```
%% Name: Mohammed Al-Sayegh, ECE 414 Homework - PID Tuner
%% Part 1, P control
clc
close all
% Using pidtuner to generate a PI controller
C p = pidtune(G, 'P');
% Pass the default terms of Kp and Ki baseline tune of PI by pidTuner
Tune = pidTuner(G, C p);
waitfor(Tune);
% Display base terms that given by pidTuner
disp('Parameters from pidTuner');
disp('Kp = ');
disp(C p.Kp);
disp('The step info of P controller using stepinfo:');
disp(stepinfo(C p));
disp('The step info of P controller using getallspecs:');
disp(getallspecs(G,C p));
% Baseline controller effort and system transfer functions
% Baseline System Transfer Function
T p = (C p*G)/(1+(G*C p));
%% Plot the Baseline vs. tuned system Step Response
figure(1);
hold on;
step(T p)
grid on;
legend('Baseline');
title('System Step Response of PI Contoller');
hold off;
%% Plot the Baseline vs. tuned system Contoller Effort Step Response
figure(2)
hold on;
Tu = T p/G;
step(Tu)
grid on;
legend('Baseline');
title('Contoller Effort Step Response of P');
hold off;
%% Part 2, PD control
```

```
clc
close all
% Using pidtuner to generate a PD controller
C pd = pidtune(G, 'PD');
% Pass the default terms of Kp and Ki baseline tune of PI by pidTuner
Tune = pidTuner(G, C pd);
waitfor(Tune);
% Display base terms that given by pidTuner
disp('Parameters from pidTuner');
disp('Kp = ');
disp(C_pd.Kp);
disp('Kd = ');
disp(C pd.Kd);
disp('Cannot simulate the time response of improper (non-causal) PD contorller\n');
disp('The step info of PD controller using getallspecs:');
disp(getallspecs(G,C pd));
% Baseline controller effort and system transfer functions
% Baseline System Transfer Function
T pd = (C pd*G) / (1+(G*C pd));
%% Plot the Baseline vs. tuned system Step Response
figure(1);
hold on;
step(T pd)
grid on;
legend('Baseline');
title('System Step Response of PD Contoller');
hold off;
%% Plot the Baseline vs. tuned system Contoller Effort Step Response
figure(2)
hold on;
Tu = T pd/G;
% step(Tu)
grid on;
legend('Baseline', 'Tuned');
title('Contoller Effort Step Response of PD');
hold off;
%% Part 3, PI control
clc
close all
```

```
% Using pidtuner to generate a PI controller
C pi = pidtune(G, 'PI');
% Pass the default terms of Kp and Ki baseline tune of PI by pidTuner
Tune = pidTuner(G, C pi);
waitfor(Tune);
% Display base terms that given by pidTuner
disp('Parameters from pidTuner');
disp('Kp = ');
disp(C pi.Kp);
disp('Ki = ');
disp(C_pi.Ki);
disp('The step info of PI controller using stepinfo:');
disp(stepinfo(C pi));
disp('The step info of PI controller using getallspecs:');
disp(getallspecs(G,C pi));
% Baseline controller effort and system transfer functions
% Baseline System Transfer Function
T pi = (C pi*G)/(1+(G*C pi));
%% Plot the Baseline vs. tuned system Step Response
figure(1);
hold on;
subplot(1,2,1);
step(T pi)
grid on;
legend('Baseline');
title('System Step Response of PI Contoller');
% Plot the Baseline vs. tuned system Contoller Effort Step Response
hold on;
subplot(1,2,2)
Tu = T pi/G;
step(Tu)
grid on;
legend('Baseline');
title('Contoller Effort Step Response of PI');
hold off;
%% Part 4, PID control
pause (1);
close all;
%% Using pidtuner to generate a PID controller
C pid = pidtune(G, 'PID');
```

```
\mbox{\%} Pass the default terms of Kp and Ki baseline tune of PI by pidTuner
Tune = pidTuner(G, C pid);
waitfor(Tune);
% Display base terms that given by pidTuner
disp('Parameters from pidTune');
disp('Kp = ');
disp(C pid.Kp);
disp('Ki = ');
disp(C_pid.Ki);
disp('Kd = ');
disp(C pid.Kd);
%disp('The step info of PID controller using stepinfo:');
%disp(stepinfo(C pid));
disp('The step info of PID controller using getallspecs:');
disp(getallspecs(G,C pid));
% Baseline controller effort and system transfer functions
% Baseline System Transfer Function
T pid = (C pid*G)/(1+(G*C pid));
%% Plot the Baseline vs. tuned system Step Response
figure(1);
hold on;
step(T pid)
grid on;
legend('Baseline');
title('System Step Response of PID Contoller');
hold off;
%% Plot the Baseline vs. tuned system Contoller Effort Step Response
figure(2)
hold on;
Tu = T pid/G;
%step(Tu)
grid on;
legend('Baseline');
title ('Contoller Effort Step Response of PID');
hold off;
%% Part 5, PIDF control
clc;
pause (1);
close all;
```

```
%% Using pidtuner to generate a PIDF controller
C pidf = pidtune(G, 'PIDF');
% Pass the default terms of Kp, Ki and Kd baseline tune of PID by pidTuner
Tune = pidTuner(G, C pidf);
waitfor(Tune);
% Display base terms that given by pidTuner
disp('Parameters from pidTune');
disp('Kp = ');
disp(C pidf.Kp);
disp('Ki = ');
disp(C pidf.Ki);
disp('Kd = ');
disp(C pidf.Kd);
disp('The step info of PIDF controller using stepinfo:');
disp(stepinfo(C pidf));
disp('The step info of PIDF controller using getallspecs:');
disp(getallspecs(G,C pidf));
% Baseline controller effort and system transfer functions
% Baseline System Transfer Function
T pidf = (C \text{ pidf*G})/(1+(G*C \text{ pidf}));
%% Plot the Baseline vs. tuned system Step Response
figure(1);
hold on;
subplot(1,2,1)
step(T pidf)
grid on;
legend('Baseline');
title('System Step Response of PIDF Contoller');
% Plot the Baseline vs. tuned system Contoller Effort Step Response
subplot(1,2,2)
Tu = T pidf/G;
step(Tu)
grid on;
legend('Baseline');
title ('Contoller Effort Step Response of PIDF');
hold off;
%% Part 6, PDF control
clc;
pause (1);
close all;
```

```
%% Using pidtuner to generate a PDF controller
C pdf = pidtune(G, 'PDF');
% Pass the default terms of Kp, Ki and Kd baseline tune of PID by pidTuner
%Tune = pidTuner(G, C pdf);
%waitfor(Tune);
% Display base terms that given by pidTuner
disp('Parameters from pidTune');
disp('Kp = ');
disp(C pdf.Kp);
disp('Ki = ');
disp(C pdf.Ki);
disp('Kd = ');
disp(C pdf.Kd);
disp('The step info of PIDF controller using stepinfo:');
disp(stepinfo(C pdf));
disp('The step info of PIDF controller using getallspecs:');
disp(getallspecs(G,C pdf));
% Baseline controller effort and system transfer functions
% Baseline System Transfer Function
T pdf = (C pdf*G)/(1+(G*C pdf));
%% Plot the Baseline vs. tuned system Step Response
figure(1);
hold on;
subplot(1,2,1)
step(T pdf)
grid on;
legend('Baseline');
title('System Step Response of PDF Contoller');
hold off;
%% Plot the Baseline vs. tuned system Contoller Effort Step Response
figure(2)
hold on;
subplot(1,2,2)
Tu = T pdf/G;
step(Tu)
grid on;
legend('Baseline');
title('Contoller Effort Step Response of PDF');
hold off;
```