Ministère de l'enseignement supérieure et de la recherche scientifique de Tunisie

Ecole Polytechnique De Tunisie

Rapport du TP théorie de la commande

Contrôle du manipulateur robotique à joint flexible

Réalisé par:
BACCAR Ala
BEN RHOUMA Abdelhamid

Encadré par:
M.Naceur BEN HADJ BRAIEK
M.Mohamed karim BOUAFFOURA

Année universitaire: 2021/2022

Sommaire

1 Introduction

D'énormes recherches sont menées dans les domaines pertinents tels que la modélisation et le contrôle des robots. Une des fonctions très courante exercée par les robots modernes est la manipulation d'objets dans leur environnement . En général ces manipulateurs ressemblent à des bras humains et ont plusieurs Degré de liberté

Dans les séances de travaux pratique de la théorie de commande, nous allons étudier la modélisation et le contrôle sophistiqué d'un manipulateur robotique à joint flexible. Le but de ce TP est de maintenir l'angle de rotation du lien à la position souhaitée. Donc le rapport va présenter premièrement le système et ses caractéristique, son modèle mathématique, puis il va illustrer la méthode de commande par retour d'état et ensuite la correction avec le correcteur PID, et enfin i l y aura une partie pour la commande numérique, et dans la conclusion il y aura une comparaison entre les différentes types de commande.

2 Présentation du système

2.1 Description du système

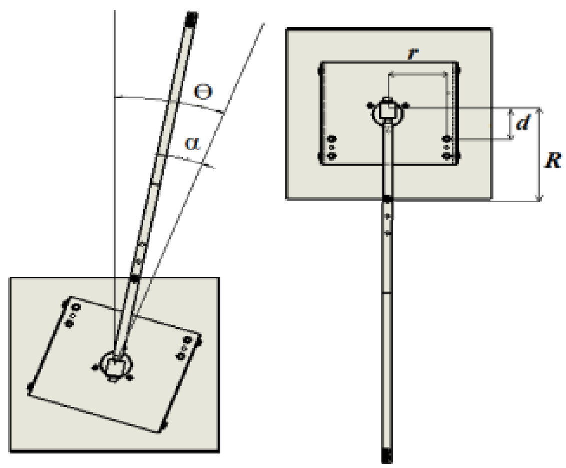
Le module Joint Flexible Rotatif se compose d'un bras libre attaché à deux ressorts identiques. Les ressorts sont montés sur un châssis en aluminium qui est fixé à l'engrenage de charge de l'unité de base du servomoteur rotatif. Le module se fixe à un moteur à courant continu sur l'unité de base servo qui fait tourner une poutre montée sur un joint flexible. Le module de joint flexible rotatif est fourni avec trois paires distinctes de ressorts, chacun avec une rigidité variable. Il est livré avec trois ancrages de base et trois bras pour les ressorts, et permet d'obtenir une grande variété de rigidité articulaire. Le module est également fourni avec une charge de bras variable qui peut être montée à trois positions d'ancrage distinctes pour modifier la longueur de la charge.

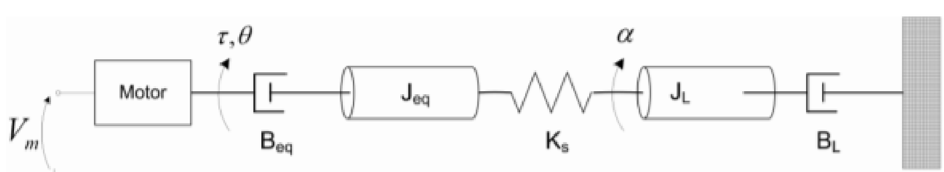
2.2 Utilisation

Ce modèle est utilisé essentiellement pour étudier les concepts rencontrés dans les grands joints de robots à engrenages avec la flexibilité exposée dans la boîte de vitesses, ainsi que les concepts liés au contrôle des vibrations et de la résonance, et à la modélisation articulations flexibles sur robots ou engins spatiaux.

3 Modèle mathématique du système

Un modèle mathématique pour un manipulateur flexible à articulation unique peut être obtenu facilement à partir des équations de Lagrange. Ainsi, Comme montré dans la figure ci-dessous, le système a deux degrés de libertés et l'articulation qui est montée sur l'arbre se déplace en fonction de sens de la rotation du moteur. θ est l'angle d'asservissement, α est l'angle de joint flexible.





Les valeurs et symboles des paramètres système sont résumé dans le tableau 1.

Paramètre	Symbole	Valeur	Unité
Inertie du manipulateur flexible	j_l	0.003882	kg.m^2
Résistance du moteur	R_m	15.5	Ω
Rapport de réduction du réducteur	k_g	1/36	kg.m^2
Constante du moteur	k_m	0.0089	N/(rad/sn)
Coefficient de flexibilité du joint	k_s	5.468	N/m
Coefficient de fraction visqueuse du manipulateur	B_{eq}	0.001	N.m.s/rad
Coefficient de fraction visqueuse du lien	B_L	-0.01	N.m.s/rad
Inertie de la plate-forme de rotation	j_h	0.00035	kg.m^2

L'équation lagrangienne est calculée à partir de la cinétique et l'énergie potentielle.

$$L = T - V$$

où T et V sont respectivement les énergie cinétique de rotation et l'énergie élastique et ils s'écrivent comme suit:

$$\begin{aligned} T &= \frac{1}{2}J\omega^2 \\ &= \frac{1}{2}J_{eq}\dot{\theta}^2 + \frac{1}{2}J_L(\dot{\theta} + \dot{\alpha})^2 \\ V &= \frac{1}{2}K_s\alpha^2 \\ L &= \frac{1}{2}J_{eq}\dot{\theta}^2 + \frac{1}{2}J_L(\dot{\theta} + \dot{\alpha})^2 - \frac{1}{2}K_s\alpha^2 \end{aligned}$$

Les équations du mouvement de Lagrange sont données comme suit:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = \tau - B_{eq}\theta$$
$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} = B_L\dot{\alpha}$$

Aprés un simple dérivé, on trouve les equations differentielles:

$$(j_l + j_{eq})\ddot{\theta} + j_l\ddot{\alpha} = \tau - B_{eq}\theta \quad (1)$$

$$j_l(\ddot{\theta} + \ddot{\alpha}) + K_s\alpha = B_L\dot{\alpha} \quad (2)$$

Alors nous somme face à un modèle linéaire. Avec τ est le couple moteur, le couple est obtenu par la tension appliquée à l'induit. Dans cette étude, la tension V est déterminée comme entrée du système. Également La relation entre le couple et la tension peut être exprimée, à l'aide de la loi de kirchhoff, comme suit :

$$v = iR_m + k_m k_g \omega$$

$$i = \frac{v}{R_m} - \frac{K_m K_g}{R_m}\omega$$

où ω est la vitesse angulaire du moteur, i est le courant de l'armature.En outre,

$$i = \frac{\tau}{K_g K_m}, \omega = \dot{\theta}$$

Finalement on aura la relation entre le couple et la tension,

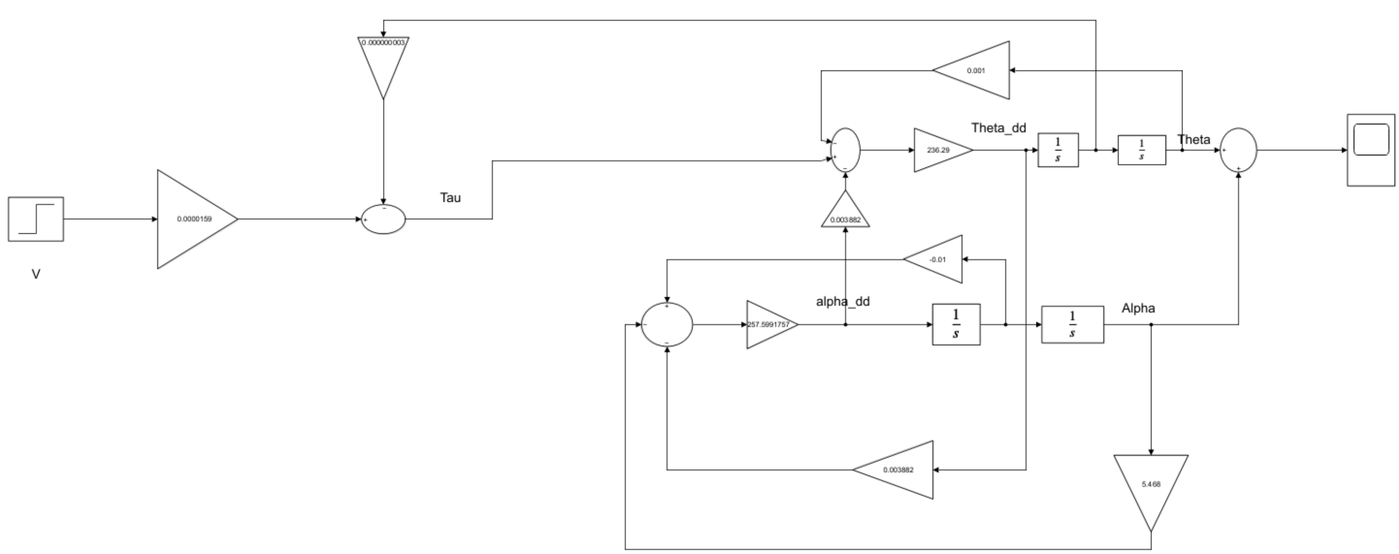
$$\tau = \frac{K_m K_g}{R_m}v - \frac{K_m^2 K_g^2}{R_m}\dot{\theta} \quad (3)$$

4 Simulation du système avec Simulink

Dans cette partie, nous allons faire une simulation pour le système avant de faire l'étude la stabilité et de la commandabilité, afin d'analyser son comportement. A partir des equations (1) et (2) on aura,

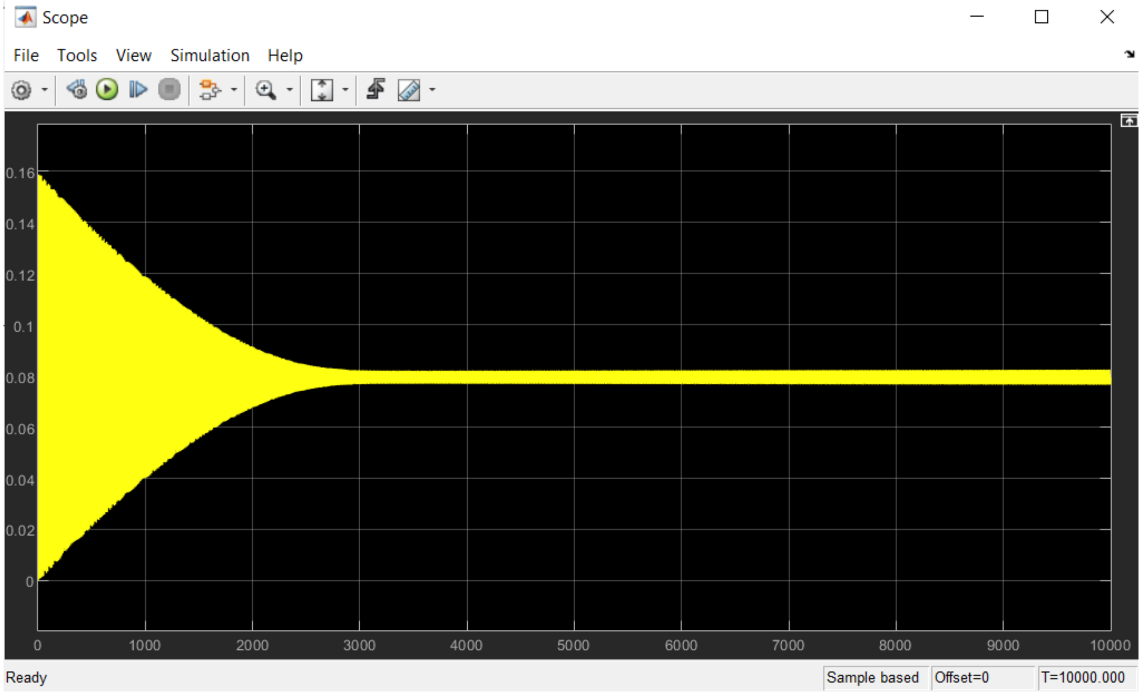
$$\ddot{\theta} = \frac{1}{j_{eq} + j_l}(-j_l\ddot{\alpha} + \tau - B_{eq}\theta)$$
$$\ddot{\alpha} = \frac{1}{j_l}(-j_l\ddot{\theta} - K_s\alpha + B_L\dot{\alpha})$$

L'entrée de notre système est la tension v, et on a deux sorties α et θ donc on va considerer la sortie de système y=θ+α

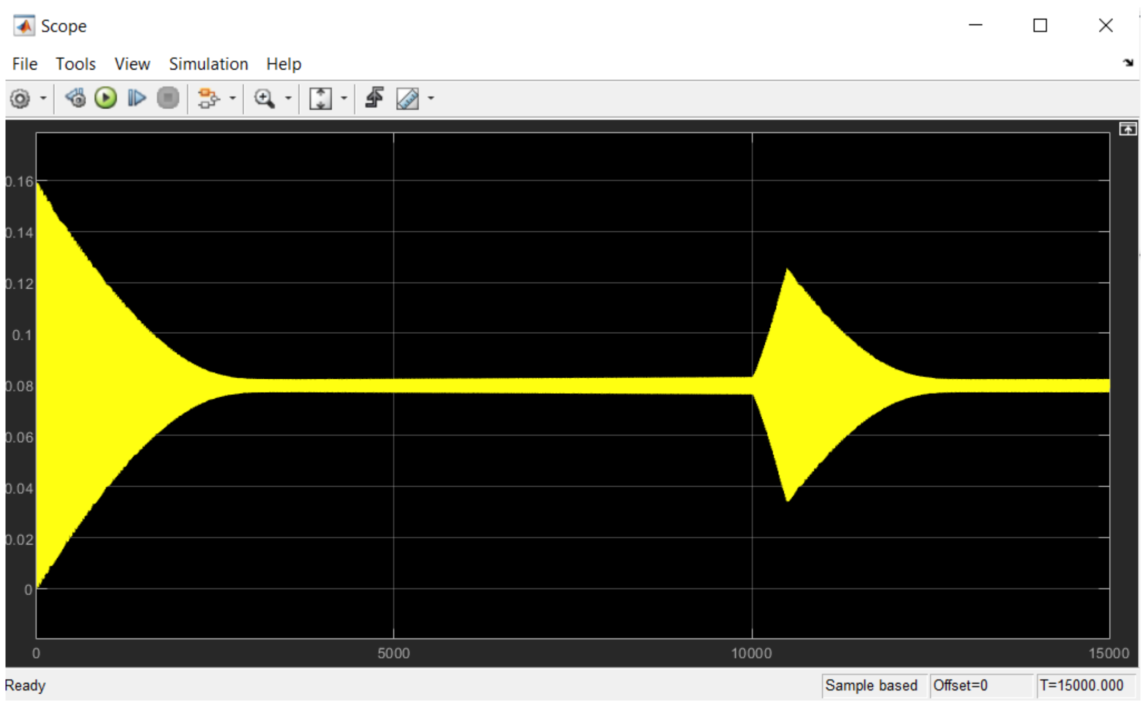


Le système a montré un comportement oscillatoir pour une entrée de 10volts, comme le montre la figure ci-dessous.

Pour une entrée de V=5v, et un stop time=10000, nous avons eu le comportement suivant:



Lorsque nous avons augmenté le stop time à 15000, nous avons eu un comportement un peu bizzard, illustré dans la figure ci-dessous.



Interprétation: En premier temps, le système a convergé aprés un certain instant à la valeur 0.08, puis il a divergé un peu, et enfin il a retourné à sa valeur initiale. On peut déduire que ce système a un comportement chaotique, pour cela dans la suite nous allons faire l'étude nécessaire pour la correction y compris la stabilité et la commandabilité.

5 Répresentation d'état et commande analogique

En substituant l'equation (3) dans l'equation (1), on obtient l'equation (4)

$$(j_l + j_{eq})\ddot{\theta} + j_l\ddot{\alpha} = \frac{K_m K_g}{R_m}v - \frac{K_m^2 K_g^2}{R_m}\dot{\theta} - B_{eq}\theta \quad (4)$$

En posant $x_1 = \alpha, x_2 = \dot{\alpha}, x_3 = \theta, x_4 = \dot{\theta}$,les equations (2) et (4) peuvent etre représenté sous la forme d'espace d'état suivante:

$\dot{x_1} = x_2$
$\dot{x_2} = -\frac{K_s}{j_l}x_1 + \frac{B_L}{j_l}x_2 + \frac{K_s}{j_l}x_3 - \frac{B_L}{j_l}x_4$
$\dot{x_3} = x_4$
$\dot{x_4} = -\frac{K_s}{j_{eq}}x_1 + \frac{B_L}{j_{eq}}x_2 + \frac{K_s}{j_{eq}}x_3 - (\frac{k_m^2 K_g^2}{R_m J_{eq}} - \frac{B_L + B_{eq}}{J_{eq}})x_4$
$y = x_1 + x_2;$

Donc on aura le système suivant;

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{-K_s}{j_l} & \frac{B_L}{j_l} & \frac{K_s}{j_l} & \frac{B_L}{j_l} \\ 0 & 0 & 0 & 1 \\ -\frac{K_s}{j_{eq}} & \frac{B_L}{j_{eq}} & \frac{K_s}{j_{eq}} & \frac{k_m^2 K_g^2}{R_m J_{eq}} - \frac{B_L + B_{eq}}{J_{eq}} \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ \frac{K_g k_m}{R_m j_{eq}} \\ \frac{-K_g k_m}{R_m j_{eq}} \end{pmatrix}$$

$$Et \quad C = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}$$

5.1 Etude de la stabilité

On calcule les valeurs propres de la matrice A à l'aide du script ci-dessous, on trouve les valeurs propres suivantes: 131.1940, -108.3389,0.2831, -0.0000. Donc notre système est non stable puisque il possède un pole réelle positive.

Script

```
%Les données
Ks=5.468;jl=0.003882;jeq=0.00035;Rm=15.5;Km=0.0089;Kt=0.0089;Beq=0.001;Kg=
    1/36;
Bl=-0.01;
Const=(((Km^2)*(Kg^2))/(Rm*jeq))-(Bl+Beq)/jeq;
%Les matrices du modèles d'état
A=[0 1 0 0;-Ks/jl Bl/jl Ks/jl -Bl/jl; 0 0 0 1; -Ks/jeq Bl/jeq Ks/jeq Const
    ];
B=[0; 0; (Kg*Km)/(Rm*jeq);-(Kg*Km)/(Rm*jeq)];
C=[1 1 0 0];
[D,poles]=eig(A);
```

Maintenant pour fixer ce comportement indésirable, nous allons commander ce système par retour d'état, mais en premier lieu nous devons étudier la commandabilité et l'observabilité du notre système.

5.2 Etude de la commandabilité et de l'observabilité

Aprés l'execution du script ci-dessous, on a trouvé que les déterminants de la matrice d'observabilité et de commandabilité sont différentes de zéro, ils sont respectivement -3.5935e+10 et -4.8248e+05, donc notre système est commandable et observable. Dans ce cas on peut faire la commande.

Script

```
%Etude de commandabilité et d'observabilité du système
Co = ctrb(A,B);%Matrice de commandabilité
dCo=det(Co);
oB=obsv(A,C); %Matrice d'observabilité
doB=det(oB);
```

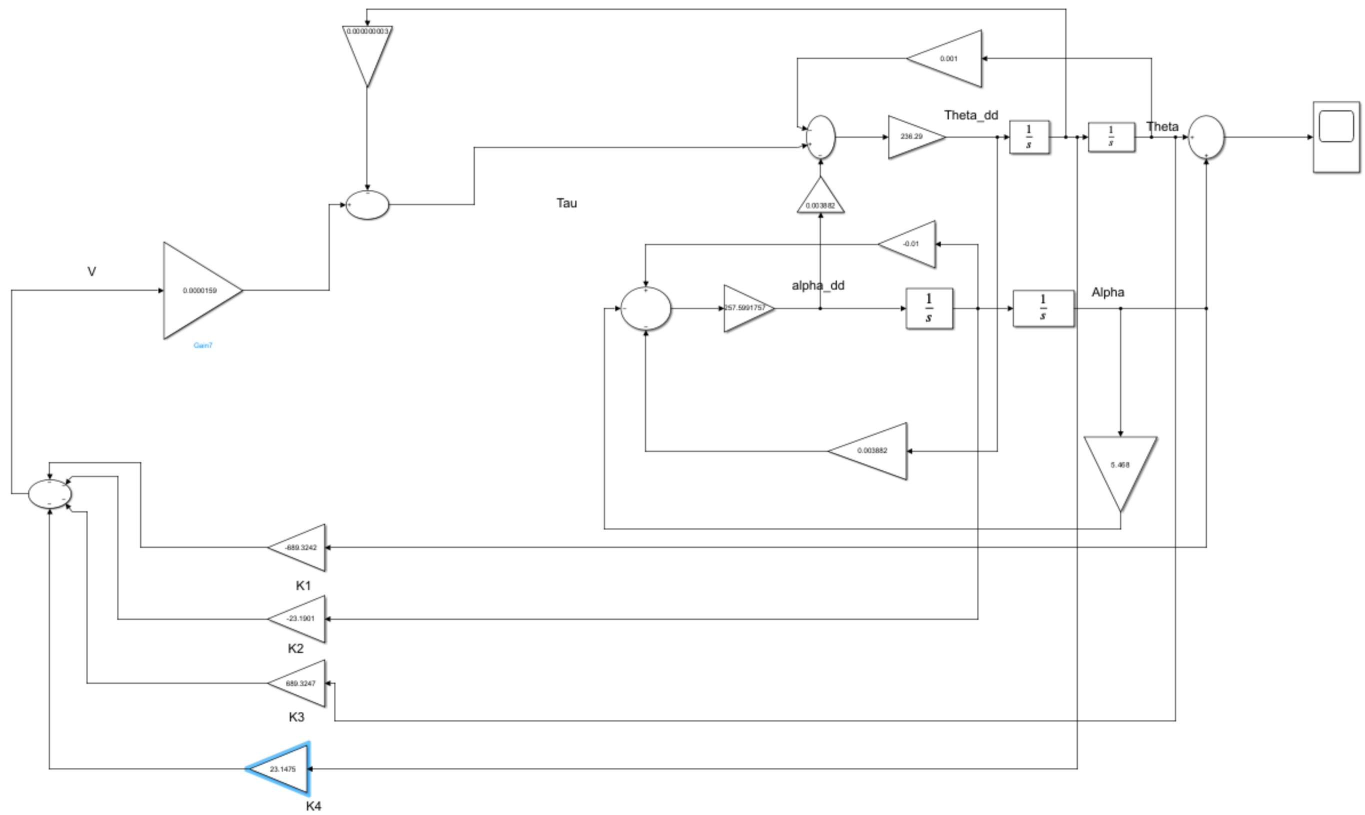
5.3 Commande par retour d'état sans consigne

Dans cette partie nous allons faire une commande par retour d'état sous la forme:v=-Kx avec k=(k1 k2 k3 k4) et x est le vecteur d'état. Nous allons déterminer les gains ki de tel sorte on aura un comportement d'un système composé de deux systèmes de deuxième ordre en cascade dont les paramètres du premier est Wn1=5rd/s et $\zeta 1 = 0.7$ et du deuxième sont Wn2=3rd/s et du même coefficient d'amortissement que le premier, donc les pôles désirés du système seront:[-3.5+3.5i,-3.5-3.5i,-0.01,-0.21], et cela conduit nécessairement à un système stable. Les valeurs du gain obtenu sont: K=[-689.3242 -23.1901 689.3247 23.1475].

Script

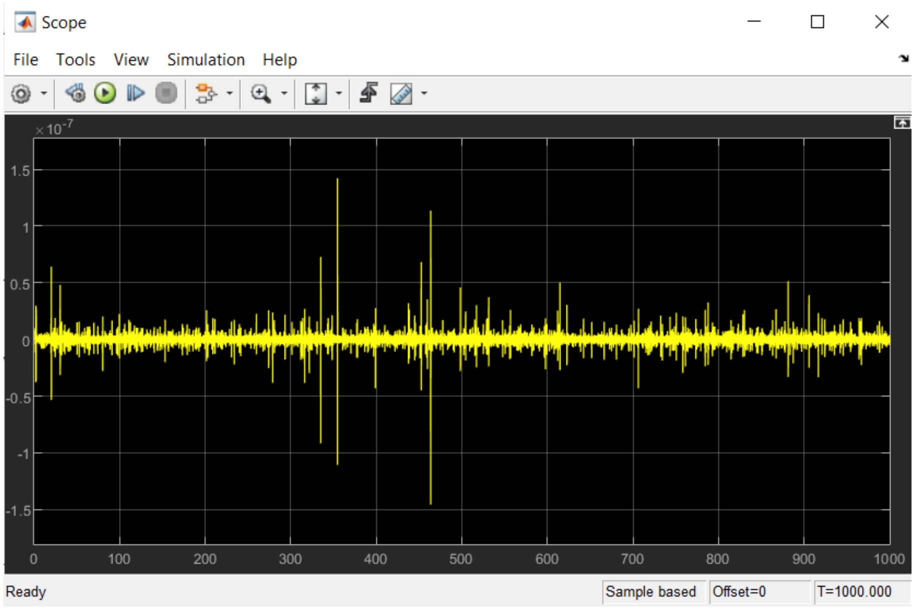
```
%u=-Kx
desired_poles=[-3.5+3.5i,-3.5-3.5i,-0.01,-0.21];
K=place(A,B,desired_poles);
```

Nouveau shéma du modèle(Simulink)



Avec cette commande, nous avons garantit la stabilité mais nous n'avons pas obtenu une réponse importante du système, mais ce problème était corrigé lorsque nous avons ajouté une consigne au système comme l'illustre la partie suivante.

Réponse non intéressante



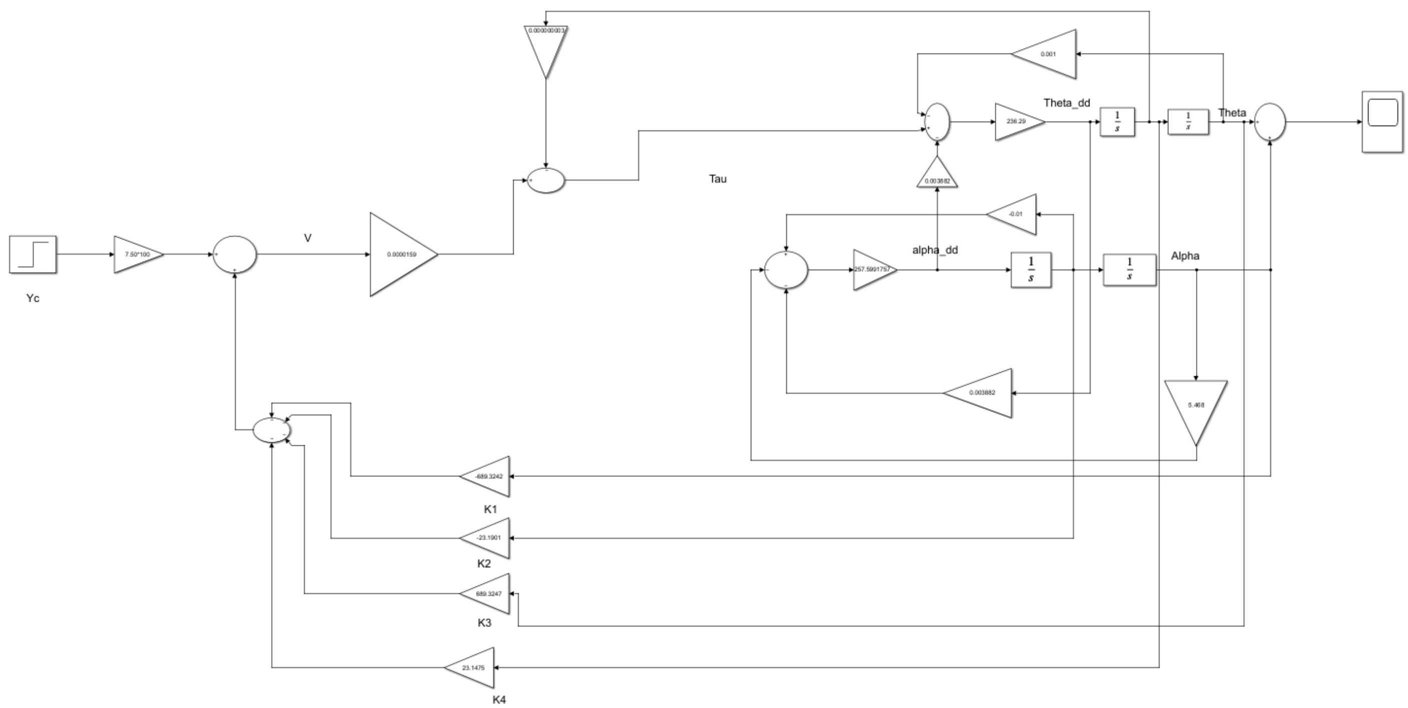
5.4 Commande par retour d'état avec consigne

Dans cette partie nous allons faire une commande par retour d'état mais cette fois en ajoutant une consigne, donc v aura la forme suivante:v=-Kx+Nyc, avec N une constante à déterminer de telle sorte on aura une réponse sans erreur statique, c'est à dire le gain du système est ègale à 1, donc $N^{-}1$=-C$(A-BK)^{-}1$B.

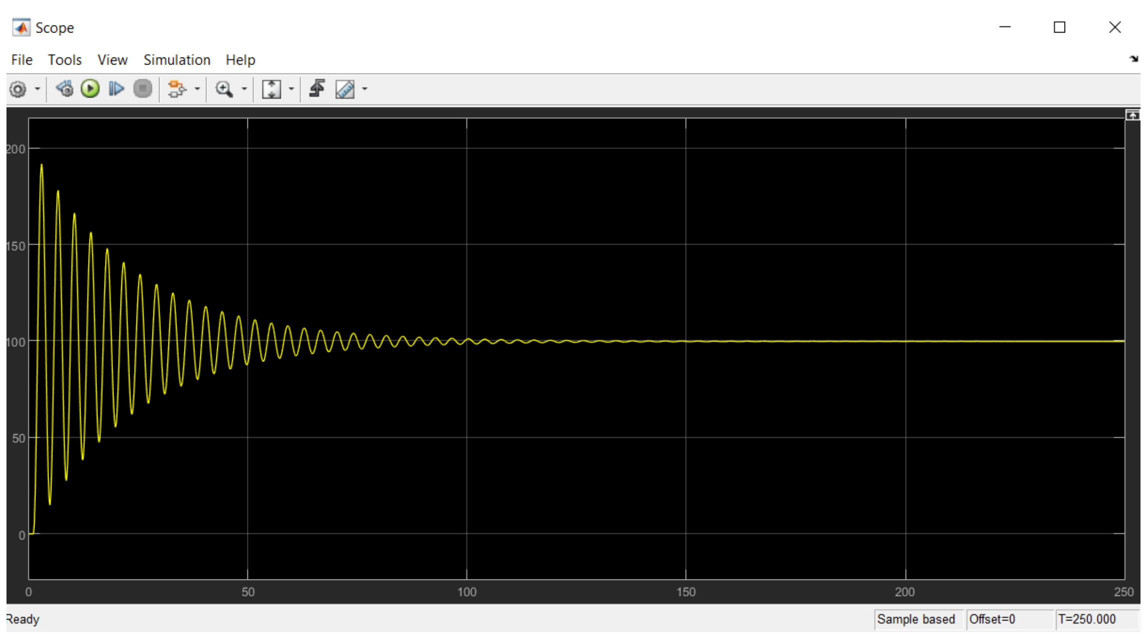
Script

```
%u=-Kx+Nv
%v??
X=inv(A-B*K);
invN=(-C*X)*B;
N=1/invN; %N_trouvé=750
```

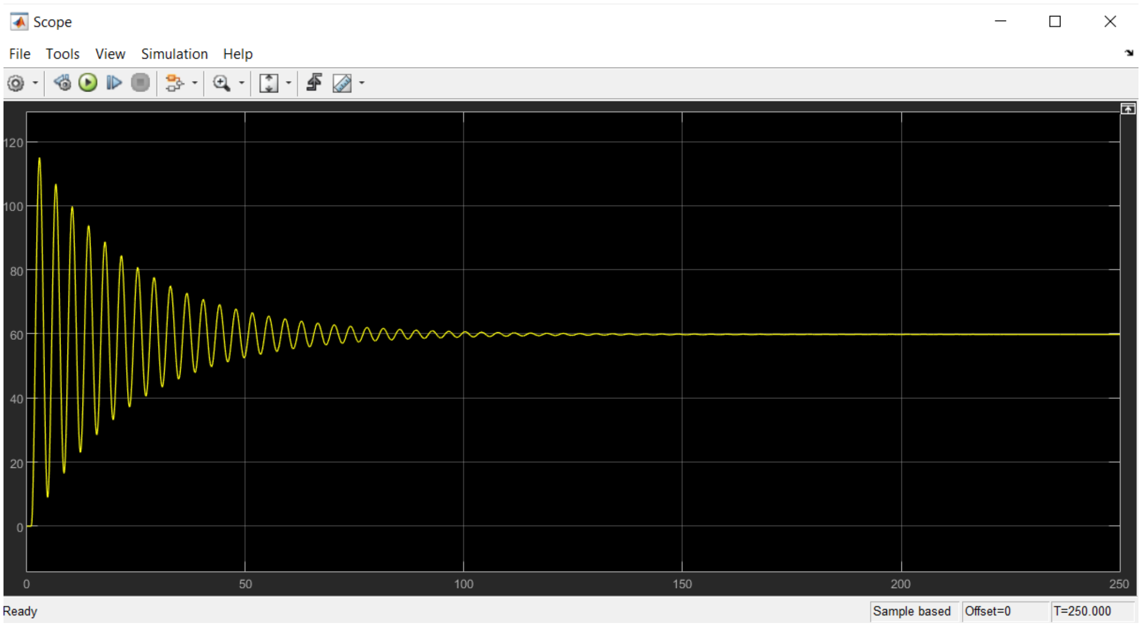
Schéma simulink



La réponse du système à une consigne Yc=100 degrés

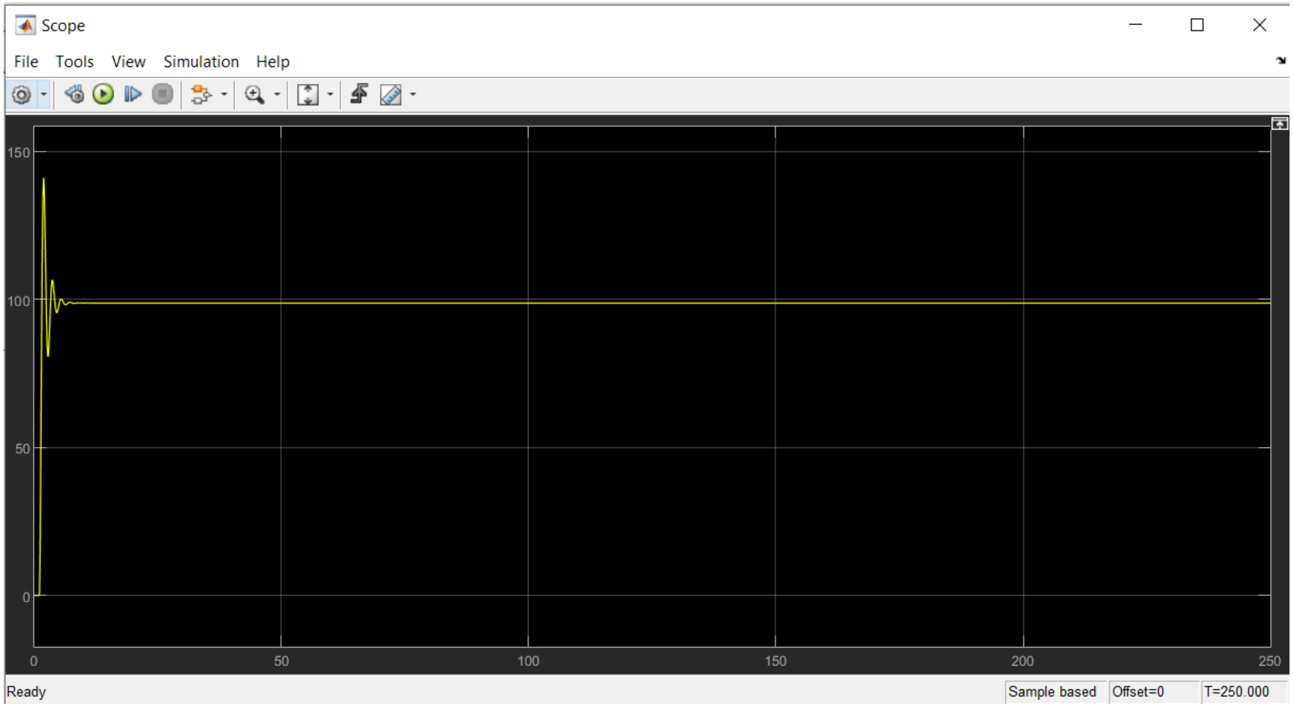


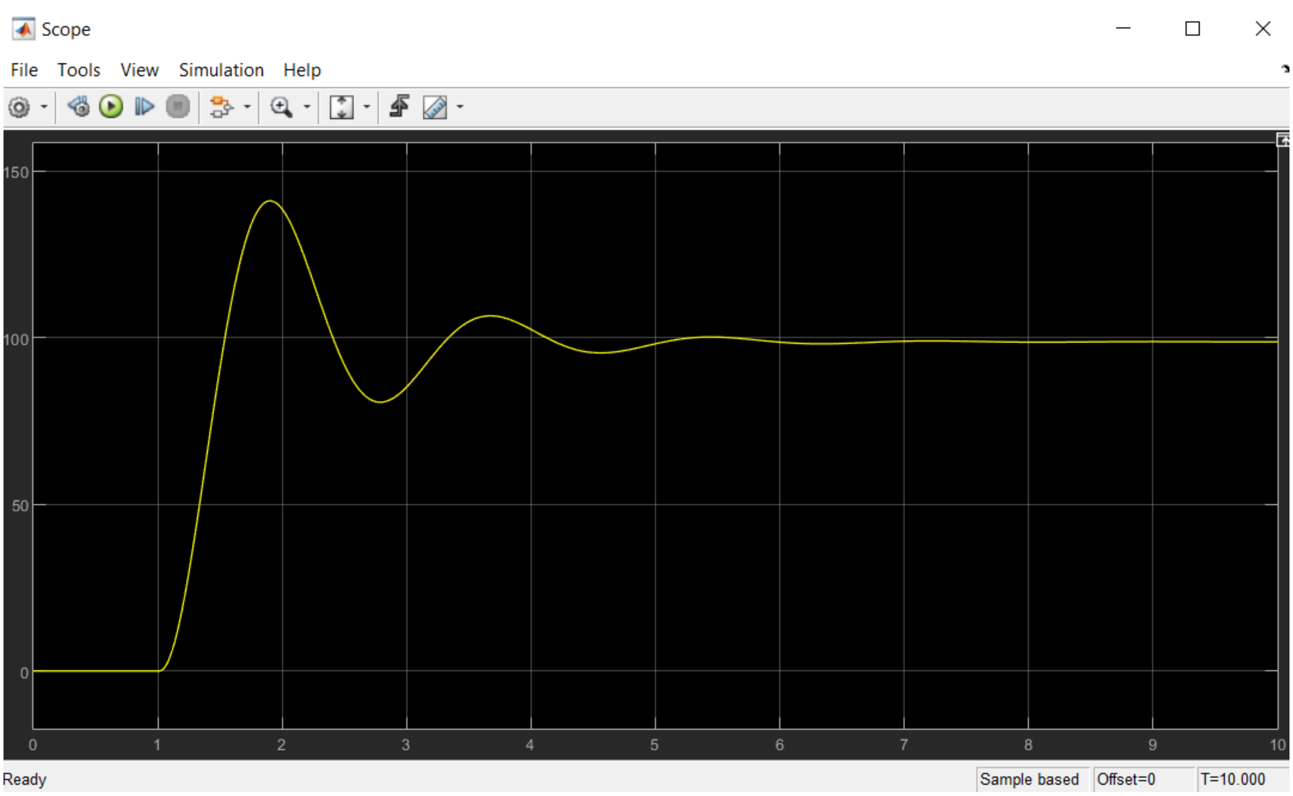
La réponse du système à une consigne Yc=60 degrés



Il est clair que avec ce type de commande nous avons pu commander notre système chaotique qui converge vers la consigne sans erreur statique, et cette convergence est établit aprés un certain temps de convergence qui est à peu prés 90s. Donc nous avons un problème dans la rapidité de convergnce vers la consigne, ce problème était résolu lorsque nous avons choisit les pôle de tel façon que nous aurons le comportement d'un système de deuxième ordre en cascade avec un autre de système de deuxième ordre mais cette fois de pulsation respective $w_1 = 25 rad/set w_2 = 50 rad/s$, donc nous avons augmenté la pulsation pour améliorer le temps d réponse, aprés un calcul simple nous aurons les pôles désirés=[-17.5+17.8i,-17.5-17.8i,-35+35i,-35-35i], et les valeurs des gains et de N obtenus sont :K=1.0e+03 *[9.7093 -5.6728 3.3554 0.5436] et N=1.3065e+04.

La réponse du système à une consigne Yc=100 degrés

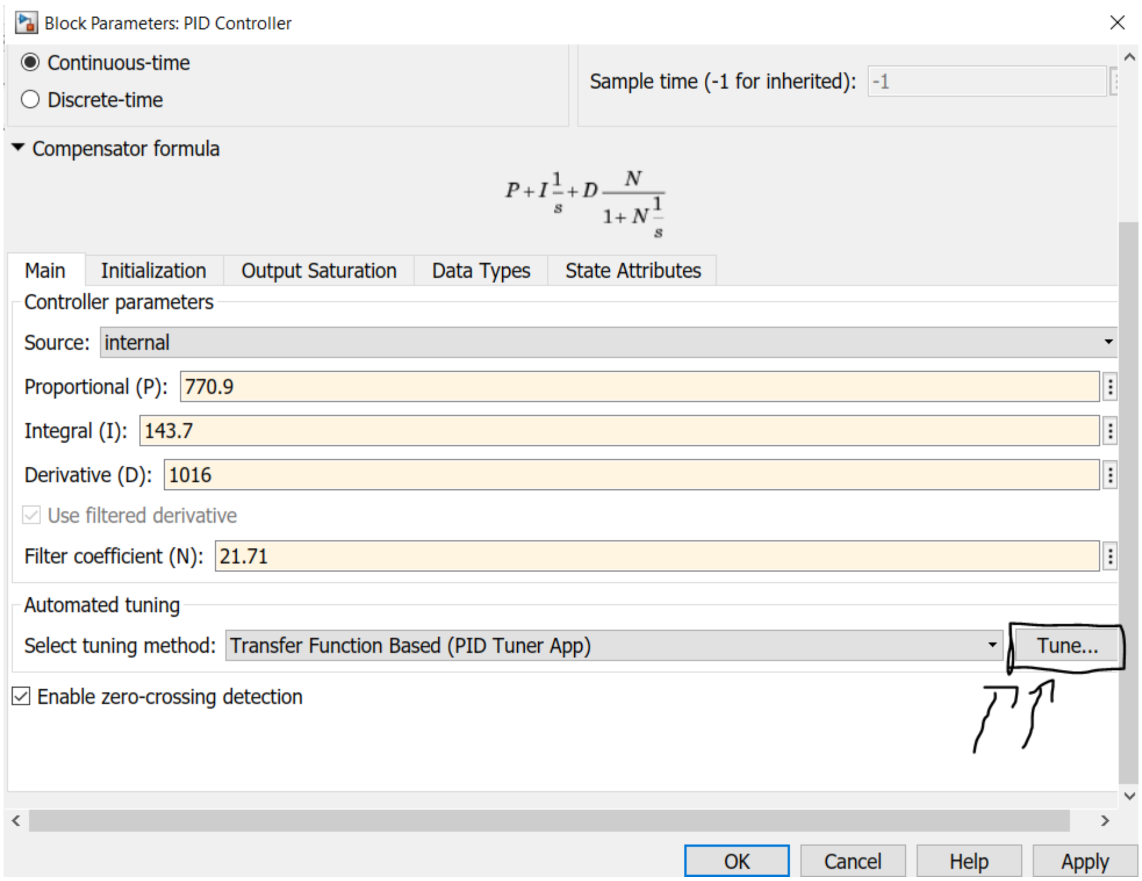


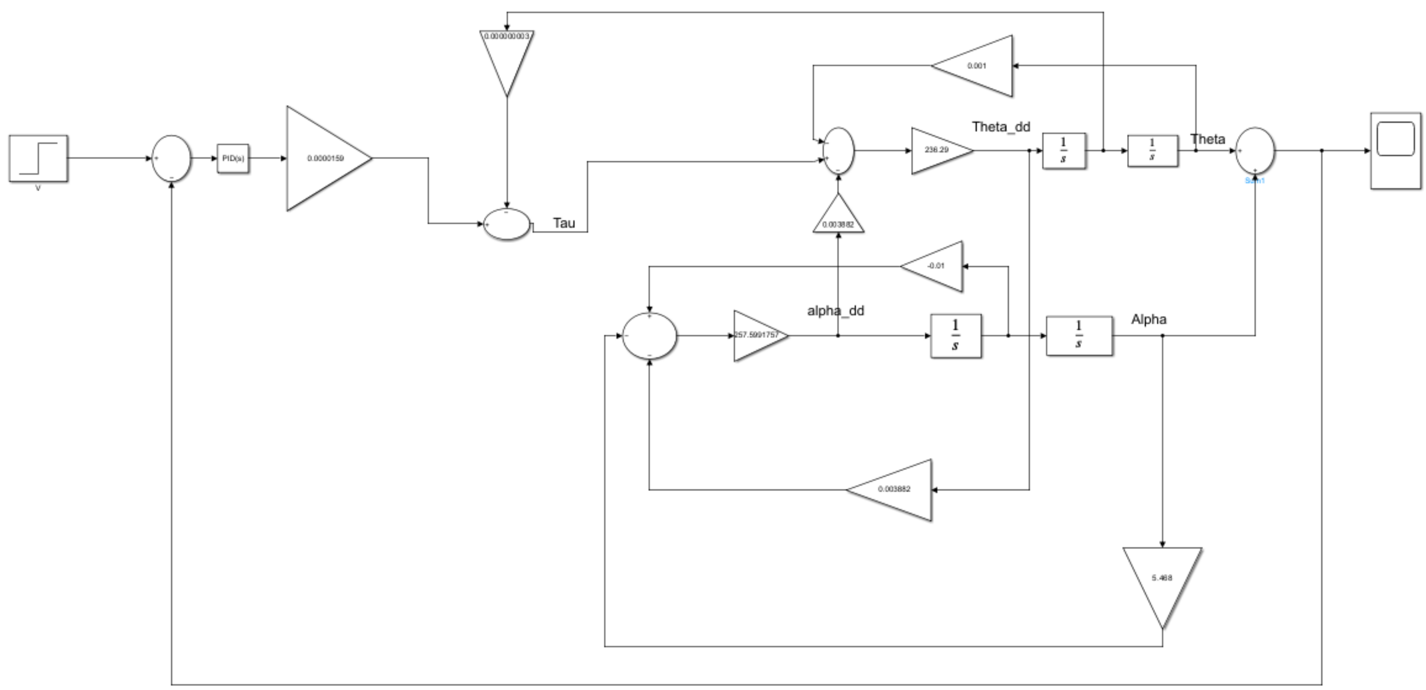


Nous avons même obtenus des réultats meilleurs à celle trouvé avec le correcteur PID(présenté dans la partie suivante).

5.5 Correction avec le correcteur PID

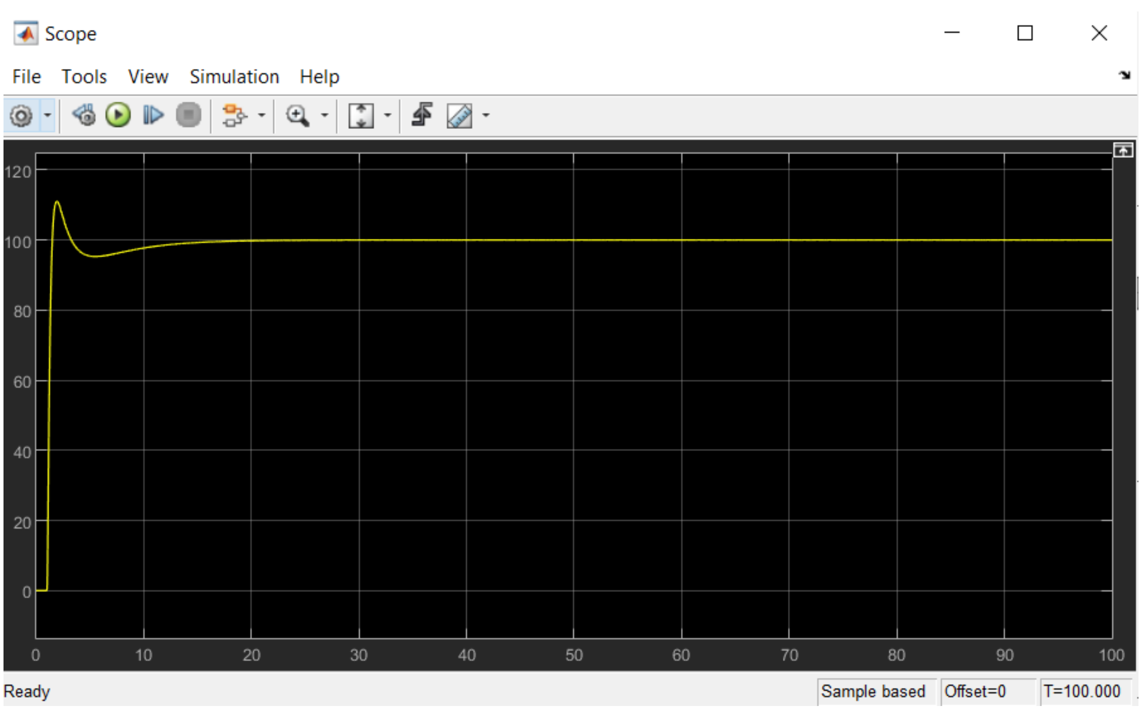
Parmi les techniques de contrôle linéaire, le PID est le plus souvent utilisé dans l'industrie de nos jours en raison de la simplicité et de la facilité en contrôle. Dans le présent travail, des simulations ont été réalisées dans MATLAB/Simulink pour montrer les performances de PID manette. Le bloc PID-Tuner règle les valeurs du PID constantes i.e $K_p, K_i et K_d$, et les valeur que nous avons obtenus sont : $K_p = 770.9, K_i = 143.7, K_d = 1016$.



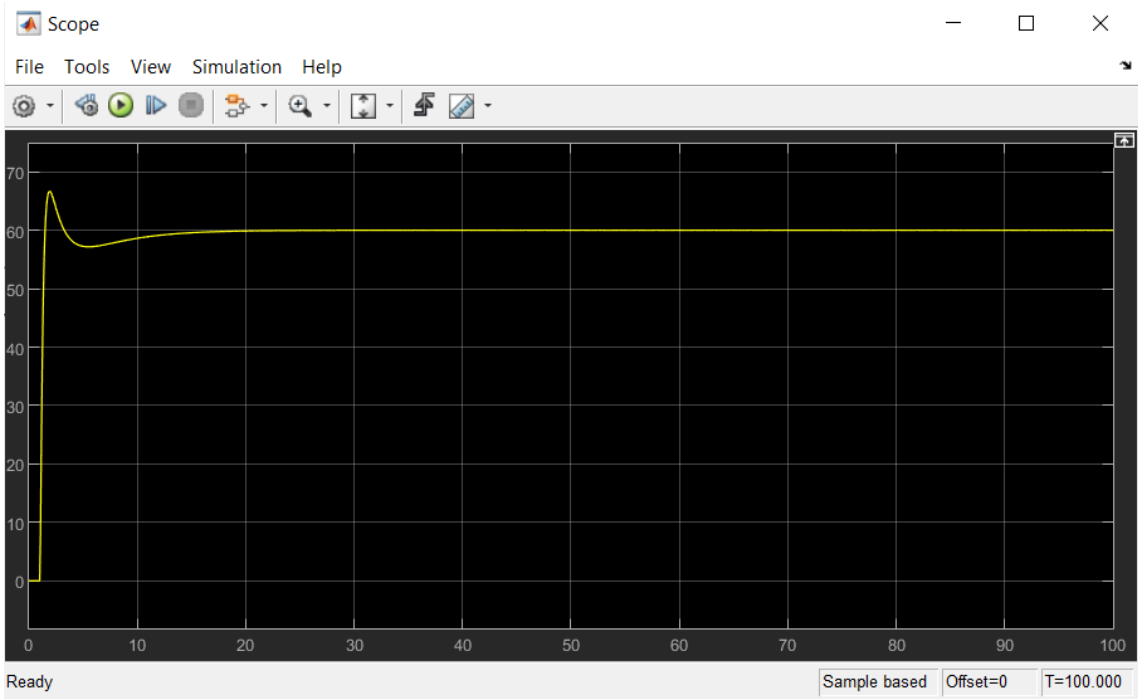


Les figures ci-dessous donnent les résultats de la correction.

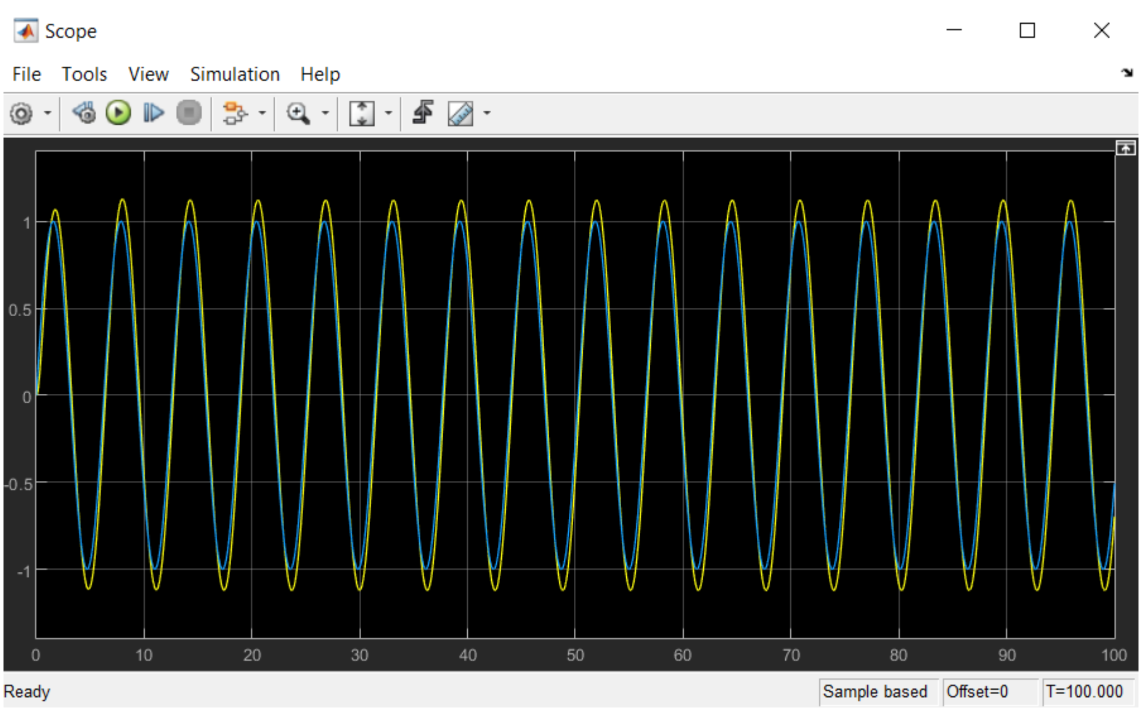
La réponse du système à une consigne Yc=100 degrés



La réponse du système à une consigne Yc=60 degrés



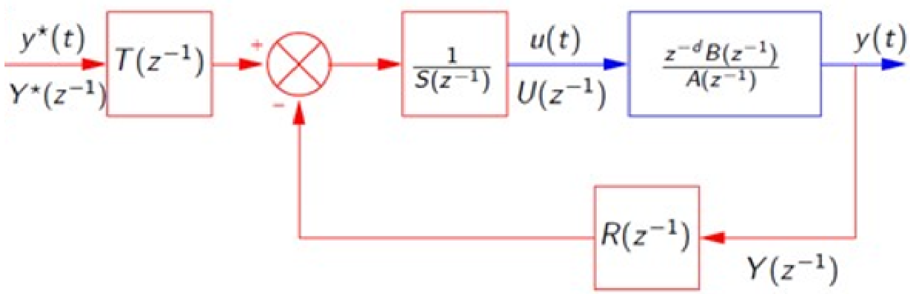
La réponse du système à une consigne sinusoidale(La réponse du système est en jaune)



6 Commande numérique(RST)

Un régulateur RST est un organe de contrôle à trois degrés de liberté permettant d'effectuer une régulation en boucle fermée d'un système industriel. Les régulateurs RST sont couramment utilisés dans les systèmes de commande, le plus souvent numériques, car, dans le contexte monovariable, ce sont les régulateurs qui offrent la plus grande souplesse d'utilisation.

Shéma fonctionnel



6.1 Modèle numérique

A partir de notre repésentation d'état nous avons pu dégager la fonction de transfert continu du notre système, pour ce faire nous avons utilisé le bloc de code suivant:

```
D=0;
[b,a]=ss2tf(A,B,C,D); %Détermination de la fonction de transfert à partir
    du répresentation %d'état
sysc=tf(b,a); %Fonction de transfert continu
```

A partir de la fonction de transfert continu, nous avons pu à l'aide de la méthode c2d déterminer la fonction de transfert discrete(en z), ne effet cette méthode utilise le bloqueur d'ordre zéro pour effectuer cette opération.

Script

```
sysd = c2d(sysc,0.1); %Fonction de transfert discrete
```

Nous avons trouvé les résultats suivants:

```
Command Window

>> sysc

sysc =

             64.07 s^2 + 183.3 s + 119.2
  ---------------------------------------------------
  s^4 - 23.14 s^3 - 1.421e04 s^2 + 4024 s + 5.899e-08

Continuous-time transfer function.

>> sysd

sysd =

          1041 z^3 - 179.7 z^2 - 2049 z + 1230
  ------------------------------------------------------
  z^4 - 4.985e05 z^3 + 1.011e06 z^2 - 5.129e05 z + 10.11

Sample time: 0.1 seconds
Discrete-time transfer function.
```

6.2 Commande RST

6.2.1 Détermination des paramètres de la commande RST

Dans cette partie nous allons déterminer les polynômes en z^{-1}, qui sont $R(z^{-1}) et S(z^{-1})$ de la commande RST. On doit dégager d'abord les expression des polynômes $B(z^{-1}) et A(z^-1)$.

$$\mathrm{H(z^{-1})} = \frac{z^{-1}(1041-179.7z^{-1}-2049z^{-2}+1230z^{-3})}{1-4.985e05z^{-1}+1.011e06z^{-2}-5.129e05z^{-3}+10.11z^{-4}}$$

donc
$\mathrm{B(z^{-1})} = 1041 - 179.7z^{-1} - 2049z^{-2} + 1230z^{-3}$

Et $\mathrm{A(z^{-1})} = 1 - 4.985e05z^{-1} + 1.011e06z^{-2} - 5.129e05z^{-3} + 10.11z^{-4}$

Aprés un simpe calcule on peut démontrer que la fonction caractéristique du système en boucle fermé s'écrit: $P(z^{-1}) = A(z^{-1})S(z^{-1}) + z^-1B(z^{-1})R(z^{-1})(5)$

Les degrés de S et R sont déterminé à l'aide du théorème de Bezout,
$n_S = n_B + d - 1 = 3 + 1 - 1 = 3$ $n_R = n_A + -1 = 4 - 1 = 3$

Nous aurons:
$$S(z^{-1}) = 1 + s_1^{-1} + s_2 z^{-2} + s_3 z^{-3}$$
$$T(z^-1) = R(z^-1) = r_0 + r_1^{-1} + r_2 z^{-2} + r_3 z^{-3}$$

Maintenant nous allons déterminer les coefficients r0,r1,r2,r3 et s1,s2,s3. Nous voulons que notre système admet un comportement de système de deuxième ordre tel que $\zeta = 0.7$ et $\omega = 5$, donc les pôles continus sont: $P_{1,2} = -3.5 \pm 3.75$, or les pôles du système discret s'écrivent $z_{1,2} = \exp(TP_{1,2})$ avec T=0.1s(Période de discretization). Aprés un calcul simple on trouve que:
$$z_1 = 0.652 + i0.2562$$
$$z_2 = 0.652 - i0.2562$$

Et
$$P(z^-1) = 1 - 1.3z^{-1} + 0.48z^{-2}$$

On aura donc

$$(1 - 4.985e05z^{-1} + 1.011e06z^{-2} - 5.129e05z^{-3} + 10.11z^{-4})(1 + s_1 z^{-1} + s_2 z^{-2} + s_3 z^{-3}) +$$
$$z^{-1}(1041 - 179.7z^{-1} - 2049z^{-2} + 1230z^{-3})(r_0 + r_1^{-1} + r_2 z^{-2} + r_3 z^{-3}) = 1 - 1.3z^{-1} + 0.48z^{-2}$$

Aprés un développement et par identification on trouve l'equaton maticielle suivante:
$$EX = P$$

Avec E est une matrice 8*8 E=
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4.85e05 & 1 & 0 & 0 & 1041 & 0 & 0 & 0 \\ 1.011e06 & -4.85e05 & 1 & 0 & -179.7 & 1041 & 0 & 0 \\ -5.1129e05 & 1.011e06 & -4.85e05 & 1 & -2049 & -179.7 & 1041 & 0 \\ 10.11 & -5.1129e05 & 1.011e06 & -4.85e05 & 1230 & -2049 & -179.7 & 1041 \\ 0 & 10.11 & -5.1129e05 & 1.011e06 & 0 & 1230 & -2049 & -179.7 \\ 0 & 0 & 10.11 & -5.1129e05 & 0 & 0 & 1230 & -2049 \\ 0 & 0 & 0 & 10.11 & 0 & 0 & 0 & 1230 \end{pmatrix}$$
$$P = \begin{pmatrix} 1 \\ -1.3 \\ 0.48 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$X = \begin{pmatrix} 1 \\ s_1 \\ s_2 \\ s_3 \\ r_0 \\ r_1 \\ r_2 \\ r_3 \end{pmatrix}$$

Al'aide du script suivant nous avons pu déterminer les coefficients de la commande RST par la résolution du système d'équations linéaire.

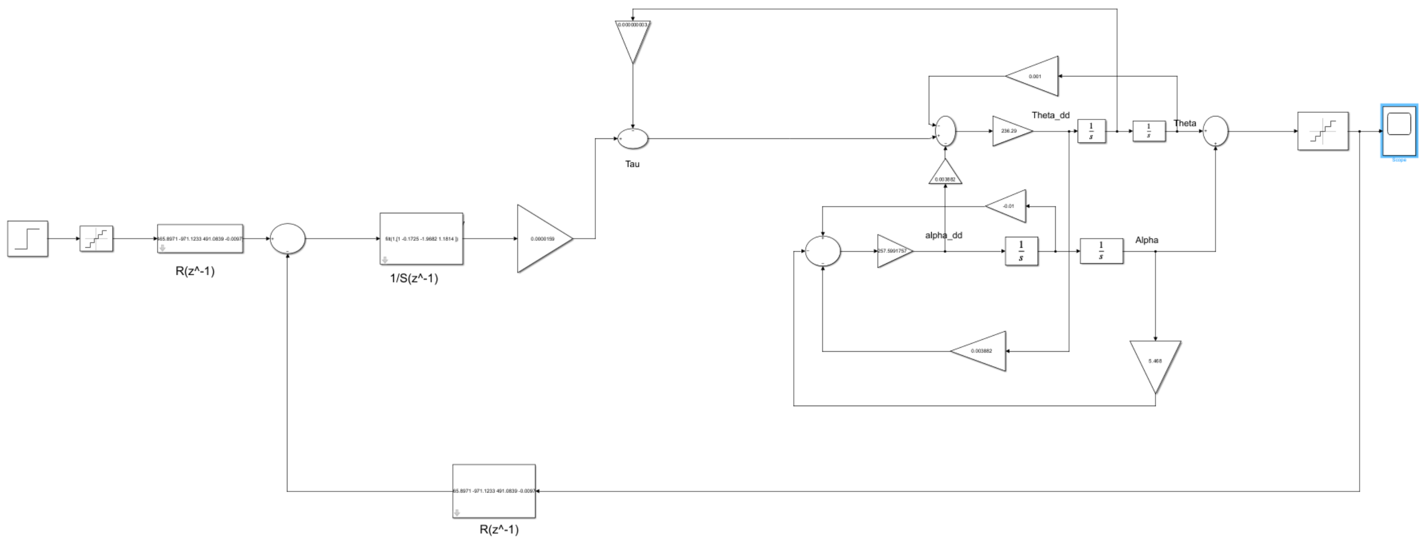
```
%Commande RST
phi=[1         0              0          0      0           0         0        0;
-4.85e05    1              0           0     1041        0           0        0 ;
   1.011e06    -4.85e05       1         0     -179.7     1041        0        0;
-5.1129e05 1.011e06     -4.85e05        1       -2049       -179.7   1041    0;
10.11       -5.1129e05    1.011e06   -4.85e05    1230        -2049  -179.7  1041;
   0           10.11      -5.1129e05  1.011e06    0       1230    -2049   -179.7;
   0           0              10.11   -5.1129e05  0      0        1230     -2049;
   0           0              0          10.11    0       0           0    1230];
P=[1; -1.3; 0.48; 0; 0; 0; 0; 0];
RST=linsolve(phi,P);
```

Le résultat est:

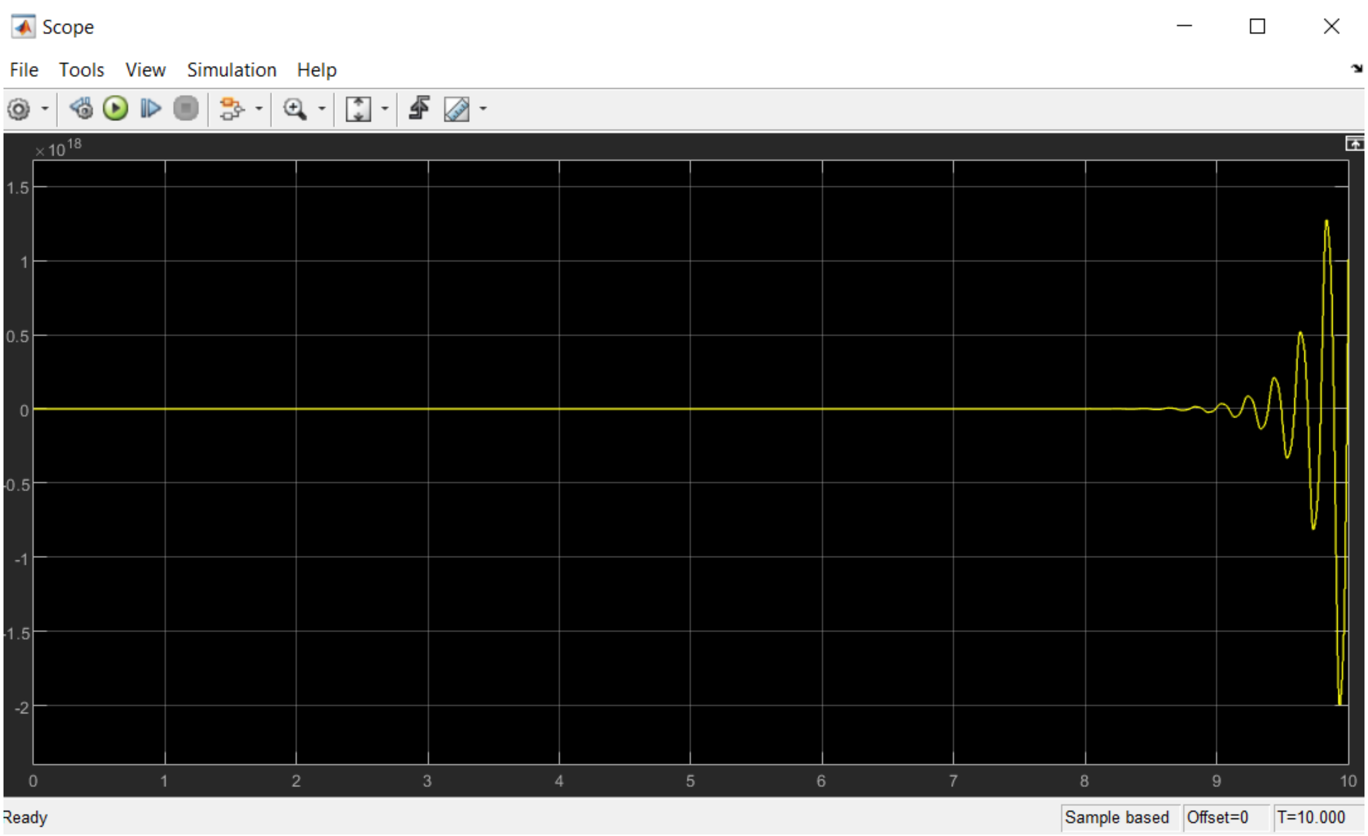
```
>> RST

RST =

    1.0000
   -0.1725
   -1.9682
    1.1814
  465.8971
 -971.1233
  491.0839
   -0.0097
```

Remarque: Dans la simulation d'un système numérique avec simulink, il n'est pas nécessaire d'ajouter un bloc pour le bloqueur d'ordre zéro parce que simulink fait la conversion de discret vers le continu par lui même.

6.2.2 Simulation avec simulink



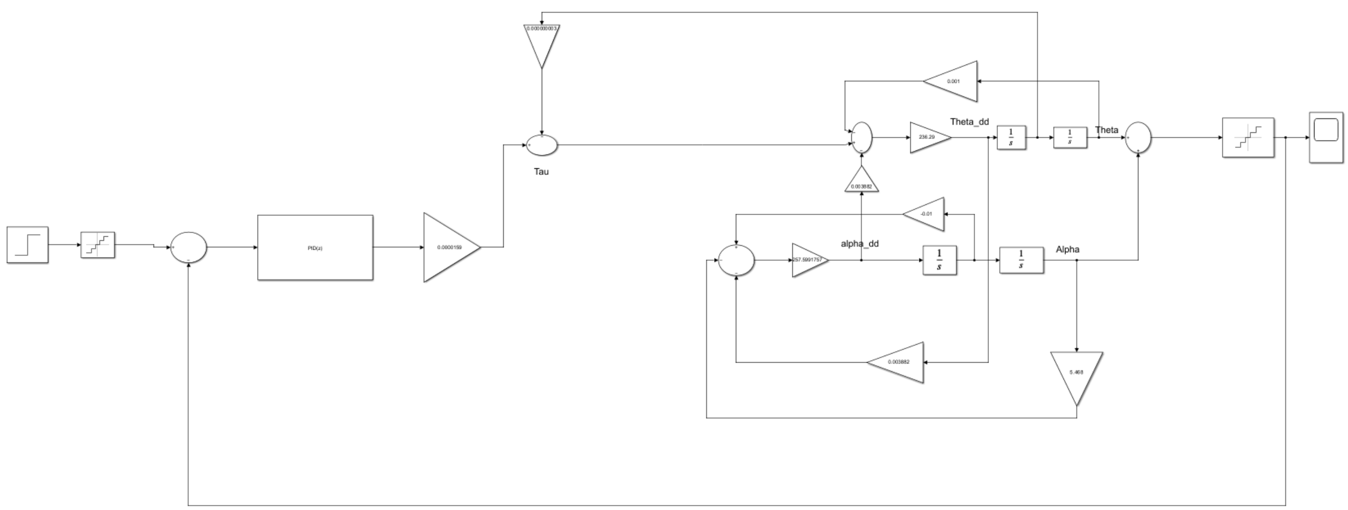
Réponse

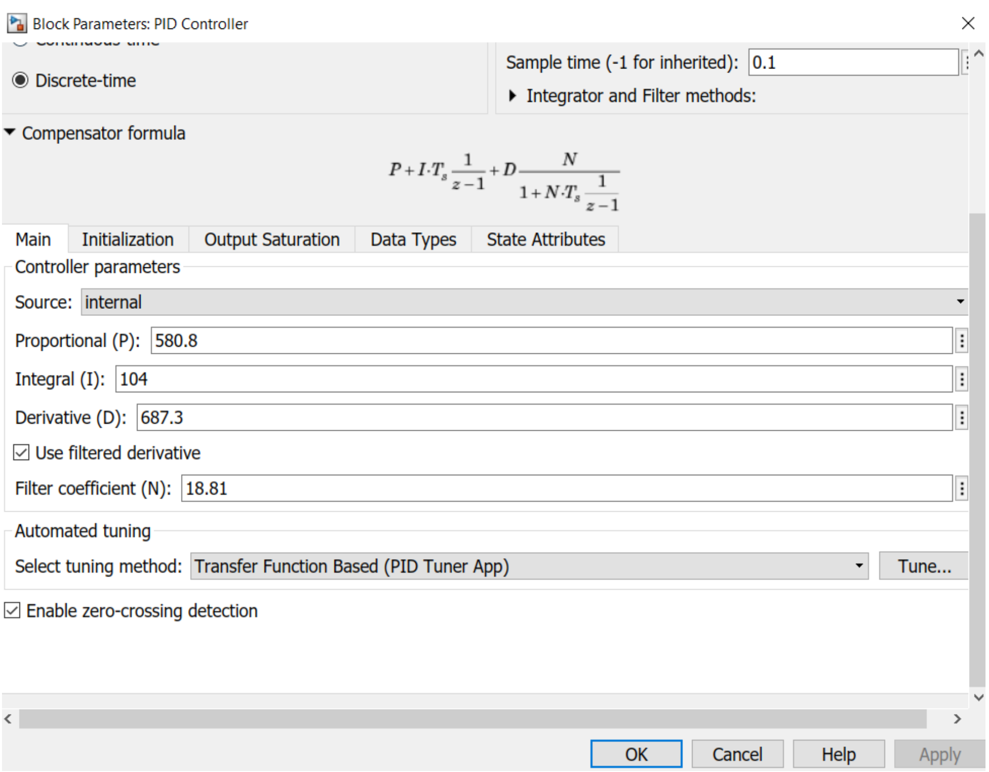


Donc il est clair que la réponse de ce modèle diverge vers l'infini, et ce résultat n'est pas acceptable physiquement, donc il faut rejeter ce modèle et ne faire pas la commande de ce système avec la commande RST.

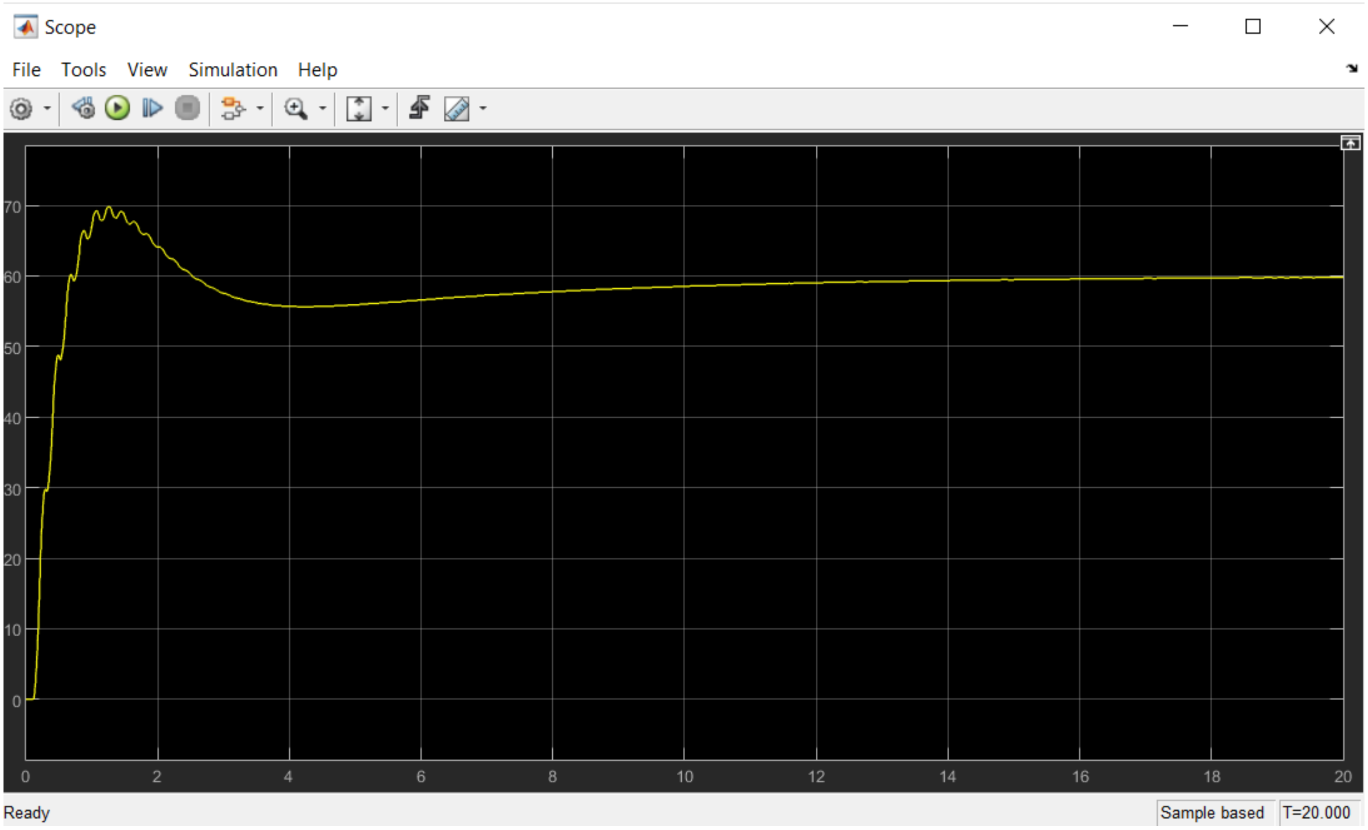
6.3 PID numérique

Simulation avec simulink





La réponse pour une consigne de yc=60 degré



Donc nous avons pu commander notre système numérique avec le correcteur PID numérique.

7 Conclusion

Pour récapituler notre travail, nous avons premièrement simuler le manipulateur robotique à joint flexible avec simulink, nous avons trouvé qu'il à un comporetement chaotique, ainsi ce système n'est pas stable, donc nous avons corriger ce problème avec une commande par retour d'état où nous avons améliorer aussi la rapidité du système, ainsi à l'aide du correcteur PID nous avons trouvé des résultats satisfaisant mais la commande par retour d'état a donné des mailleurs résultat. Maintenant pour la commande numéique, nous avons commencer par la commande RST où nous avons déterminer ses polynômes à l'aide d'une matrice 8*8, mais malheureusement nous avons eu un système qui diverge ce qui est n'est pas acceptable, donc nous avons recouru au correcteur PID numérique afin de corriger ce problème.

8 Références

References

[1] https://www.researchgate.net/publication/329619743$_{C}ontrol_{o}f_{F}lexible_{J}oint_{R}obotic_{M}anipulator$

[2] https://www.quanser.com/products/rotary-flexible-joint/

[3] https://quanserinc.box.com/shared/static/hwzqahebu3ghaveueqede5a6kuq6vf23.pdf