [s]');
ylabel('Lenkradwinkel [Grad]');
legend('Ohne Vorsteuerung','Mit Vorsteuerung','delta_max');
grid on;
print(gcf,'-djpeg','figure9.jpg')
```

```
%% Aufgabe 2.4
```

```
% Berechnung K10
```

```
vx = 10;
```

```
A10 = [0,1,0,0;
        0,-(in.c_v+in.c_h)/in.m/vx, (in.c_v+in.c_h)/in.m, -(in.c_v*in.l_v-in.c_h*in.l_h)/in.m/vx;
        0,0,0,1;
        0,-(in.c_v*in.l_v-in.c_h*in.l_h)/in.T_z/vx, (in.c_v*in.l_v-in.c_h*in.l_h)/in.T_z, -
        (in.c_v*in.l_v*in.l_v+in.c_h*in.l_h*in.l_h)/in.T_z/vx];
```

```
K10 = place(A10,B1,P);
```

```
% Simulation
```

```

n = 2; % Slalom
switch n
    case 1
        data = crg_read('Spurwechsel_ab30m.crg'); %Spurwechsel
        param.course='Spurwechsel_ab30m.crg';
    case 2
        data = crg_read('Slalom_3x18m.crg'); % Slalom
        param.course='Slalom_3x18m.crg';
end
data=rmfield(data,{'ct','struct'}); % Struct Elemente löschen damit embedded Matlab Function diese
verarbeiten kann

% Mit Angepasster Regler
delta_condi = 3; % Durch angepassten Regler
v_condi = 0; % Erhöhende Geschwindigkeit
k3_condi = 2; % Angepasste Vorsteuerung
vorsteuer_condi = 0; % Angepasste Vorsteuerung

sim('Projekt_FARE2_SS19_template');

mitAngepassterRegler.delta_out = delta_out/pi*180;
mitAngepassterRegler.x_out = x_out;
mitAngepassterRegler.y_out = y_out;

% Mit Konstanter Regler
delta_condi = 2; % Durch Konstanter Regler
v_condi = 0; % Erhöhende Geschwindigkeit
k3_condi = 1; % Konstante Vorsteuerung
vorsteuer_condi = 0; % Konstante Vorsteuerung

sim('Projekt_FARE2_SS19_template');

mitKonstanterRegler.delta_out = delta_out/pi*180;
mitKonstanterRegler.x_out = x_out;
mitKonstanterRegler.y_out = y_out;

figure
plot(mitAngepassterRegler.x_out,mitAngepassterRegler.y_out,mitKonstanterRegler.x_out,mitKonstanterR
egler.y_out,Slalom.Pos_x,Slalom.Pos_y);
title('Slalom');
xlabel('X [m]');
ylabel('Y [m]');
legend('Angepas. Regler','Konst. Regler','Referenz');
grid on;
print(gcf,'-djpeg','figure10.jpg')

figure
plot(tout,mitAngepassterRegler.delta_out,tout,mitKonstanterRegler.delta_out,tout,delta_max);
title('Lenkradwinkel beim Slalom');
xlabel('Zeit [s]');
ylabel('Lenkradwinkel [Grad]');
legend('Angepas. Regler','Konst. Regler','delta_max');
grid on;
print(gcf,'-djpeg','figure11.jpg')

```



