%% Date: 4/7/2016

%% Author: EE 3413 -- Analysis & Design of Control Systems

%% In this MATLAB script, we'll see how we can derive a CLTF for P,PI,PID

%% controllers given one example of a third order system modeled by G(s).

%% We'll also see how we can extract the output y\_step(t) for different

%% controllers and how to obtain the step reponse of a system.

clear all; % clearing all variables

clc; % clearing the command prompt

syms s; % variable s

syms K; % variable K

G=(1)/(s\*(s+5)\*(s+6)); % defining a generic G(s) which will be used for

% the three controller designs

%% Next, we simulate a generic P controller (only proportional term)

G\_P = K;

CLTF = (G\*G\_P)/(1+G\*G\_P); % constructing the closed-loop transfer function

CLTF=factor(CLTF);

disp('The CLTF given a P controller is')

pretty(CLTF)

K=100; % you can select the K you want here

num=[K]; % this has been extracted after looking at pretty(CLTF)

den=[1 11 30 K]; % this has been extracted after looking at pretty(CLTF)

figure

step(tf(num,den))

title('Step Response after the Design of a P Controller')

[y\_step\_P,t]=step(tf(num,den)); % This command generates the output (y\_step(t))

% and the time-signal (t) used to plot the step response

% Notice that for K=330, you get sustained oscillations

% How can we plot the tracking error (SSE) function e(t)?

% Recall that e(t)=y(t)-u(t) (or u(t)-y(t)). The input in this case is the

% step function, which is u(t)=1 for all t. Hence:

u=ones(length(t),1);

% This command defines the input u(t) as ones for the length of the time-vector

error\_P=y\_step\_P-u;

figure

plot(t,error\_P);

xlabel('Time (sec)')

ylabel('e(t)')

title('Tracking Error Given a P Controller')

%% Next, we simulate a generic PI controller (only proportional term)

clear K; % clearing K

syms K;

syms s;

syms Ti % defining the integrator constant

G\_PI = K\*(1+(1/(Ti\*s))); % defining a PI controller

CLTF\_PI = (G\*G\_PI)/(1+G\*G\_PI); % constructing the closed-loop transfer function

CLTF\_PI=factor(CLTF\_PI);

disp('The CLTF given a PI controller is')

pretty(CLTF\_PI)

K=10; % you can select the K you want here

Ti=3; % you can select the Ti you want here

num=[K\*Ti K]; % this has been extracted after looking at pretty(CLTF)

den=[Ti^2 11\*Ti 30\*Ti K\*Ti K]; % this has been extracted after looking at pretty(CLTF)

figure

step(tf(num,den))

xlabel('Time (sec)')

ylabel('e(t)')

title('Step Response after the Design of a PI Controller')

[y\_step\_PI,t]=step(tf(num,den)); % This command generates the output (y\_step(t))

% and the time-signal (t) used to plot the step response

% Plotting tracking error for PI controller

u=ones(length(t),1);

% This command defines the input u(t) as ones for the length of the time-vector

error\_PI=y\_step\_PI-u;

figure

plot(t,error\_PI);

title('Tracking Error Given a PI Controller')

%% Next, we simulate a generic PID controller (only proportional term)

clear K Ti; % clearing K, Ti

syms K;

syms s;

syms Ti Td

G\_PID = K\*(1+Td\*s+(1/(Ti\*s))); % defining a PI controller

CLTF\_PID = (G\*G\_PID)/(1+G\*G\_PID); % constructing the closed-loop transfer function

CLTF\_PID=factor(CLTF\_PID);

disp('The CLTF given a PID controller is')

pretty(CLTF\_PID)

K=10; % you can select the K you want here

Ti=3; % you can select the Ti you want here

Td=1; % you can select the Ti you want here

num=[K\*Ti\*Td K\*Ti K]; % this has been extracted after looking at pretty(CLTF)

den=[Ti 11\*Ti 30\*Ti+K\*Td\*Ti K\*Ti K]; % this has been extracted after looking at pretty(CLTF)

figure

step(tf(num,den))

title('Step Response after the Design of a PID Controller')

[y\_step\_PID,t]=step(tf(num,den)); % This command generates the output (y\_step(t))

% and the time-signal (t) used to plot the step response

% Plotting tracking error for PID controller

u=ones(length(t),1);

% This command defines the input u(t) as ones for the length of the time-vector

error\_PID=y\_step\_PID-u;

figure

plot(t,error\_PID);

xlabel('Time (sec)')

ylabel('e(t)')

title('Tracking Error Given a PID Controller')