

## Module : Logiciels de simulation

### La commande d'une Machine à courant continu via un régulateur PI

(Corrigé type)

#### INTRODUCTION (0.5pt)

#### PARTIE MATLAB (3.75pts)

En utilisant un fichier SCRIPT :

1. Ecrire la fonction du transfert du moteur CC sur MATLAB dans le cas où

- La tension est nulle
- Le couple de résistant est nul

```
R=1.4;  
L=4.055*1e-3;  
K=0.4095;  
F=0.0002;  
J=0.02;  
H_sstension=tf(-[L R],[J*L (F*L+R*J) F*R+K^2]);0.25pt  
H_sscouple=tf([K],[J*L (F*L+R*J) F*R+K^2]);0.25pt
```

#### EXECUTION

H\_sstension =

-0.004055 s - 1.4

-----  
8.11e-05 s^2 + 0.028 s + 0.168  
Continuous-time transfer function.

H\_sscouple =

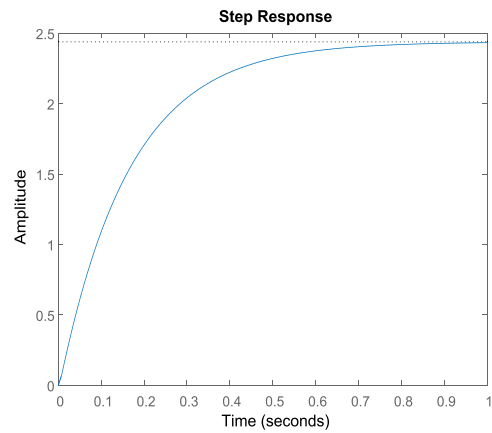
0.4095

-----  
8.11e-05 s^2 + 0.028 s + 0.168  
Continuous-time transfer function.

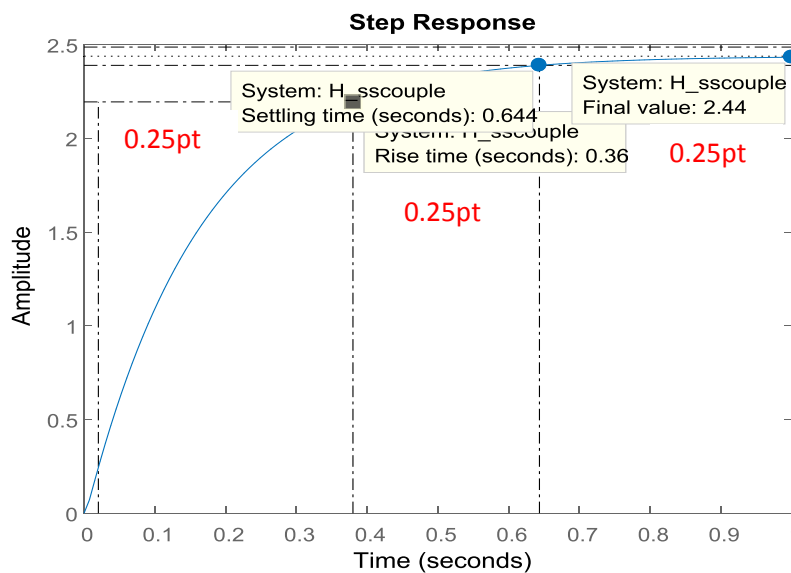
On pose dans la suite  $C_r=0$ ,

**2. Tracer la vitesse du moteur CC  $\Omega_m(t)$  dans le cas d'une entrée échelon unitaire**

```
figure(1)
step(H_sscouple) 0.25pt
```

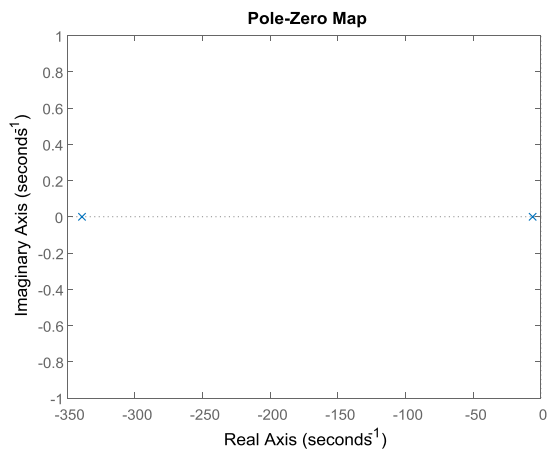


A partir du tracé, caractériser la réponse en termes de temps de montée, temps de réponse et précision.



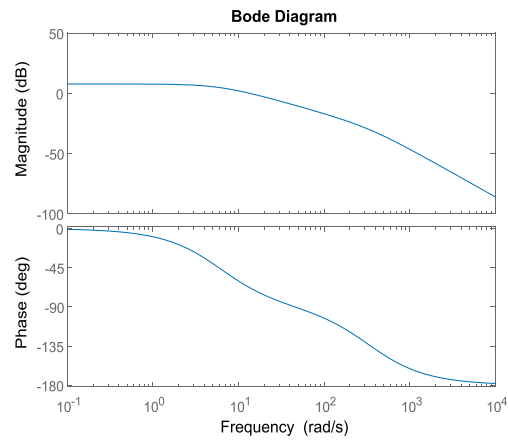
**3. Représenter les pôles et les zéros du moteur CC en Boucle Ouverte (BO).**

```
figure(2)
pzmap(H_sscouple) 0.25pt
```



4. Donner le diagramme de Bode du moteur CC

```
figure(3)
bode (H_sscouple) 0.25pt
```

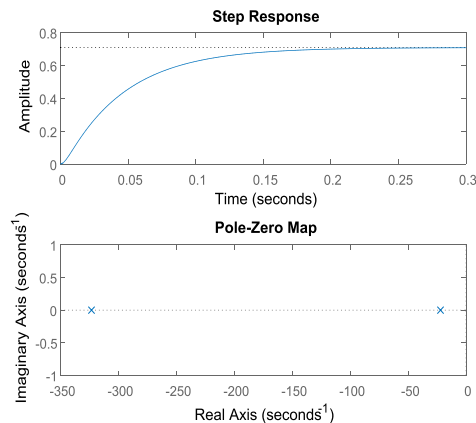


5. Déterminer la fonction du transfert en Boucle Fermée (BF) avec retour unitaire.

```
H_sscouple_BF=feedback(H_sscouple,1,-1) 1pt
```

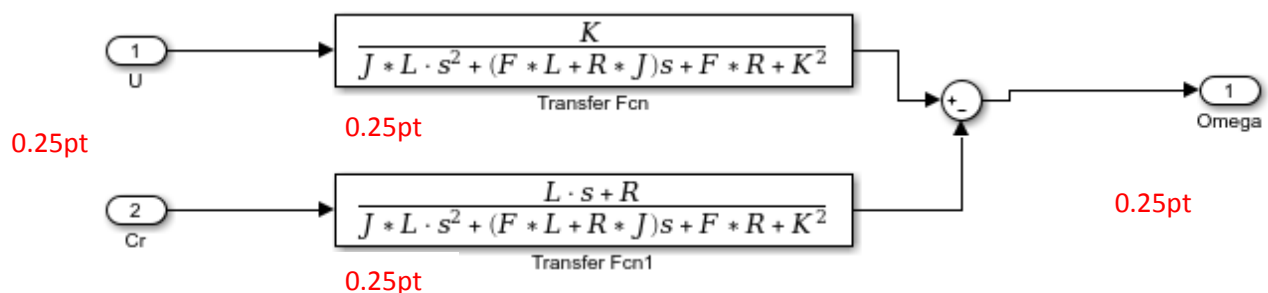
6. Dans une seule figure substituée en deux sous figures, représenter la réponse indicielle et les pôles et zéros de la fonction du transfert BF.

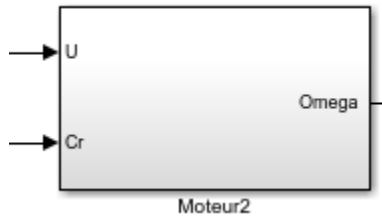
```
figure(4)
subplot(2,1,1) 0.25pt, step(H_sscouple_BF) 0.25pt
subplot(2,1,2), pzmap(H_sscouple_BF) 0.25pt
```



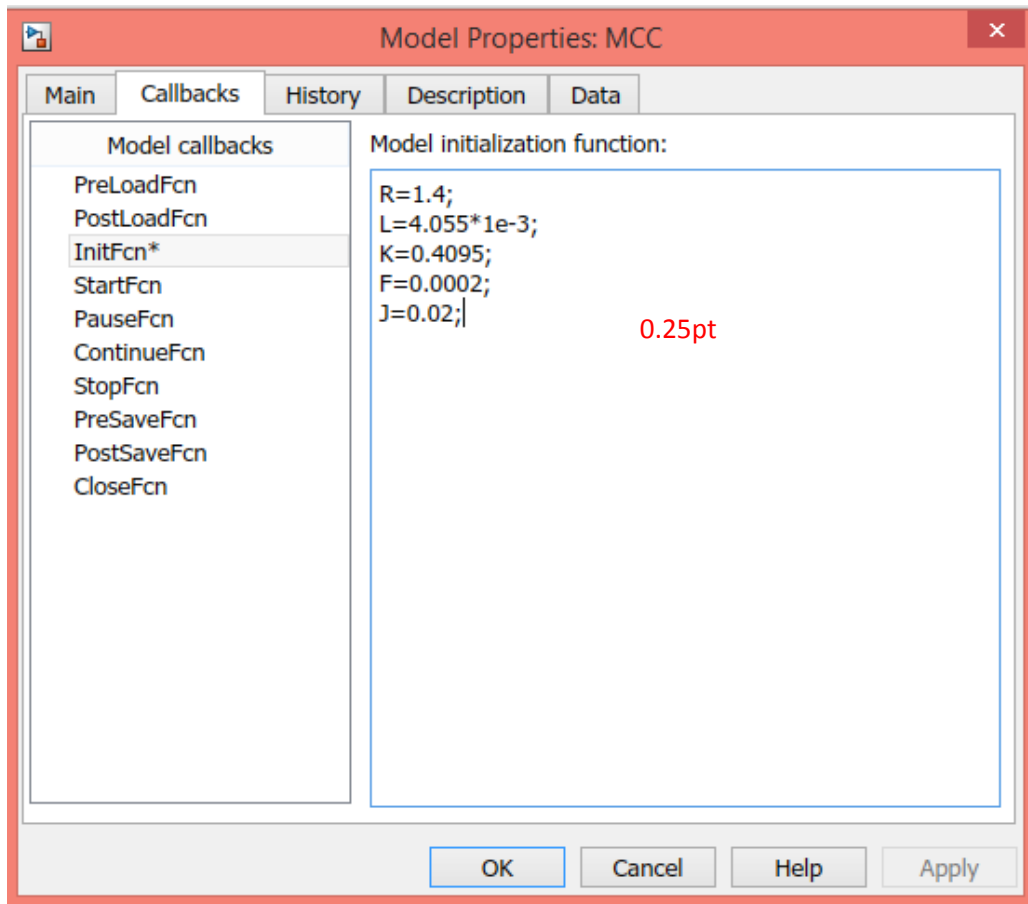
**PARTIE SIMULINK (11.25pts)**

1. Réaliser à l'aide des blocs SIMULINK le schéma du moteur à courant continu. Mettre le sous la forme d'un « Sous Système » à deux entrées-une seule sortie.

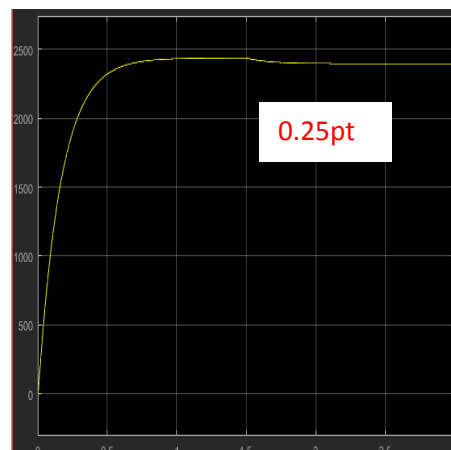
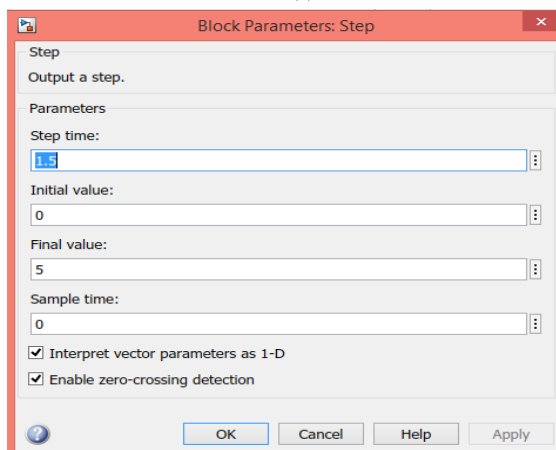




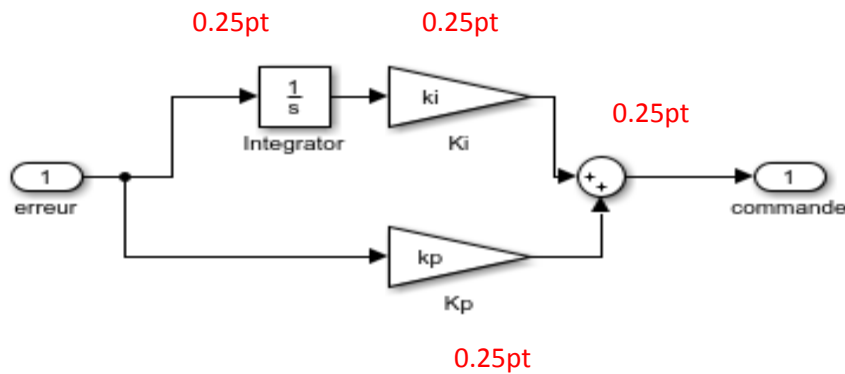
2. Utiliser les callbacks SIMULINK pour introduire les différents paramètres du moteur.



3. Fixer le temps de simulation à 3 secondes 0.25pt, la tension à 1000v 0.25pt et imposer un couple résistant de 5N.m à l'instant 1,5 seconde 0.25pt. Visualiser la courbe de la vitesse  $\Omega_m(t)$  en BO.

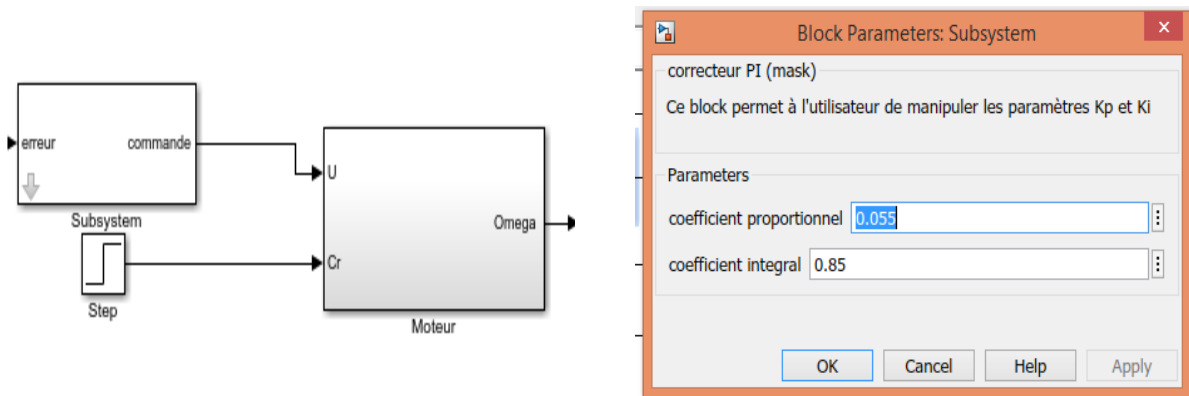


4. On souhaite contrôler ce moteur par un régulateur PI branché en série avec le moteur.  
La fonction de transfert du régulateur s'écrit comme suit :

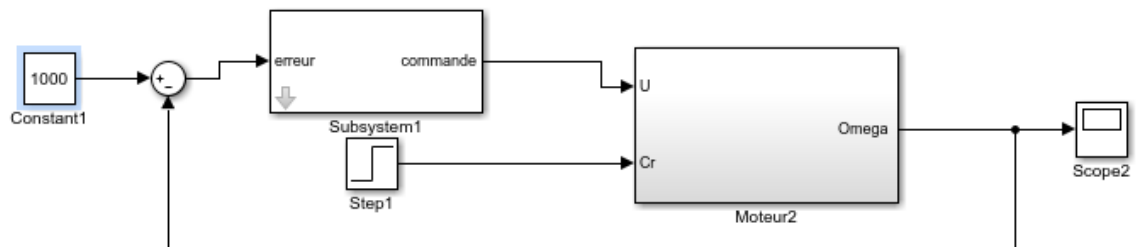


$$R(p) = K_p + \frac{K_I}{p} \quad , \text{ avec } K_p = 0.055 \text{ et } K_I = 0.85$$

Insérer le régulateur avant le bloc moteur CC. Utiliser le masquage SIMULINK pour pouvoir manipuler les paramètres de ce correcteur. 1pt

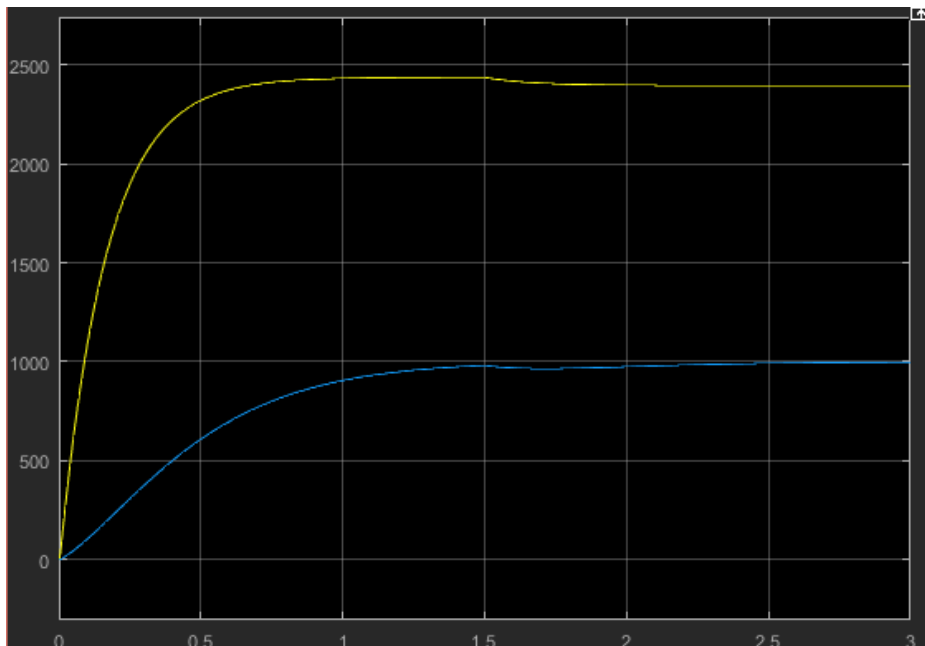
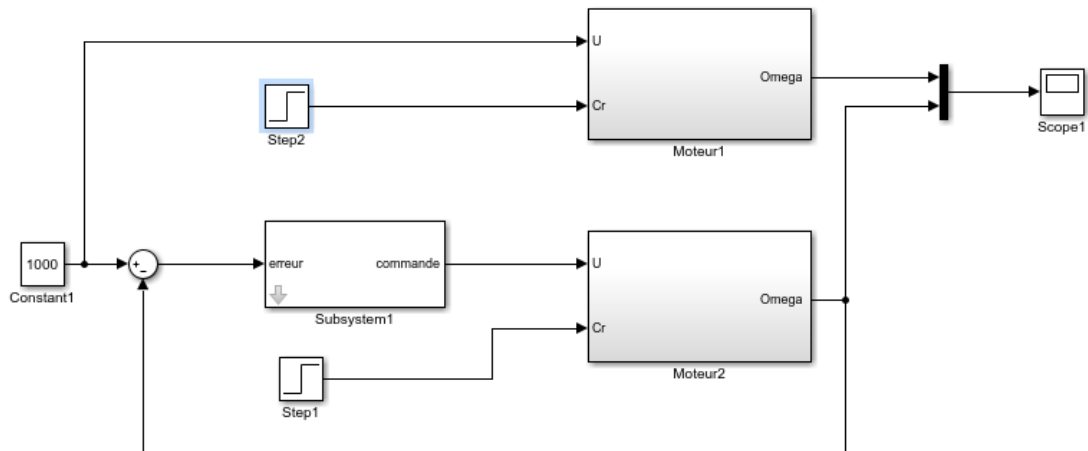


5. Réaliser le schéma de commande de vitesse du moteur associé au régulateur PI (i. e. commande en boucle fermée) 1pt



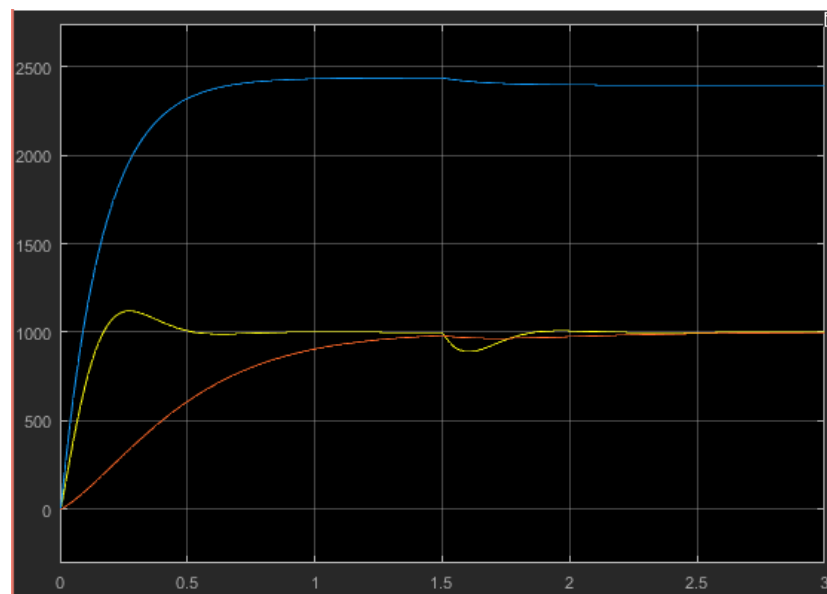
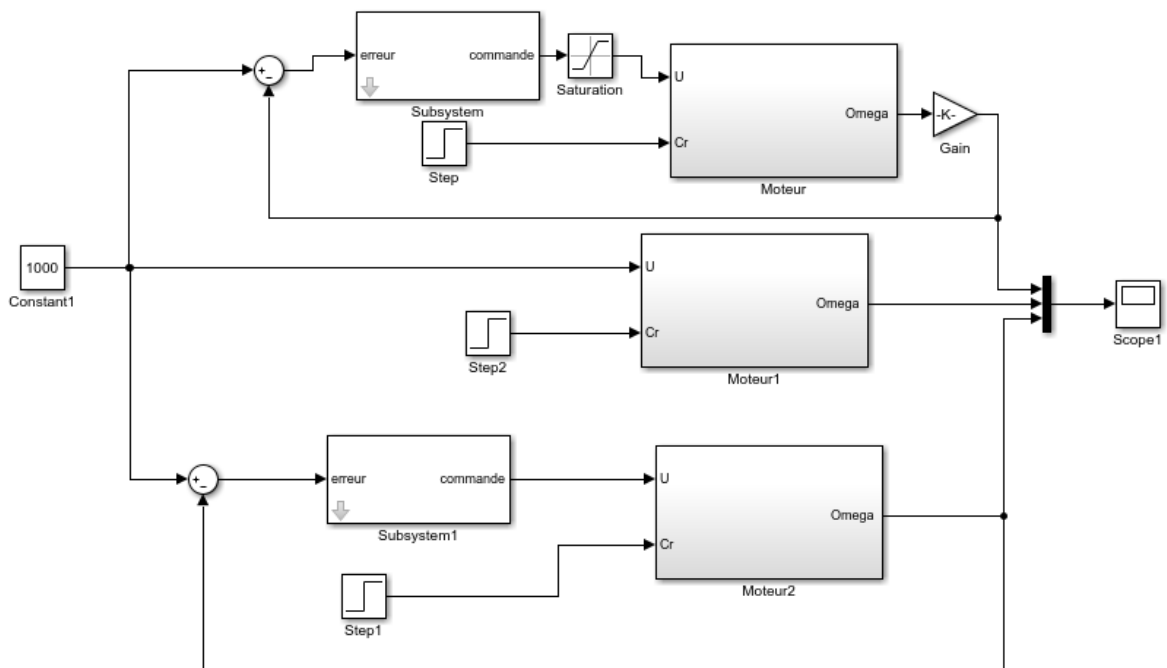
6. A l'aide d'un multiplexeur, visualiser sur la même figure la vitesse du moteur  $\Omega_m(t)$ .  
0.25pt

Expliquer l'allure du graphe obtenu.



La sortie (vitesse du moteur) suit l'entrée ( $U=1000v$ ) lorsque le régulateur PI est inséré. **0.25pt**

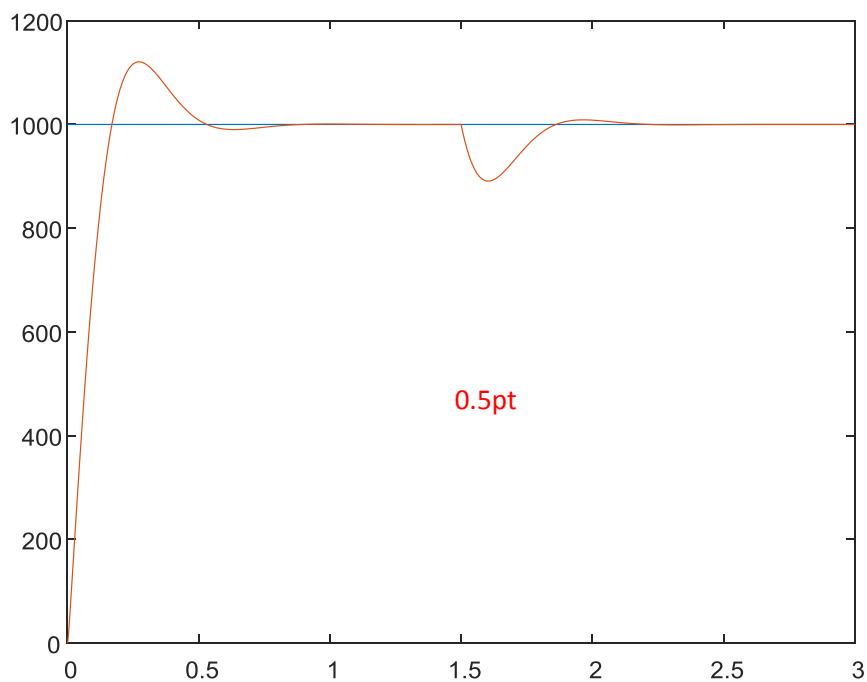
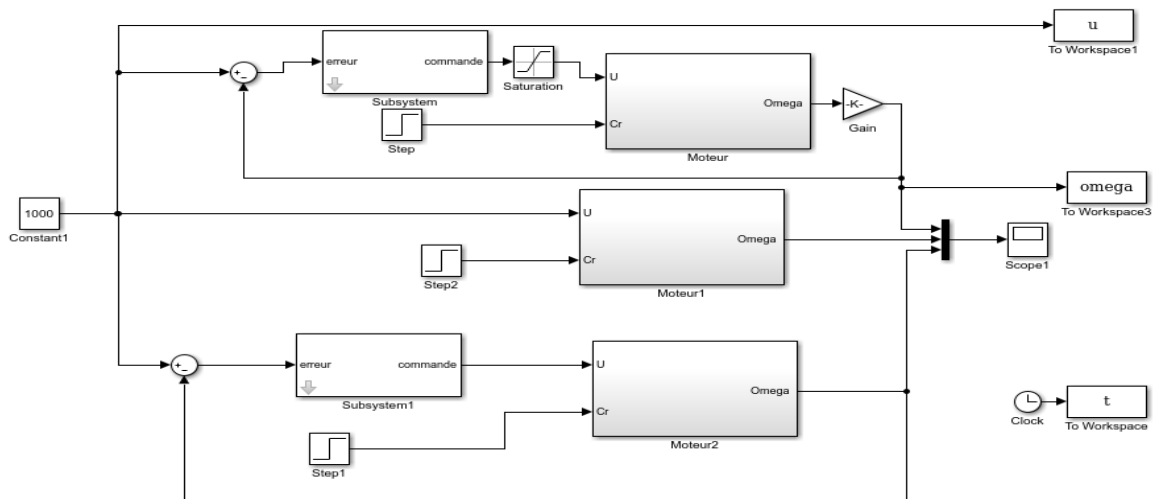
- 7.** Placer un gain  $K=9.5492$  à la sortie du bloc moteur **0.25pt** et un bloc de saturation (borne inférieure=0 et supérieure =100) à la sortie du bloc régulateur **0.25pt**. Comparer les résultats obtenus avec les précédents. Conclure.



Après insertion du gain et du bloc de saturation le temps de montée est réduit mais la perturbation à  $t=1.5$ s du couple résistant ( $C_r$ ) est plus visible **0.5pt**. Le gain et le bloc de saturation rendent le système plus rapide et amplifient l'effet de la perturbation.**0.5pt**

8. Jouer sur les paramètres  $K_p$  et  $K_I$  à travers le masque ainsi créé 0.5pt. Donner la courbe de la vitesse dans chaque cas 0.5pt. Justifier 0.5pt (tout dépend des valeurs choisies)
9. Introduire le temps via le bloc « Clock » 0.5pt, enregistrer les résultats dans l'espace de travail de MATLAB 0.5pt puis reproduire sur la même figure la courbe de vitesse  $\Omega_m(t)$  et d'entrée  $u(t)$  à partir de MATLAB. Commenter .

```
>> plot(t,u,t,omega)0.5pt
```





Le bloc « To workspace » nous permet de transmettre les données de SIMULINK à l'espace de travail MATLAB. Donc, de redessiner une figure affichée par un oscilloscope en MATLAB via la commande plot **0.5pt**.

**CONCLUSION (0.5pt)**

**FICHER MATLAB EXECUTABLE « .m » (2pts)**

**FICHER SIMULINK EXECUTABLE « .xls » (2pts)**