Systèmes Linéaires Invariants 2

Correction - Sujet 1

Ceci est une correction du sujet 1 proposé. Il ne propose pas l'ensemble des explications et remarques possibles. Il représente le niveau attendu dans le cadre du M1ISTR-RODECO, mais avec une précision de rédaction un peu plus poussée que celle d'un examen de travaux pratiques de 2h. Les commentaires supplémentaires seront écrits en rouge. Les diverses critiques concernant la formulation du sujet seront écrits en bleu. On ne rappellera pas ici le sujet. Le script Matlab, les schémas Simulink et les courbes titrées et commentées sont en annexe.

1 PARTIE 1 : RETOUR D'ETAT

Question I.1 - Analyse de stabilité

La matrice dynamique est diagonale supérieure, donc ses valeurs propres sont sur sa diagonale : 0, 3 et -5. L'une d'entre elles est à partie réelle strictement positive donc le modèle linéarisé est asymptotiquement instable.
Ceci est vérifié avec Matlab par le calcul des valeurs propres. De plus, la réponse à une condition initiale arbitraire diverge (figure 1).
Quant à la stabilité entrée bornée / sortie bornée, on l'étudiera après l'analyse de commandabilité et d'observabilité.

Question I.2 - Analyse de commandabilité et observabilité

La matrice de commandabilité est de rang 3, égal à la dimension de l'état. Le modèle est donc commandable. Donc il est possible de choisir toutes les valeurs propres de ce modèle asservi par retour d'état.
La matrice d'observabilité est de rang 3, égal à la dimension de l'état. Le modèle est donc observable. Donc lors de la création d'un observateur il est possible de choisir toutes les valeurs propres de la matrice dynamique de l'erreur.
Le modèle étant commandable et observable, toutes les valeurs propres de la matrice dynamique seront racines du dénominateur de la fonction de transfert entre $U(p)$ et $Y(p)$. Donc la conclusion sur la stabilité entrée bornée / sortie bornée est identique : le modèle est instable.

Question I.3 - Choix des valeurs propres du modèle asservi par retour d'état

Erreur statique nulle : aucun incidence sur le choix des valeurs propres.
Temps de réponse de 2 s : on choisit une dynamique dominante du premier ordre (p_1 réel négatif, p_2 et p_3 associés à des modes plus rapides), auquel cas le temps de réponse est de l'ordre de grandeur de $-3/p_1$. En cas de problème, il faudra de plus veiller à ce que les éventuels zéros soient à partie réelle bien plus négative que p_1

pour qu'ils ne dégradent pas la dynamique dominante.
La commande ne doit pas excéder 3 : spécification impossible à réaliser dans l'absolu car, par homogénéité, si la consigne et la condition initiale est multipliée par α, le signal de commande sera aussi multiplié par α.

BILAN : On propose $p_1 = -1.8$, $p_2 = -5$ et $p_3 = -10$.

Question I.4 - Calcul du gain du retour d'état (et du gain de précompensation)

Avec Matlab on trouve $K = [0.0643 \quad 2.0118 \quad 13.6654]$ et la vérification confirme que $A - B * K$ possède les bonnes valeurs propres désirées.
Pour assurer la précision demandée, on calcule aussi le gain de précompensation. On trouve $N_{\texttt{precomp}} = 0.0643$

Question I.5 - Schéma de l'asservissement

(figure 2) La simulation sous Matlab du système asservi avec gain de précompensation (entre consigne z_c échelon unitaire et sortie de performance z, conditions initiales nulles) confirme les deux premiers objectifs. Les zéros de l'asservissement ont effectivement une partie réelle suffisamment négative pour que la dynamique dominante soit du premier ordre.
(figure 3) La simulation sous Matlab de ce même système (entre consigne z_c échelon unitaire et sortie commande u, conditions initiales nulles) confirme le troisième objectif, en supposant que celui-ci soit explicité pour ces conditions expérimentales.
(figure 4) Les résultats sont confirmés par les simulations sous Simulink.

2 PARTIE 2 : OBSERVATEUR MINIMAL IDENTITE

Question II.1 - Équations d'un observateur minimal identité

Les équations n'étant pas à connaître, leur démonstration qui n'est pas demandée dans le sujet est donc exigible. En voici une :
On constate que la matrice de mesure est déjà la première ligne d'une matrice identité. On n'a donc pas besoin de faire un changement de base $x = Pw$ pour obtenir $y = [I \quad O]w$. Pour réaliser l'observateur minimal identité demandé, on découpe par bloc les matrices A et B de sorte à obtenir :

$$\begin{aligned} \begin{pmatrix} \dot{x_1} \\ \dot{x_2} \end{pmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y &= x_1 \end{aligned}$$

Pour un observateur minimal, la sortie $\hat{x_2}$ de l'observateur doit converger vers x_2. Si on nomme z l'état de ce système dynamique d'entrées x_1 et u, son équation de sortie s'écrit donc :

$$\hat{x_2} = Mz + Nx_1 + Qu$$

L'observateur étant de plus identité, il vient que $M = I$. On choisit $Q = O$ et N reste à déterminer. L'erreur d'observation s'écrit alors $\epsilon = \hat{x_2} - x_2 = Nx_1 - Ix_2 + Iz$. Pour la matrice N à déterminer et une matrice dynamique de l'erreur F dont on choisit les valeurs propres, les équations fondamentales de l'observateur minimal identité sont donc :

$$\begin{aligned}\dot{\epsilon} &= F\epsilon \\ z &= [-N \quad I]x + \epsilon\end{aligned}$$

On développe l'équation dynamique de l'observateur en dérivant l'expression précédente :

$$\begin{aligned}\dot{z} &= -N\dot{x_1} + \dot{x_2} + \dot{\epsilon} \\ \dot{z} &= -N(A_{11}x_1 + A_{12}x_2 + B_1u) + (A_{21}x_1 + A_{22}x_2 + B_2u) + F(z + Nx_1 - x_2) \\ \dot{z} &= Fz + (-NA_{11} + A_{21} + FN)y + (-NA_{12} + A_{22} - F)x_2 + (-NB_1 + B_2)u\end{aligned}$$

Par identification avec l'équation dynamique générique $\dot{z} = Fz + Gy + Hu$, on obtient les équations de l'observateur suivantes :

$$\begin{aligned}F &= A_{22} - NA_{12} \\ G &= FN + A_{21} - NA_{11} \\ H &= B_2 - NB_1\end{aligned}$$

On remarque que les valeurs propres de F peuvent être choisies librement si et seulement si la paire A_{22}, A_{12} est observable.

Question II.2 - Réglage de l'observateur minimal identité

Pour obtenir une convergence de l'erreur en 3 secondes, on propose de régler les valeurs propres de l'observateur de sorte que chaque mode soit apériodique et de constante de temps inférieure à 1 seconde. On propose par exemple -2 et -2.
On trouve $N = [0.0897 \quad 0.0066]^T$.

Question II.3 - Vérifications par simulation

(figure 5) On constate la convergence de l'erreur vers 0 avec les modes demandés pour différentes conditions initiales de l'observateur. Le procédé observé, en revanche, reçoit un signal de consigne nul et a des conditions initiales nulles. Ceci ne nous permet de valider que les valeurs propres de F.

Question II.4 - Dynamique autonome de l'erreur

La tournure de la question est incorrecte, un observateur n'est pas autonome, seule la dynamique de son erreur l'est. Pour valider l'observateur dans son ensemble, il est nécessaire de prouver que la dynamique de l'erreur est bien autonome, c'est à dire que ni y ni u ne modifient le comportement de u.

Par superposition, on donne donc des conditions initiales nulles à l'observateur et au procédé et on place une commande u arbitraire. À chaque expérience (forme de signal, amplitude) l'erreur reste nulle (aux erreurs numériques de Simulink et du solveur près).
On donne ensuite des conditions initiales arbitraires non nulles à l'observateur et au procédé, ainsi qu'une commande u. Le signal d'erreur continue de converger vers 0 avec la dynamique désirée. Ces expériences n'invalident donc pas l'observateur et il est légitime de penser que la dynamique de l'erreur est bien autonome et celle qui est désirée.

3 PARTIE 3 : RETOUR D'ETAT ET OBSERVATEUR

Question III.1 - Premier test du retour d'état basé observateur

(figure 6) On visualise ici la dynamique de l'erreur et le comportement de la sortie de performance pour une consigne échelon et des conditions initiales de l'observateur égales à $[0.1; -0.1]$. Par superposition, la réponse de la sortie de performance est constituée de la somme entre sa réponse à la consigne seule (l'allure souhaitée par retour d'état) et sa réponse aux conditions de l'observateur.
On constate ici que le comportement obtenu ne correspond pas au comportement souhaité et que les deux premiers objectifs de commande ne sont pas atteints. C'est forcément dû à la réponse aux conditions initiales de l'observateur.
Parmi les solutions possibles pour régler ce problème, on pourrait proposer d'avoir des conditions initiales plus faibles (plus réalistes), on pourrait laisser converger l'observateur avant que commence l'échelon de consigne, on pourrait aussi choisir une dynamique d'observateur un peu plus rapide pour que la perturbation qu'il ajoute au transfert souhaité soit de courte durée. Ces trois propositions ont pour objectif d'avoir une plus faible amplitude de perturbation. (La question est vague, je pense que vous n'attendiez que le réglage de rapidité.)

Question III.2 - Amélioration de l'observateur

(figure 7) On choisit par exemple de rendre l'observateur 4 fois plus rapide (valeurs propres -8 et -8) avec les mêmes conditions initiales $[0.1; -0.1]$ et d'attendre 5 secondes avant de démarrer l'échelon de consigne, pour que l'observateur converge et que le procédé retrouve une condition initiale nulle.
Le résultat est cohérent avec l'analyse préalable. La convergence de l'erreur est plus rapide et laisse le temps au procédé asservi de converger en environ 5 secondes. Alors la réponse à l'échelon de consigne retardé est identique à la réponse obtenue dans la première partie. En revanche, on n'a pas explicitement amélioré (diminué) les conditions initiales de l'observateur, et on constate que leur effet sur le comportement du procédé avant l'échelon est très important.

Question III.3 - Système composite

On décompose aussi K par bloc en $K = [K_1 \quad K_2]$. L'équation du retour d'état est alors :

$$\begin{aligned} u &= N_{\texttt{precomp}} y_c - K_1 x_1 - K_2(z + N x_1) \\ &= N_{\texttt{precomp}} y_c - K x - K_2 \epsilon \\ \dot{x} &= A x + B N_{\texttt{precomp}} y_c - B K x - B K_2 \epsilon \end{aligned}$$

Le système composite a donc l'équation dynamique suivante :

$$\begin{pmatrix} \dot{x} \\ \dot{\epsilon} \end{pmatrix} = \begin{bmatrix} A - BK & -BK_2 \\ O & F \end{bmatrix} \begin{pmatrix} x \\ \epsilon \end{pmatrix} + \begin{bmatrix} BN_{\texttt{precomp}} \\ 0 \end{bmatrix} y_c$$

(figures 8 et 9) Pour confirmer, on trace avec Matlab la réponse indicielle et la réponse aux conditions initiales $[0; 0; 0; 0.1; -0.1]$.

Question III.4 - Commandabilité et observabilité

La matrice de commandabilité est de rang 3, égal à la dimension de x seulement. Le modèle est donc non commandable. C'est cohérent dans la mesure où l'on a rendu la dynamique de l'erreur de l'observateur non commandable.
La matrice d'observabilité est de rang 5, égal à la dimension de l'état total. Le modèle est donc observable. On observe donc l'effet de l'erreur d'observation sur le comportement de la sortie de performance, ce qui est cohérent avec les précédentes expériences.

Question III.5 - Sur le système non-linéaire

Je me permets de retourner cette question avec les choix suivants :

- Le modèle est faux
- Le système est cassé
- L'analyse est fausse
- Le modèle est inadapté
- Je n'ai pas eu de chance
- C'est un problème insurmontable
- Ce n'est pas une science exacte
- Une des hypothèses est fausse
- Le modèle est non valide
- Il y a eu une interférence interne
- Il y a eu une interférence externe
- J'ai fait une erreur de manipulation

– Ma manipulation est trop imprécise
– C'est normal, la théorie n'est pas la pratique
– C'est inexplicable
– Mes hypothèses sont toutes fausses
– Le système a changé

ANNEXE : COURBES ET SCHEMAS

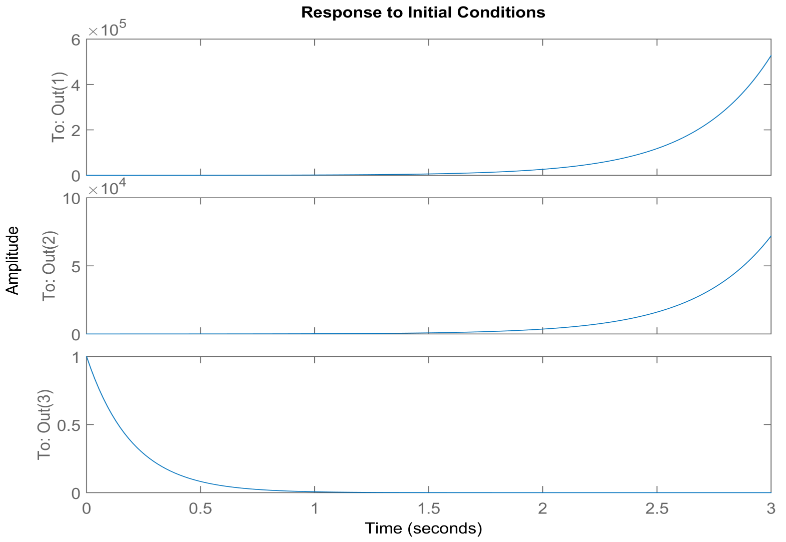

FIG. 1 – *I.1. Réponse x du modèle linéarisé à une condition initiale* $[1; 1; 1]$ *seulement*

On remarque qu'au moins une variable d'état diverge, ce qui est caractéristique d'une instabilité asymptotique.

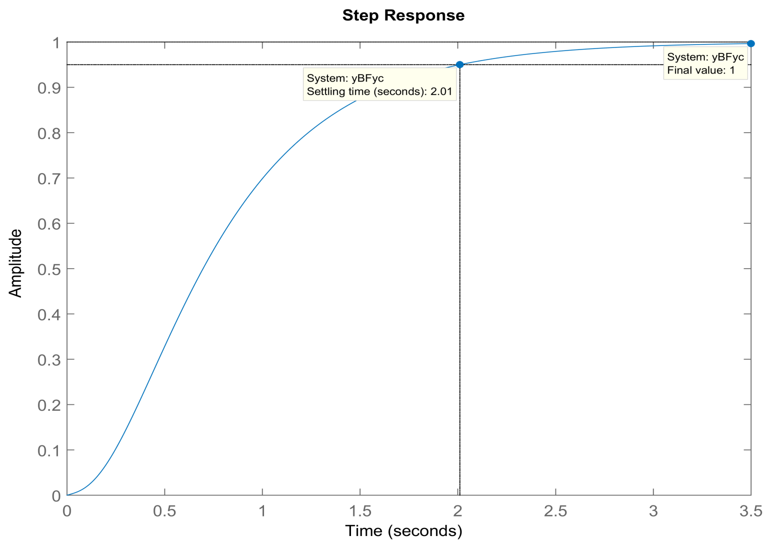

FIG. 2 – *I.5. Réponse y du modèle asservi par retour d'état, consigne échelon unitaire, CI nulles*

Les deux mesures confirment les deux premiers objectif demandés.

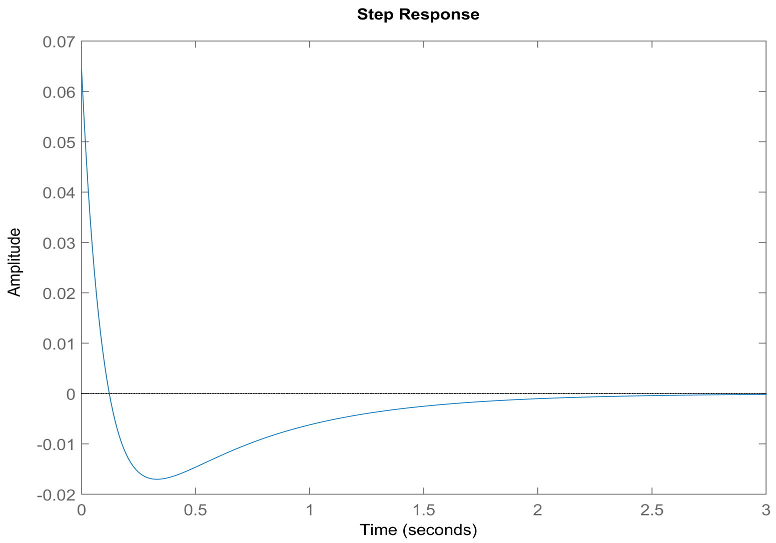

FIG. 3 – *I.5. Réponse u du modèle asservi par retour d'état, consigne échelon unitaire, CI nulles*

L'amplitude de u est bien inférieure à 3.

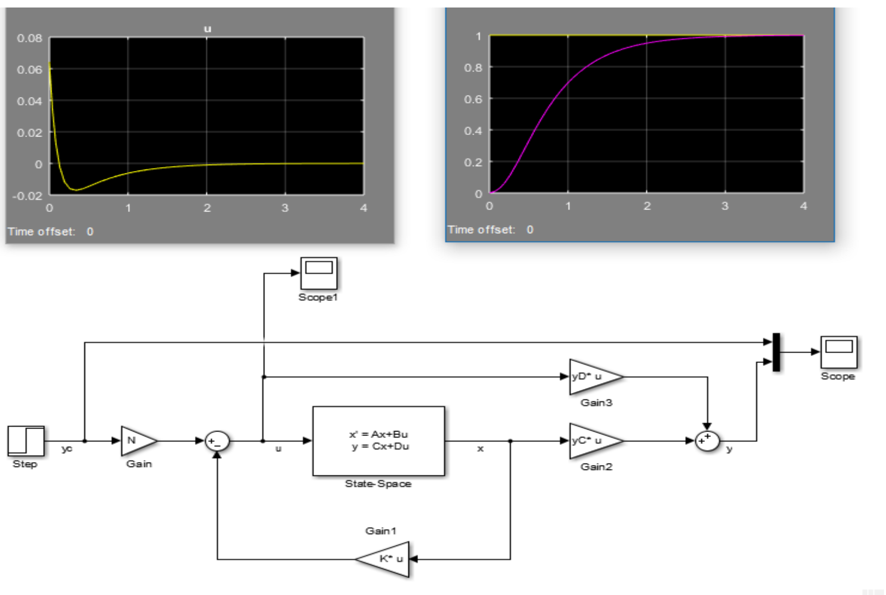

FIG. 4 – *I.5. Schéma Simulink du retour d'état et résultats de simulations*

Les résultats confirment ceux de Matlab. (désolé pour les courbes sur fond noir, ça sera corrigé par la suite)

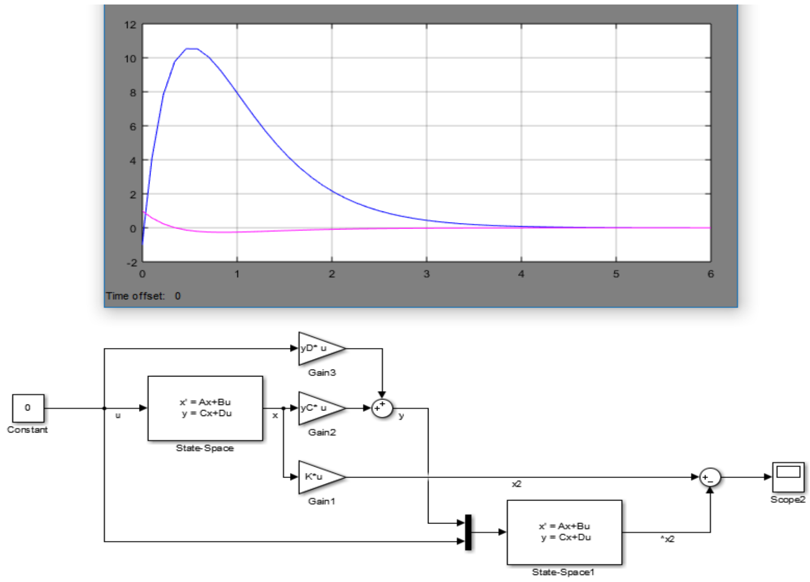

FIG. 5 – *II.3. Schéma Simulink de l'observateur en boucle ouverte et résultats de simulations*

Le résultat de simulation est pour une condition initiale arbitraire de $[1; -1]$. Le comportement obtenu de l'erreur est cohérent avec les modes demandés et satisfait l'objectif de rapidité.

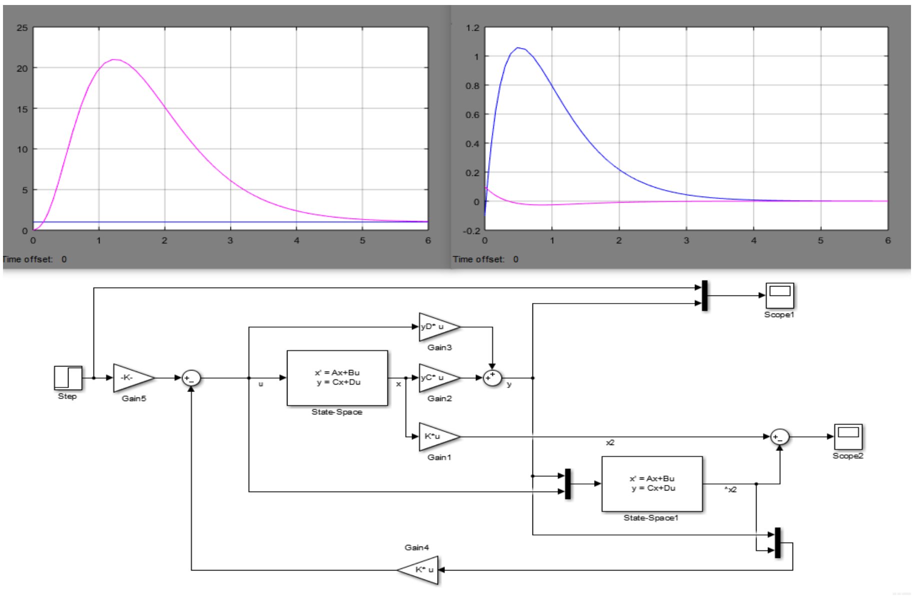

FIG. 6 – *III.1. Premier schéma Simulink du retour d'état basé observateur, condition initiales de l'observateur* $[0.1; -0.1]$

La dynamique de convergence de l'erreur de l'observateur est bien conservée. En revanche, la réponse du système bouclé n'est pas satisfaisante, dans la mesure où on ne retrouve pas vraiment la dynamique choisie par retour d'état.

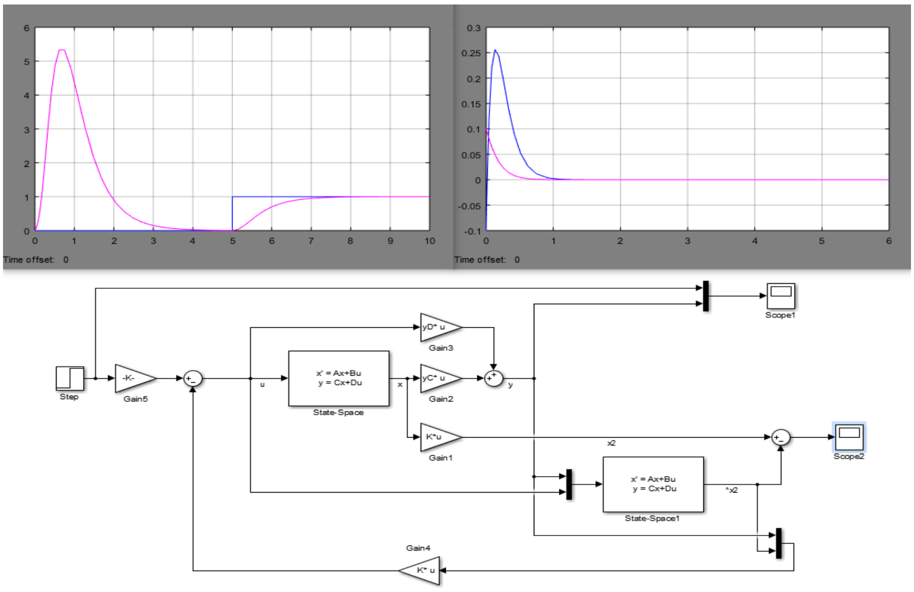

FIG. 7 – *III.2. Second schéma Simulink du retour d'état basé observateur, échelon retardé, condition initiales de l'observateur* $[0.1; -0.1]$

On constate effectivement que le retard de l'échelon donne le temps au procédé de répondre aux conditions initiales de l'observateur et de retrouver des conditions initiales nulles. On retrouve ensuite l'allure désirée de la réponse à la consigne.

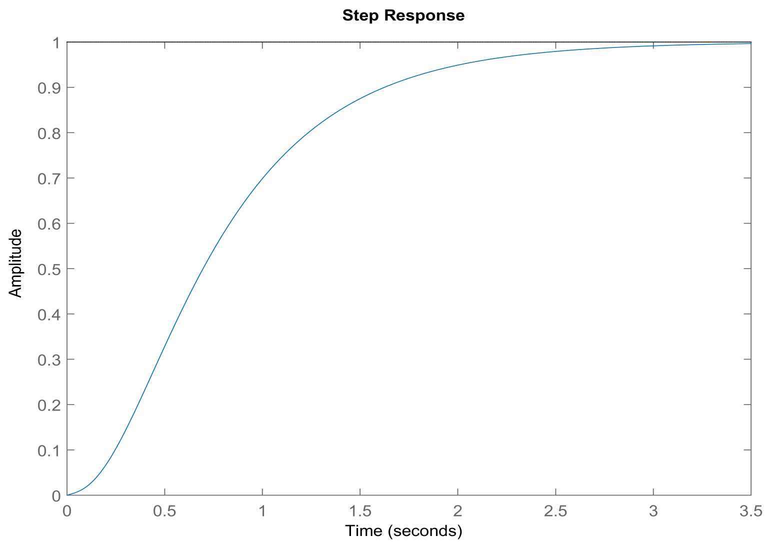

FIG. 8 – *III.3. Réponse indicielle système composite*

Le comportement correspond au comportement désiré et à celui de la simulation précédente après 5 secondes.

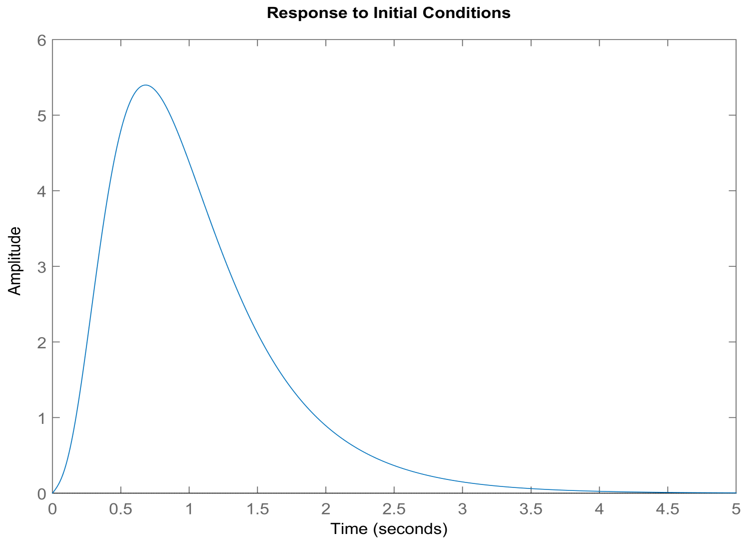

FIG. 9 – *III.3. Réponse aux conditions initiales du système composite*

Le comportement correspond à celui de la simulation précédente avant 5 secondes.

ANNEXE : SCRIPT MATLAB

```
clear all
close all

%% Données du problème

% Equation dynamique
A=[0 22 4;0 3 63;0 0 -5];
B=[2;0.5;1];

% Equation de la sortie état x
xC=eye(3);
xD=zeros(3,1);
% Equation de la mesure y
yC=[1 0 0];
yD=[0];
% Equation de la sortie de performance z=y
zC=[1 0 0];
zD=[0];

Traces = 0;

%% Partie 1
```

```
% Q1 : vérification
eig(A)
xBOu = ss(A,B,xC,xD);
% Tracé de la réponse à des CI
if(Traces)
figure(1)
initial(xBOu,[1;1;1],3);
end

% Q2 : commandabilité et observabilité
Com = ctrb(A,B);
rank(Com)
Obs = obsv(A,yC);
rank(Obs)
yBOu = ss(A,B,yC,yD);
pole(tf(yBOu))

% Q3 : choix des valeurs propres de l'asservissement

Pdes = [-1.8 -5 -10];

% Q4 : calcul de K

K = acker(A,B,Pdes)
eig(A-B*K) % pour vérifier
yBFv = yC*feedback(xBOu,K)+yD;
Nprecomp = 1/dcgain(yBFv)
yBFyc = yBFv*Nprecomp;

% Q5 : simulation du retour d'état

if(Traces)
figure(2)
step(yBFyc);
end
zero(yBFyc)

uBFv = feedback(1,K*xBOu)*Nprecomp;
if(Traces)
figure(3)
```

```
step(uBFv);
end

%% Partie 2

% Q1 : Préparation de l'observateur minimal identité

% Pas de changement de base nécessaire pour x car C est déjà la première
% ligne d'une identité

A11 = A(1,1);
A12 = A(1,2:3);
A21 = A(2:3,1);
A22 = A(2:3,2:3);
B1 = B(1);
B2 = B(2:3);

% Q2 : Réglage de l'observateur

PobsDes = [-2 -2];
N = acker(A22',A12',PobsDes)'
F = A22 - N*A12;
G = F*N + A21 - N*A11;
H = B2 - N*B1;
M = eye(2);
eig(F)

% Q3 : Vérifications expérimentales

% Si on souhaite le tester sous Matlab seulement :
% x2Obsyu = ss(F,[G H],M,[N zeros(2,1)]);

%% Partie 3

% Q2 Amélioration de l'observateur

PobsDes = [-8 -8];
N = acker(A22',A12',PobsDes)'
F = A22 - N*A12;
G = F*N + A21 - N*A11;
H = B2 - N*B1;
```

```
M = eye(2);
eig(F)

% Q3 Système composite

Atotal = [A-B*K -B*K(1,2:3);zeros(2,3) F];
Btotal = [B*Nprecomp;zeros(2,1)];
Ctotal = [yC zeros(1,2)];
Dtotal = 0;

yBFTOTALyc = ss(Atotal,Btotal,Ctotal,Dtotal);
if(Traces)
figure(4)
step(yBFTOTALyc);
figure(5)
initial(yBFTOTALyc,[0;0;0;0.1;-0.1],5);
end

% Q4 Observabilité et Commandabilité

Comtotal = ctrb(Atotal,Btotal);
rank(Comtotal)
Obstotal = obsv(Atotal,Ctotal);
rank(Obstotal)
```