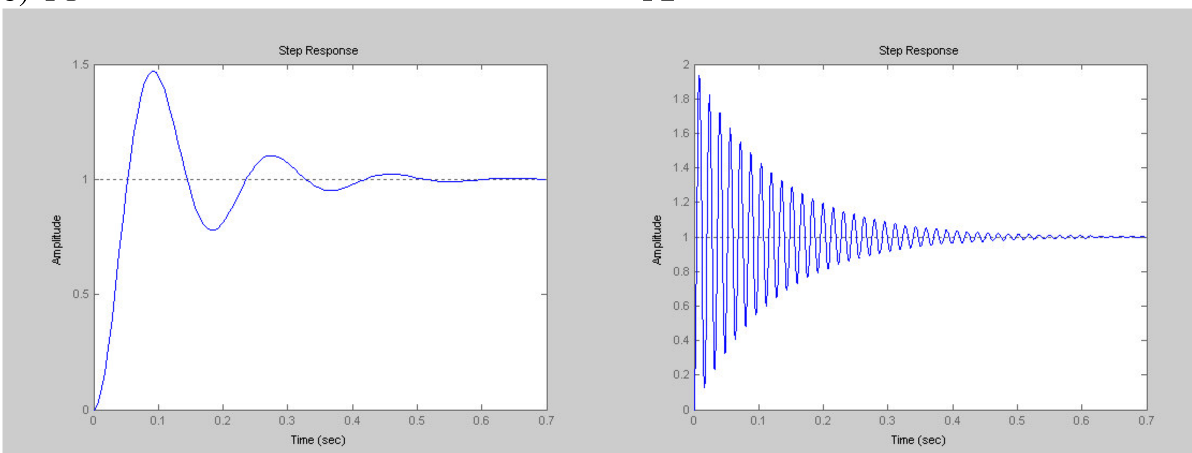
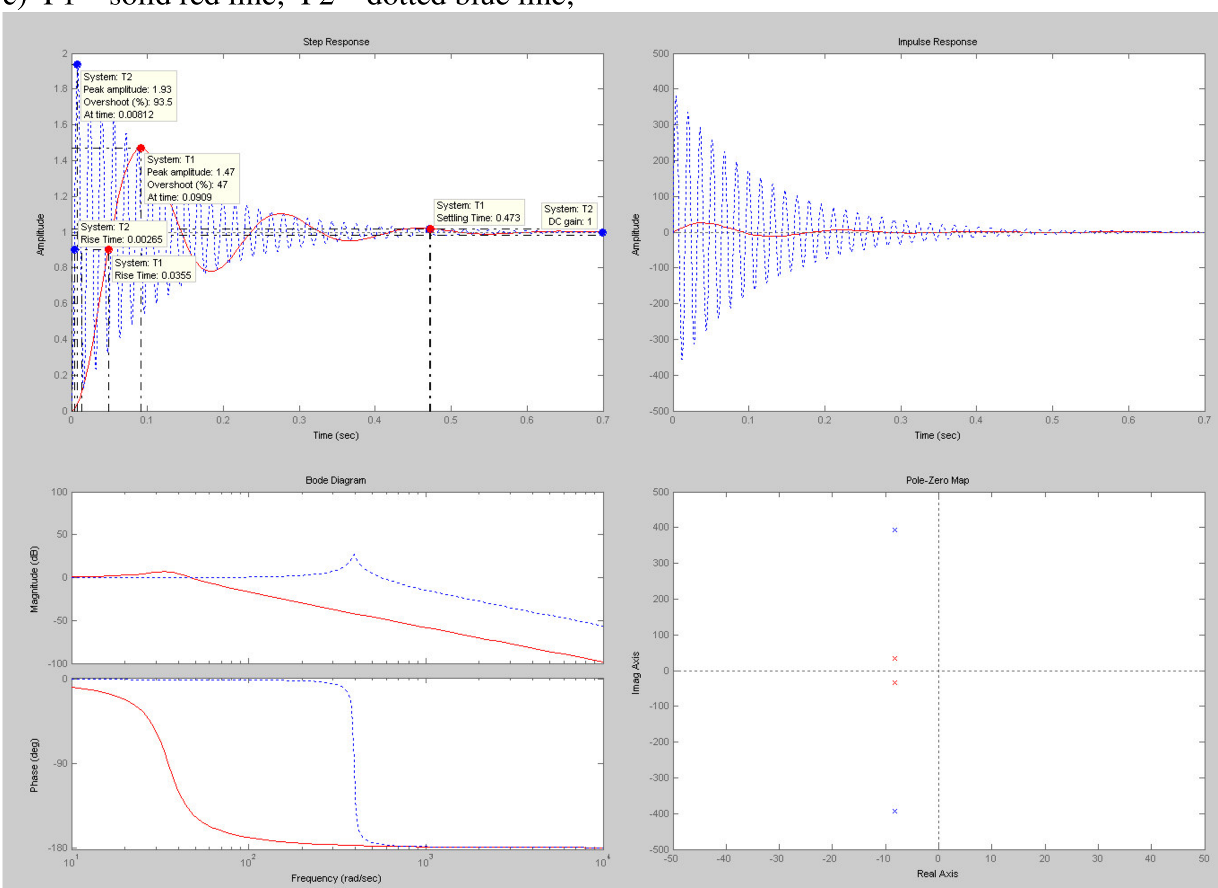
Lab 7 Solutions

a) see Matlab code appended at end

b) F1 F2



c) F1 – solid red line; F2 – dotted blue line;



d) i) open-loop poles for K * F1

K=100	K=10	K=1
-8.2353 +34.2715i	-8.2353 +34.2715i	-8.2353 +34.2715i
-8.2353 -34.2715i	-8.2353 -34.2715i	-8.2353 -34.2715i

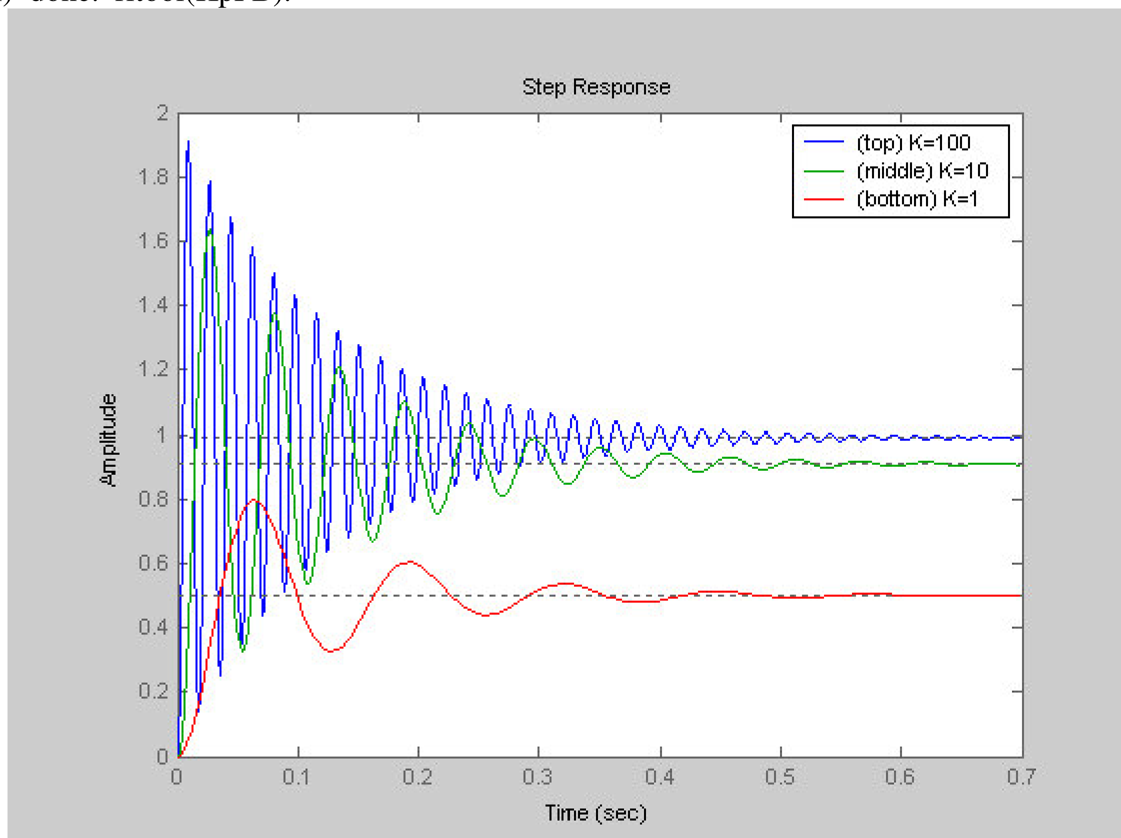
ii)

$$\frac{1056}{0.85\,s^2 + 14\,s + 2112} \quad (K=1)$$

closed-loop poles for K * F1

K=100	K=10	K=1
1.0e+002 *		
-0.0824 + 3.5413i	-0.0824 + 1.1661i	-0.0824 + 0.4916i
-0.0824 - 3.5413i	-0.0824 - 1.1661i	-0.0824 - 0.4916i

iii) done. rltool(KpFB).



iv) As the proportional gain (Kp) in the feedback system is increased, frequency of oscillations increases dramatically (due to decreased damping from the imaginary part of the poles being greater), %overshoot increases slowly, settling time remains unchanged (because the real parts of the poles are not moving), rise time and peak time decrease, and steady-state error approaches zero as Kp approaches infinity.

e) i)

$$\frac{1056\, s + 1056}{0.85\, s^2 + 1070\, s + 2112} \qquad (K=1)$$

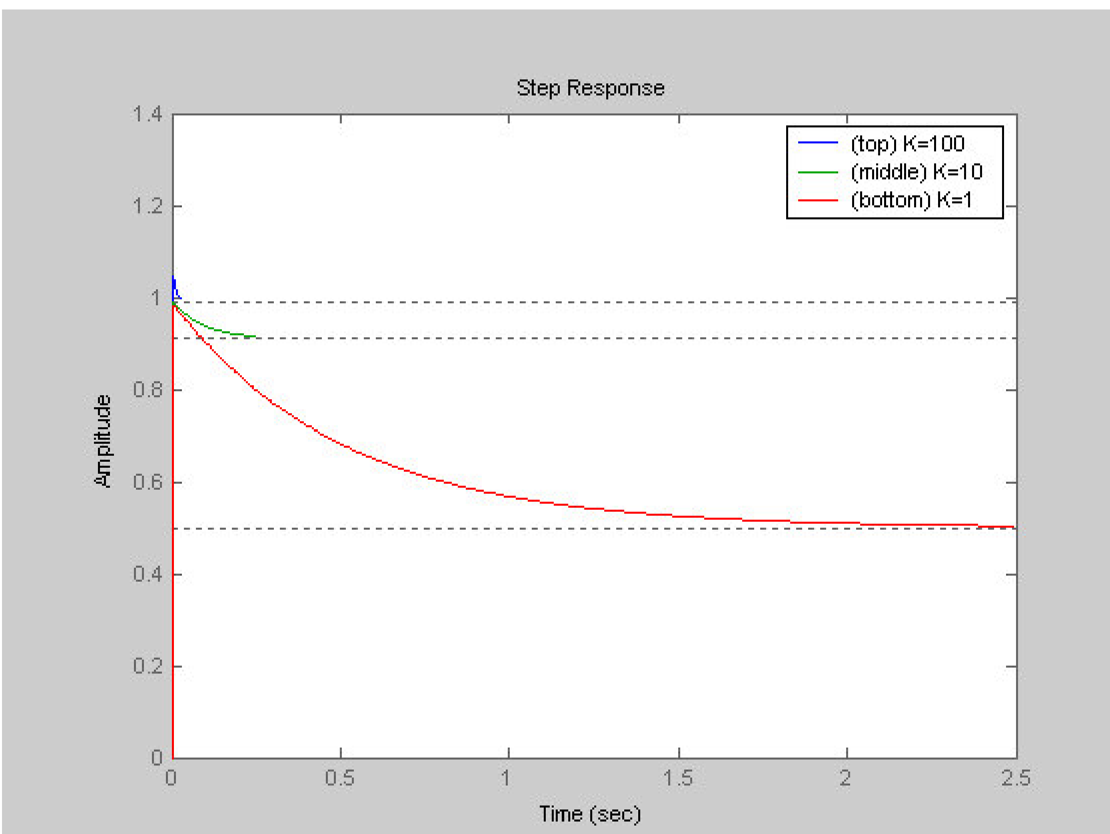
ii) closed-loop poles for (s + K) * F1

K=100	K=10	K=1
1.0e+003 *		
-1.1497	-1.2479	-1.2568
-0.1091	-0.0110	-0.0020

closed-loop zeros for (s + K) * F1

K=100	K=10	K=1
-100	-10	-1

iii) rltool(KdFB)



iv) adding a zero decreases the steady-state error, increases settling time, decreases the rise and peak times dramatically, and eliminates oscillations. increasing K further improves the steady state error and decreases %overshoot of the system

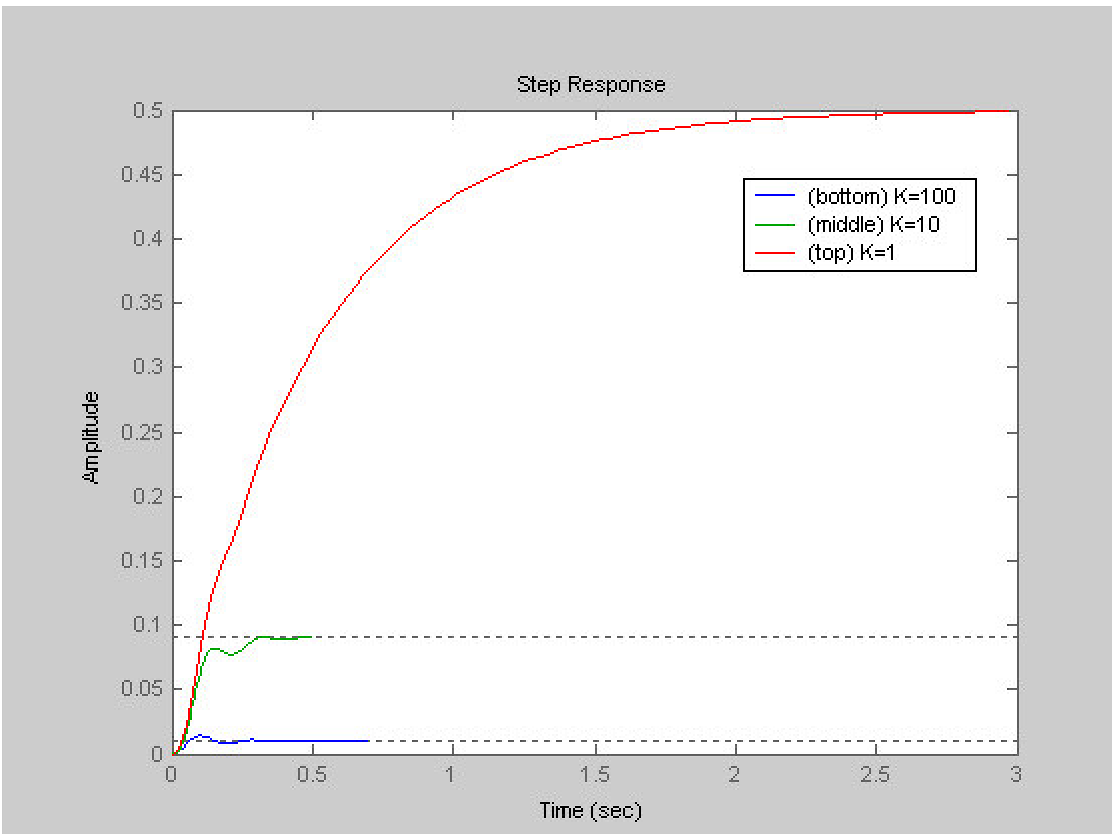
f) i)

$$\frac{1056}{0.85 s^3 + 14.85 s^2 + 1070 s + 2112} \qquad (K=1)$$

ii) closed-loop poles for 1/(s + K) * F1

K=100	K=10	K=1
1.0e+002 *		
-1.0013	-0.0771 + 0.3431i	-0.0772 + 0.3417i
-0.0817 + 0.3444i	-0.0771 - 0.3431i	-0.0772 - 0.3417i
-0.0817 - 0.3444i	-0.1105	-0.0202

there are no zeros for this system.



adding a pole increases the rise and peak times, and settling time, and provides damping to eliminate oscillations. further increasing K is detrimental, however, as it increases oscillations, %overshoot, and steady state error, but decreases settling time.

```
%------ Matlab code

% a)  define transfer functions using TF();

F1 = TF([1056],[.85 14 1056]);
F2 = TF([132000],[.85 14 132000]);

% b)  visuallize the step response using STEP();
figure(1)
STEP(F1)
figure(2)
STEP(F2)

% c)  play with these in LTI Viewer.

LTIView({'step';'impulse';'bode';'pzmap'},F1,'r-',F2,'b:')

% d)  proportional gain

kk = [100 10 1];

for i=1:length(kk)
    Kp = TF([kk(i)],[1]);
    KpF = series(Kp,F1);
    ppo(:,i)=pole(KpF);
    zpo(:,i)=zero(KpF);

    KpFB = feedback(KpF,1);
    ppc(:,i)=pole(KpFB);
    zpc(:,i)=zero(KpFB);

    figure(3)
    STEP(KpFB);
    hold on;

    Kd = TF([1 kk(i)],[1]);
    KdF = series(Kd,F1);
    KdFB = feedback(KdF,1)
    pdc(:,i)=pole(KdFB);
    zdc(:,i)=zero(KdFB);

    figure(4)
    STEP(KdFB);
    hold on;

    Ki = TF([1],[1 kk(i)]);
```

```
    KiF = series(Ki,F1);
    KiFB = feedback(KiF,1);
    pic(:,i)=pole(KiFB);
    zic(:,i)=zero(KiFB);

    figure(5)
    STEP(KiFB);
    hold on;
end

'open-loop poles for K * F1'
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')
disp(ppo);

'closed-loop poles for K * F1'
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')
disp(ppc);

'closed-loop poles for (s + K) * F1'
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')
disp(pdc);

'closed-loop zeros for (s + K) * F1'
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')
disp(zdc);

'closed-loop poles for 1/(s + K) * F1'
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')
disp(pic);

'closed-loop zeros for 1/(s + K) * F1'
fprintf('\tK=100\t\t\t\tK=10\t\t\t\tK=1\n')
disp(zic);
```