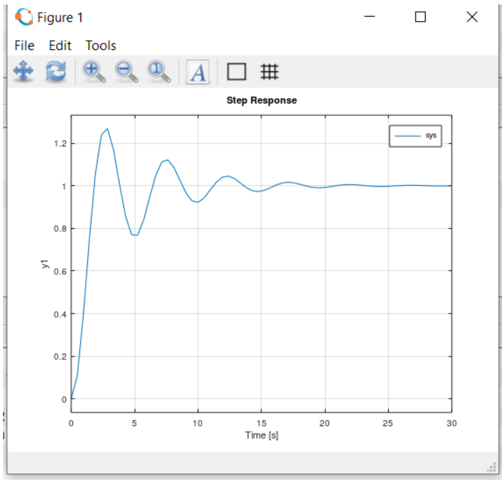
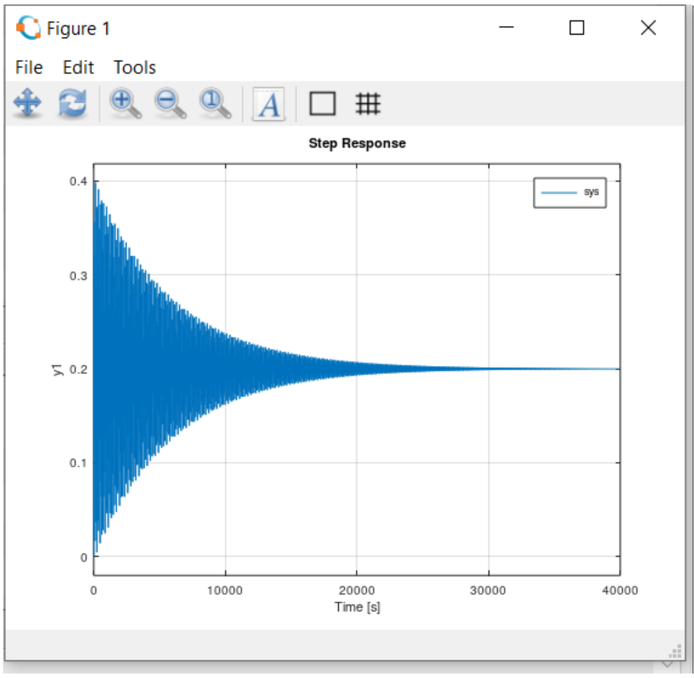
Douraïd BEN HASSEN

```
>> pkg load control
>> num1 = [1 1];
>> den1 = [1 1 2 1];
>> sys = tf(num1, den1)

Transfer function 'sys' from input 'u1' to output ...

            s + 1
 y1:  -------------------
      s^3 + s^2 + 2 s + 1

Continuous-time model.
```




```
Read https://www.octave.org/bugs.h
For information about changes fro

>> pkg load control
>> num1 = [1 1];
>> den1 = [1 1 2 1];
>> sys = tf(num1, den1)

Transfer function 'sys' from input

            s + 1
 y1:  -------------------
      s^3 + s^2 + 2 s + 1

Continuous-time model.
>> step(sys);
>> grid ;
```

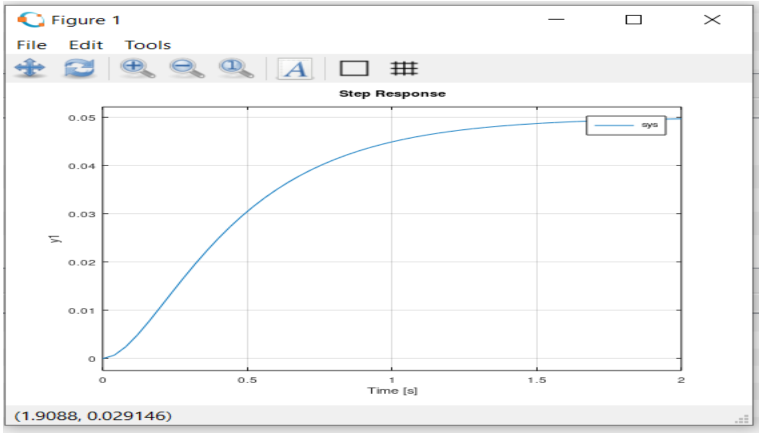
TP 1 :




```
num1 = [1 1];
den1 = [150 0.05 5];
sys = tf(num1, den1)

step(sys);
```

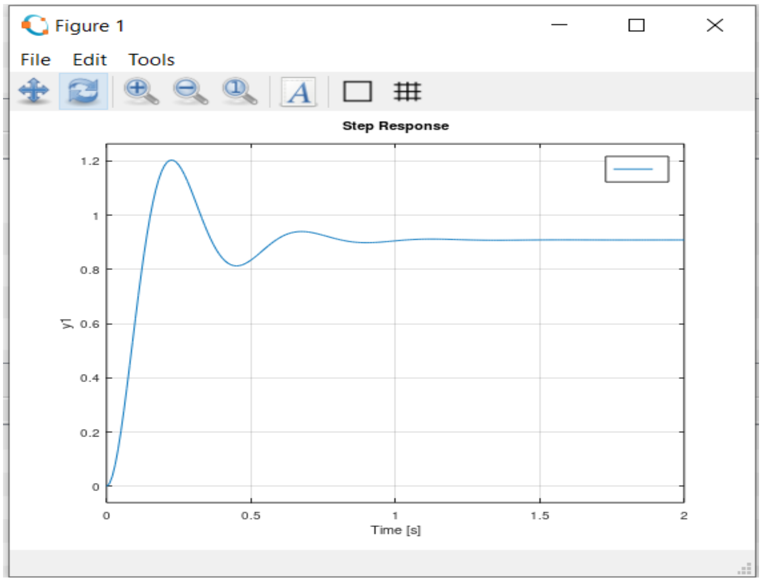
TP 2 :

1. $X(s) / F(s) = 1 / (1s^2 + 10s + 20)$



La correction fait tendre le signal à 0.5 et elle est lente

2.



Correction plus rapide faite à 0.09

3. L’utilisation du gain proportionnel réduit le temps de montée, augmente le dépassement et réduit l'erreur en régime permanent (d’après le doc)

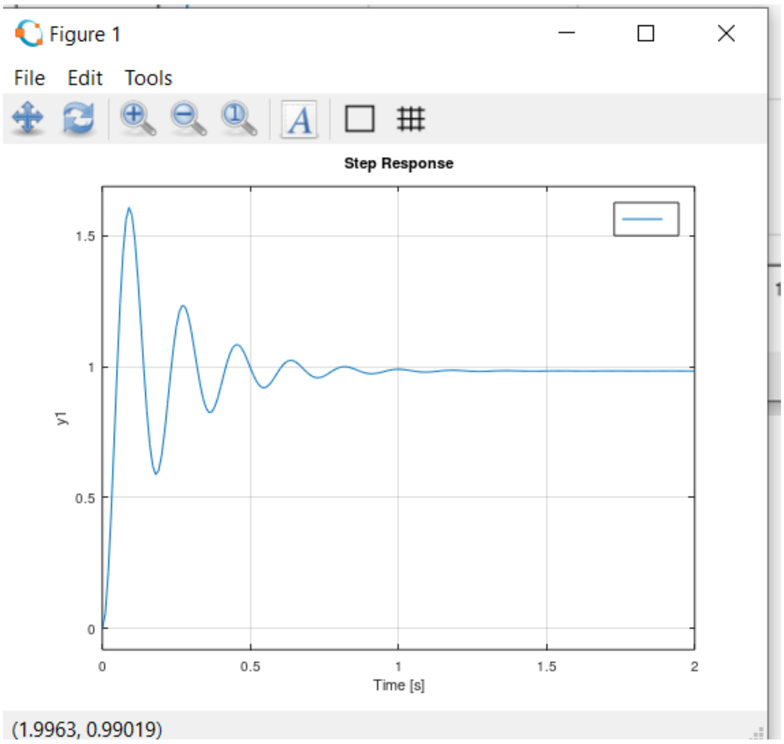
4. La valeur tend vers 1 pour Kp = 1200

Temps de monté : plus court,

Dépassement : monte plus haut,

Temps de réponse : plus rapide,

Erreur statique : moins d’erreur



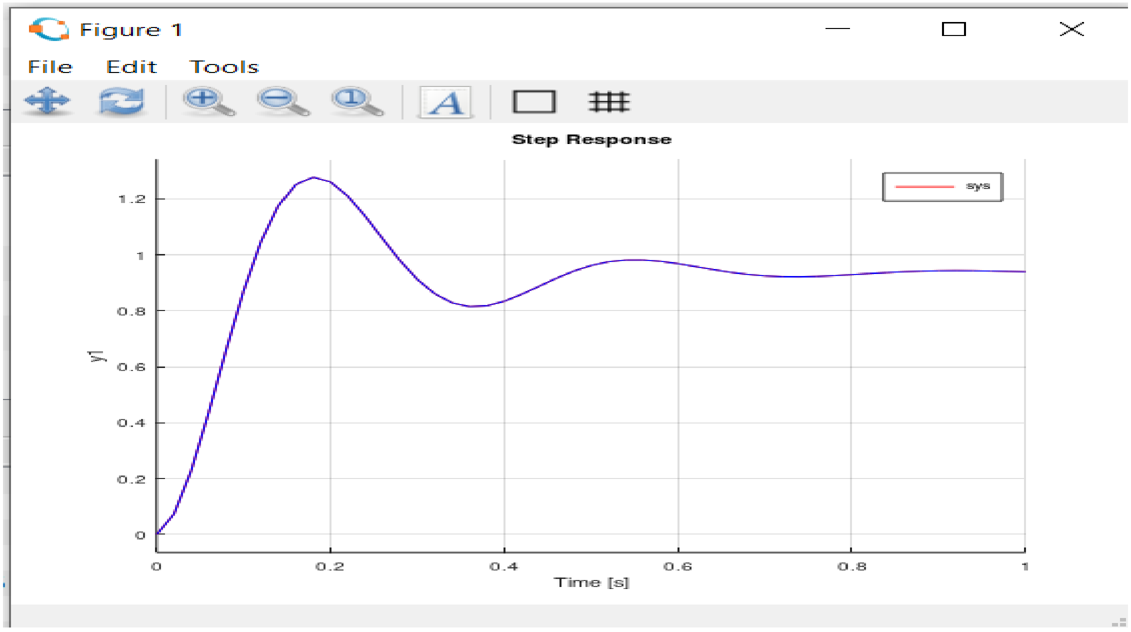
5. X(s)/F(s) $= ((KpS + Kp) / (mS^2 + cS + K)) / 1 + ((KpS + Kp) / (ms^2 + cs + K))$

$= KpS + Kp / ms^2 + cS + K + KdS + Kp$

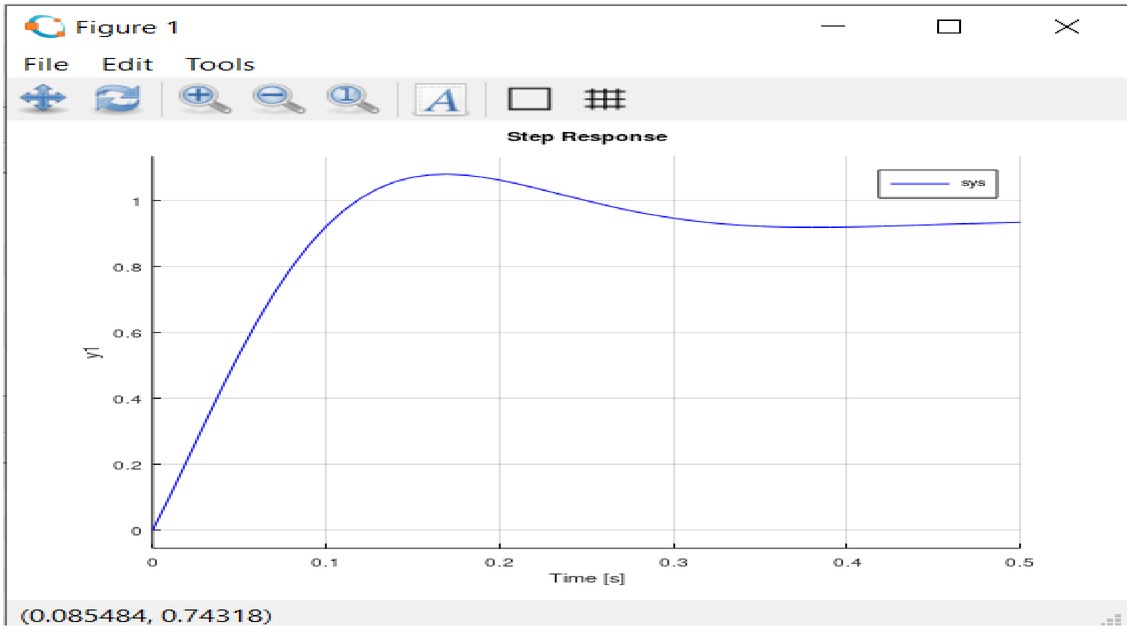
$= KpS + Kp / ms^2 + (c+Kp)\ s + (K + Kp)$

6.

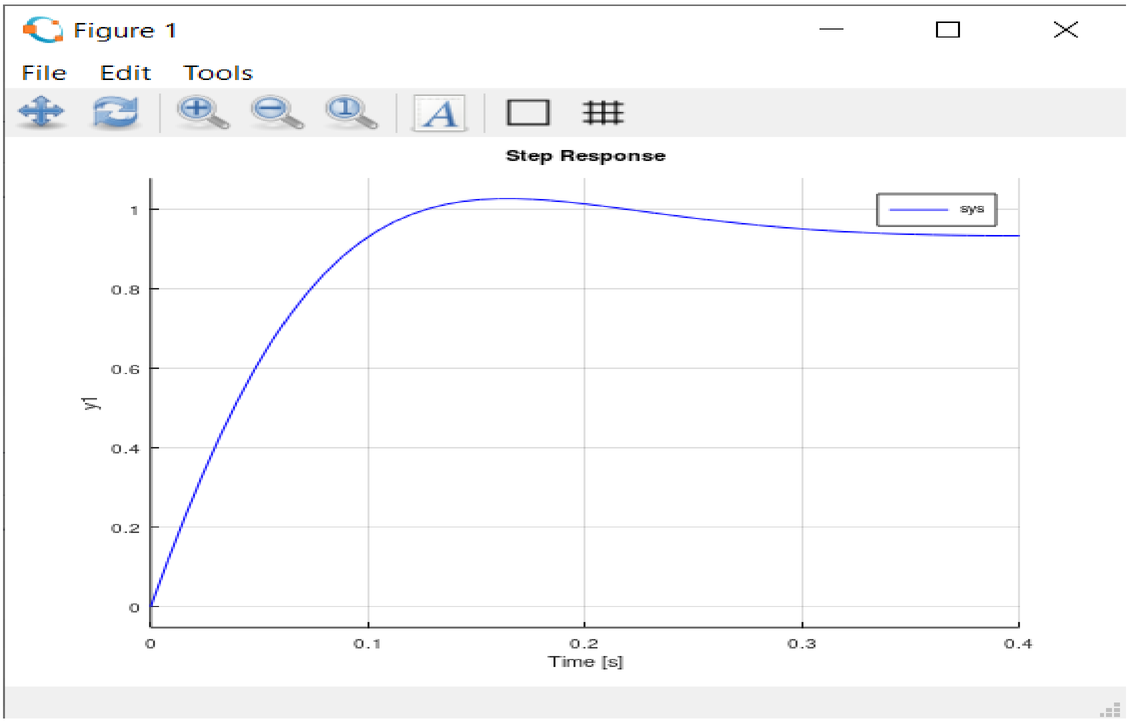
Kp = 1



Kp = 10



$Kp = 15$



7. Le premier dépassement est moins haut et le tr met plus de temps, la valeur de Kp qui fonctionne le mieux est $Kp = 1$

8. $X(s) / F(s) \quad = (Kps + KI) / p^2(m + c) + s\,(Kp + K) + KI$

9.

10.

11. $X(s) / F(s) \quad = (((1/s)KI + Kp + skd) / (ms^2 + cs + k)) / 1 + (((1/s)KI + Kp + skd) / (ms^2 + cs + k))$

$= ((1/s)KI + Kp + skd) / (ms^2 + cs + k + (1/s)KI + Kp + skd)$

$= (Kl + skp + s^2 Kd) / (ms^3 + cs^2 + Ks + Kl + Kps + Kds^2)$

$= (Kds^2 + Kps + Kl) / ((ms^3 + (cKd)s^2 + (K + Kp)s + Kl)$

12.

13.