Compte rendu de TP

Identification du modèle d'un système par la méthode des moindres carrés

Fleytoux Yoann , Aurélien Bernier Levalois

11 octobre 2016

Table des matières

1 Présentation du TP **1**

2 Identification d'une fonction de transfert à pôles et zéros inconnus **1**
2.1 Travail préparatoire 2
2.2 Mise en oeuvre sous Matlab 5

1 Présentation du TP

L'objectif de ce TP est d'utiliser la méthode des moindres carrés, dans le contexte de l'automatique. Il s'agira ici d'identifier les paramètres du modèle d'un système dynamique donné. Pour cela, on sollicitera ce système avec une entrée connue et on observera ses sorties. Après avoir modélisé le problème, on appliquera la méthode des moindres carrés pour retrouver la meilleure valeur des paramètres compte tenu des entrées et sorties mesurées.

2 Identification d'une fonction de transfert à pôles et zéros inconnus

On considère ici un système dynamique préalablement modélisé. Cette modélisation a permis d'établir que ce système est d'ordre 3. Il peut donc être caractérisé par la fonction de transfert ci-après :

$H(p) = Y(p)/U(p) = N(p)/D(p) = (ap^3 + bp^2 + cp + d)/(p^3 + \alpha p^2 + \beta p + \gamma)$

2.1 Travail préparatoire

Sachant que l'on peut identifier directement une fonction de transfert par la méthode des moindres carrés, on cherche dans un premier temps à adapter le modèle de manière à pouvoir appliquer cette technique. Pour cela, on effectue les étapes suivantes :

1.En écrivant le produit en croix $D(p)Y(p) = N(p)U(p)$, montrer que :

$Y(p) = U(p)(a + b/p + c/p^2 + d/p^3) - Y(p)(\alpha/p + \beta/p^2 + \gamma/p^3)$

Réponse :

On sait que $N(p) = ap^3 + bp^2 + cp + d$ et que $D(p) = p^3 + \alpha p^2 + \beta p + \gamma$

Si on remplace dans $D(p)Y(p) = N(p)U(p)$, on obtient :

$Y(p)(p^3 + \alpha p^2 + \beta p + \gamma) = U(p)(ap^3 + bp^2 + cp + d)$

On divise par p^3 des deux côtés :

$Y(p)(1 + \alpha/p + \beta/p^2 + \gamma/p^3) = U(p)(a + b/p + c/p^2 + d/p^3)$

On soustrait $Y(p)(\alpha/p + \beta/p^2 + \gamma/p^3)$ des deux côtés :

$Y(p) = U(p)(a + b/p + c/p^2 + d/p^3) - Y(p)(\alpha/p + \beta/p^2 + \gamma/p^3)$

2. Prendre la transformée de Laplace inverse de l'expression précédente et montrer que le modèle possible de y(t) est donné par :

$ymod(t) = \varphi^T(t)P$

où $\varphi(t)$ est à déterminer et P le vecteur des paramètres à déterminer. $u(t)$ et $ymod(t)$ désignent respectivement la transformée de Laplace inverse de $U(p)$ et $Ymod(p)$. On précisera les dimensions de tous les termes.

Réponse :

On utilise la propriété $X(p)/p = \int_0^t X(\tau)d\tau$:

On sait que que : $Y(p) = U(p)(a + b/p + c/p^2 + d/p^3) - Y(p)(\alpha/p + \beta/p^2 + \gamma/p^3)$ grace à la question 1.

En utilisant la propriété ,la transformée de Laplace inverse de $Y(p)$ devient donc :

$L(Y(p))^-1 = aU(t) + b\int_0^t U(\tau)d\tau + c\int_0^t\int_0^t U(\tau)d\tau^2 + d\int_0^t\int_0^t\int_0^t U(\tau)d\tau^3 - \alpha\int_0^t Y(\tau)d\tau - \beta\int_0^t\int_0^t Y(\tau)d\tau^2 - \gamma\int_0^t\int_0^t\int_0^t Y(\tau)d\tau^3$

On met tout ça sous forme matricielle :

$$\varphi_{7,1}(t) = \begin{pmatrix} U(t) \\ \int_0^t U(\tau)d\tau \\ \int_0^t\int_0^t U(\tau)d\tau^2 \\ \int_0^t\int_0^t\int_0^t U(\tau)d\tau^3 \\ -\int_0^t Y(\tau)d\tau \\ -\int_0^t\int_0^t Y(\tau)d\tau^2 \\ -\int_0^t\int_0^t\int_0^t Y(\tau)d\tau^3 \end{pmatrix} \text{ et } P_{7,1} = \begin{pmatrix} a \\ b \\ c \\ c \\ \alpha \\ \beta \\ \gamma \end{pmatrix}$$

on a donc : $ymod(t) = \begin{pmatrix} U(t) \\ \int_0^t U(\tau)d\tau \\ \int_0^t\int_0^t U(\tau)d\tau^2 \\ \int_0^t\int_0^t\int_0^t U(\tau)d\tau^3 \\ -\int_0^t Y(\tau)d\tau \\ -\int_0^t\int_0^t Y(\tau)d\tau^2 \\ -\int_0^t\int_0^t\int_0^t Y(\tau)d\tau^3 \end{pmatrix}^T (t) * \begin{pmatrix} a \\ b \\ c \\ c \\ \alpha \\ \beta \\ \gamma \end{pmatrix}$

3. On suppose maintenant que l'on a effectué une expérience fournissant le vecteur de mesures suivant $Y = [y(t1), y(t2), ..., y(tN)]^T$ où $k = 1..N$

(a) Exprimer $ymod(tk)$.

Réponse :

: $ymod(tk) = \begin{pmatrix} U(tk) \\ \int_0^{tk} U(\tau)d\tau \\ \int_0^{tk}\int_0^{tk} U(\tau)d\tau^2 \\ \int_0^{tk}\int_0^{tk}\int_0^{tk} U(\tau)d\tau^3 \\ -\int_0^{tk} Y(\tau)d\tau \\ -\int_0^{tk}\int_0^{tk} Y(\tau)d\tau^2 \\ -\int_0^{tk}\int_0^{tk}\int_0^{tk} Y(\tau)d\tau^3 \end{pmatrix}^T (tk) * \begin{pmatrix} a \\ b \\ c \\ c \\ \alpha \\ \beta \\ \gamma \end{pmatrix}$

(b) Déterminer les paramètres a,b et c qui minimisent le critère :

$J = \sum_{k=1}^{N}(y(tk) - ymod(tk))^2$

Réponse :

On pose $J = \varepsilon^T\varepsilon$ avec :

$\varepsilon_{N,1} = \begin{pmatrix} y(t1) - ymod(t1) \\ y(t2) - ymod(t2) \\ ... \\ y(tN) - ymod(tN) \end{pmatrix}$

ce qui revient à :

$\varepsilon_{N,1} = \begin{pmatrix} y(t1) \\ y(t2) \\ ... \\ y(tN) \end{pmatrix} - \begin{pmatrix} ymod(t1) \\ ymod(t2) \\ ... \\ ymod(tN) \end{pmatrix}$

or :

$\begin{pmatrix} ymod(t1) \\ ymod(t2) \\ ... \\ ymod(tN) \end{pmatrix} = \phi_{N,7}P_{7,1}$

avec $\phi_{N,7} = \begin{pmatrix} \varphi_{7,1}(t1)^T \\ \varphi_{7,1}(t2)^T \\ ... \\ \varphi_{7,1}(tN)^T \end{pmatrix}$, $\varphi_{7,1}(tk) = \begin{pmatrix} U(tk) \\ \int_0^{tk} U(\tau)d\tau \\ \int_0^{tk}\int_0^{tk} U(\tau)d\tau^2 \\ \int_0^{tk}\int_0^{tk}\int_0^{tk} U(\tau)d\tau^3 \\ -\int_0^{tk} Y(\tau)d\tau \\ -\int_0^{tk}\int_0^{tk} Y(\tau)d\tau^2 \\ -\int_0^{tk}\int_0^{tk}\int_0^{tk} Y(\tau)d\tau^3 \end{pmatrix}$ et $P_{7,1} = \begin{pmatrix} a \\ b \\ c \\ c \\ \alpha \\ \beta \\ \gamma \end{pmatrix}$

On note : $Y_{N,1} = \begin{pmatrix} y(t1) \\ y(t2) \\ ... \\ y(tN) \end{pmatrix}$

On a donc : $J = (Y_{N,1} - \phi_{N,7}P_{7,1})^T(Y_{N_1} - \phi_{N,7}P_{7,1})$

Or $(AB)^T = A^TB^T$ donc :

$J = (Y_{N,1}^T - \phi_{N,7}^T P_{7,1}^T)(Y_{N_1} - \phi_{N,7}P_{7,1})$

On peut voir que J dépend explicitement de P, et donc de a,b et c.

$J = Y_{N,1}^T Y_{N,1} - Y_{N,1}^T \phi_{N,7} P_{7,1} - P_{7,1}^T \phi_{N,7}^T Y_{N_1} + P_{7,1}^T \phi_{N,7}^T \phi_{N,7} P_{7,1}$

Condition du 1er ordre : $\Delta j = 0$

$\Delta Y^T Y = d(Y^T Y)/dP = 0_{N,1}$

$\Delta(Y^T\phi P) = d(Y^T\phi P)/dP = \phi^T Y$

$\Delta(P^T\phi^T Y) = d(P^T\phi^T Y)/dP = \phi^T Y$

$\Delta(P^T\phi^T\phi P) = d(P^T\phi^T\phi P))/dP$

$= [\phi^T\phi + (\phi^T\phi)^T]P$

$= 2\phi^T\phi P$

Donc : $\Delta J = 0_{N,1} - \phi^T Y - \phi^T Y + 2\phi^T\phi P$

Ce qui donne : $2\phi^T\phi P - 2\phi^T Y$

Condition du 1er ordre : $\hat{P}$ est optimum si $\Delta J = 0$ en $\hat{P}$

Donc : $2\phi^T\phi P - 2\phi^T Y = 0$

$2\phi^T\phi P = 2\phi^T Y$

$\phi^T\phi P = \phi^T Y$

Estimateur des moindres carrés : $\hat{P} = (\phi^T\phi)^{-1}\phi^T Y$

$\hat{P} = (\begin{pmatrix} \varphi_{7,1}(t1)^T \\ \varphi_{7,1}(t2)^T \\ ... \\ \varphi_{7,1}(tN)^T \end{pmatrix}^T \begin{pmatrix} \varphi_{7,1}(t1)^T \\ \varphi_{7,1}(t2)^T \\ ... \\ \varphi_{7,1}(tN)^T \end{pmatrix})^{-1} \begin{pmatrix} \varphi_{7,1}(t1)^T \\ \varphi_{7,1}(t2)^T \\ ... \\ \varphi_{7,1}(tN)^T \end{pmatrix}^T \begin{pmatrix} y(t1) \\ y(t2) \\ ... \\ y(tN) \end{pmatrix}$

Quel sens à ce critère ?

On cherche à minimiser l'erreur entre le système réel et celui modélisé.

2.2 Mise en oeuvre sous Matlab

1. Rappeler brièvement le principe de la méthode d'intégration des rectangles.

Réponse :

La méthode d'intégration des rectangles permet de calculer l'intégrale d'une fonction en additionnant les aires des rectangles formant la courbe. Cette aire est donc le produit du pas d'échantillonnage et de la valeur de l'échantillon en un tk donné.

En quoi la fonction cumsum sous matlab peut-elle être intéressante pour réaliser cette fonction d'intégration ?

Réponse :

B = cumsum(A) retourne la somme cumulée A partir du début de la première dimension du tableau A dont la taille est différente de 1.

Si A est un vecteur, alors cumsum(A) renvoie un vecteur contenant la somme cumulée des éléments A.

En multipliant les échantillons par le pas d'échantillonnage, elle permettra de calculer la somme des aires des rectangles et donc réaliser l'intégration.

2.A l'aide de la fonction précédente, déterminer la valeur du vecteur de paramètres P sous matlab. Stocker la fonction de transfert estimée dans une variable sous matlab.

Réponse :

En exécutant ce code matlab, on obtient :

```
load data_ech_10000
Te = 0.001 %pas d'échantillonage
%on note phi =(u1;u2;u3;u4;y1;y2;y3) pour simplifier l'ecriture dans matlab
u1 = u
u2=cumsum(u1*Te)
u3=cumsum(u2*Te)
u4=cumsum(u3*Te)
y1=cumsum(-Ybruit*Te)
y2=cumsum(-y1*Te)
y3=cumsum(-y2*Te)
Phi=[ u1 u2 u3 u4 y1 y2 y3]
% Pestime = [a b c d alpha beta gamma]
Pestime = inv(transpose(Phi)*Phi)*transpose(Phi)*Ybruit
```

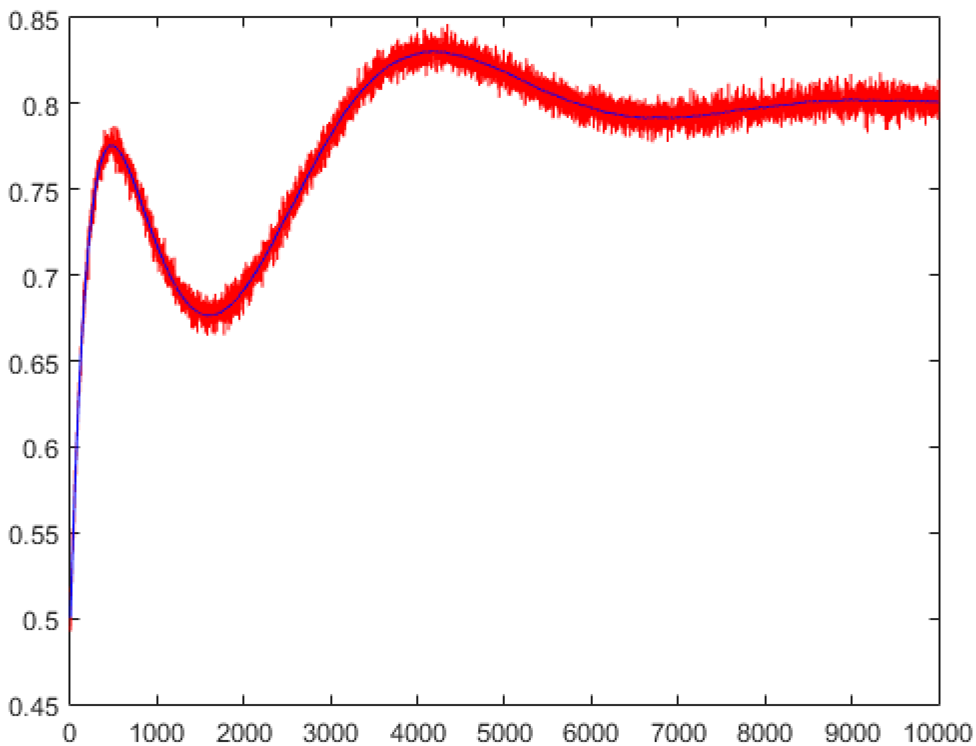
$$\hat{P} = \begin{pmatrix} 0.4987 \\ 4.1954 \\ 3.2927 \\ 5.6041 \\ 4.9945 \\ -5.9935 \\ 7.0058 \end{pmatrix}$$

3. Superposer la sortie du modèle aux mesures. Conclure quand à la qualité de l'identification.

Réponse :

```
Estimation=theta*Pestime
plot(Ybruit, 'R')
hold on;
plot(Estimation, 'B')
```

Pour N=10000, avec en l'amplitude ordonnée et le numéro d'échantillon en abscisse.



On a ici, en bleu le modèle estimé et en rouge ce que l'on a mesuré. On observe que la qualité de l'identification est bonne, la courbe bleue est juste lissée par rapport à la courbe mesurée (pas de bruit du aux mesures).

4. On souhaite maintenant analyser l'impact du nombre de mesures prélevées sur la qualité de l'identification. Pour cela, on effectue une seconde expérience où l'on sollicite le système avec le même échelon, maintenant en prélevant seulement $N = 50$ points.

(a) Estimer les paramètres pour ce deuxième cas de figure et donner leur valeur.

Réponse :

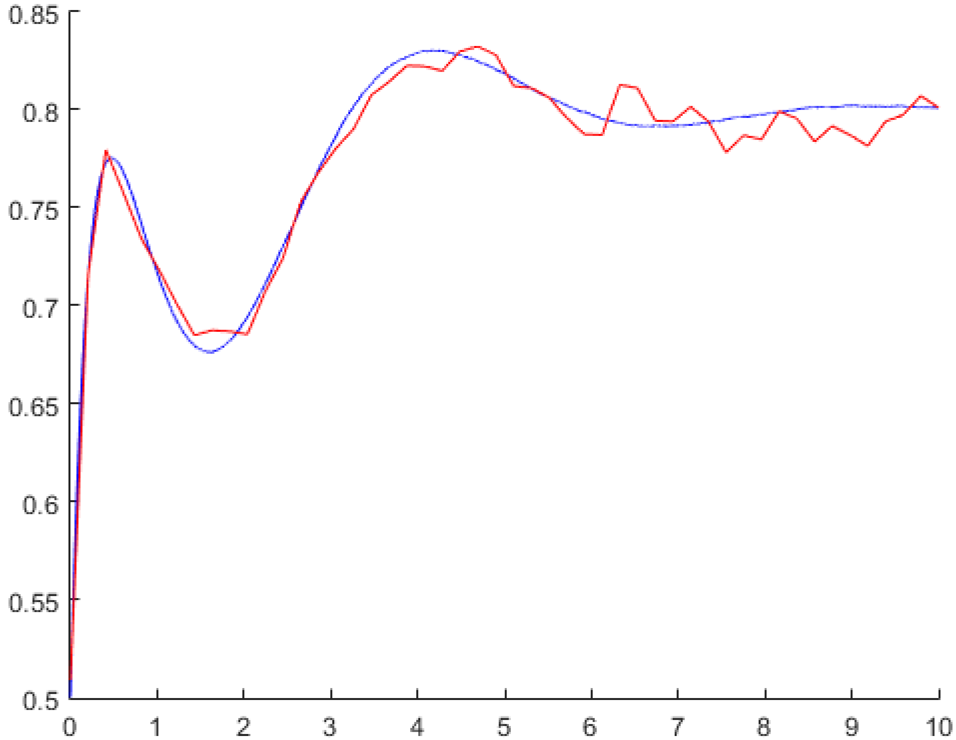
```
load data_ech_50
t2=t
Te = 0.2041;%pas d'échantillonage
%on note phi =(u1;u2;u3;u4;y1;y2;y3) pour simplifier l'ecriture dans matlab
u1 = u
u2=cumsum(u1*Te)
u3=cumsum(u2*Te)
u4=cumsum(u3*Te)
y1=cumsum(-Ybruit*Te)
y2=cumsum(-y1*Te)
y3=cumsum(-y2*Te)
Phi2=[ u1 u2 u3 u4 y1 y2 y3]
% Pestime = [a b c d alpha beta gamma]
Pestime2 = inv(transpose(Phi2)*Phi2)*transpose(Phi2)*Ybruit

Estimation2=Phi2*Pestime2
```

$$\hat{P} = \begin{pmatrix} -0.0450 \\ 6.0828 \\ 2.3675 \\ 8.8804 \\ 6.6415 \\ -6.4200 \\ 11.1065 \end{pmatrix}$$

(b) Superposer maintenant les mesures et les sorties du modèle obtenues pour N=50 et N=1000 mesures. Conclure **Réponse :**

```
load data_ech_10000
t1=t
Te1 = 0.001 %pas d'échantillonage
%on note phi =(u1;u2;u3;u4;y1;y2;y3) pour simplifier l'ecriture dans matlab
u1 = u
u2=cumsum(u1*Te1)
u3=cumsum(u2*Te1)
u4=cumsum(u3*Te1)
y1=cumsum(-Ybruit*Te1)
y2=cumsum(-y1*Te1)
y3=cumsum(-y2*Te1)
Phi1=[ u1 u2 u3 u4 y1 y2 y3]
% Pestime = [a b c d alpha beta gamma]
Pestime1 = inv(transpose(Phi1)*Phi1)*transpose(Phi1)*Ybruit

Estimation1=Phi1*Pestime1


figure;
hold on;
plot(t1,Estimation1, 'b')
plot(t2,Estimation2, 'r')
%plot(t1,Ybruit, 'b')
```

Pour N=10000 en bleue et N=50 en rouge, avec en ordonnée l'amplitude et le temps en secondes en abscisse.



Le modèle à 50 échantillons est moins précis que celui à 10000. Le nombre d'échantillons est donc un facteur déterminant de la qualité de l'identification.

5. Afin d'évaluer la validité de l'estimation, on effectue une dernière expérience où le système est sollicité avec une sinusoïde. Celle-ci est définie par la transformée de Laplace suivante :

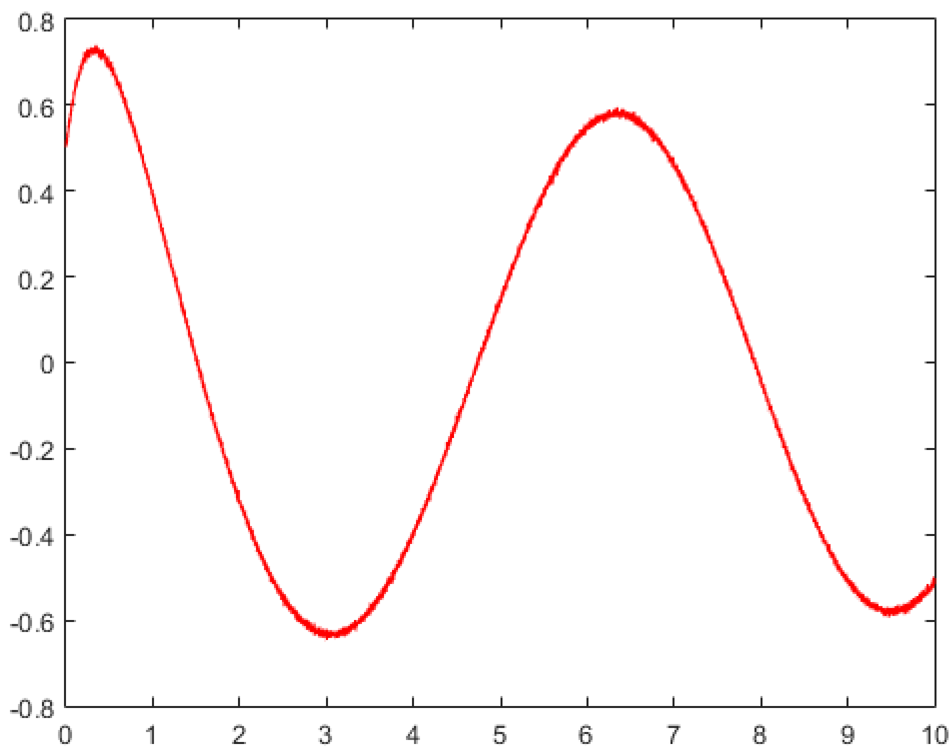
$p/(p^2 + 1)$

Simuler la réponse du modèle obtenu précédemment avec la sinusoïde.

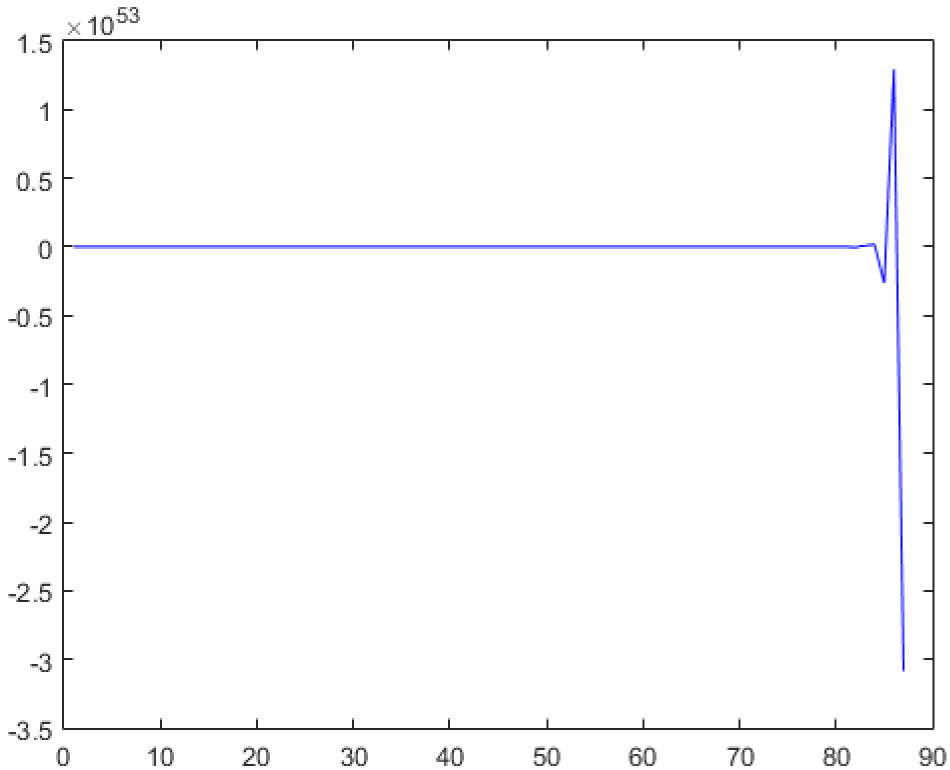
Réponse :

```
load('data_sinus_10000.mat')

num = [1 0];
den = [1 0 1];
%sys = tf(Numerator,Denominator) creates a continuous-time transfer
%function with numerator(s) and denominator(s) specified by Numerator
%and Denominator.
F=tf([Phi1(1) Phi1(2) Phi1(3) Phi1(4)], [1 Phi1(5) Phi1(6) Phi1(7)]);
Sin = tf(num,den);
G = F*Sin;
Estimation3=impulse(G)

figure;
plot(t,Ysinus,'R') %comportement réel du système
figure;
plot(Estimation3,'B')
```

Le système réel :



La réponse à la sinusoïde du système modélisé précédemment :



Conclusion : Notre modélisation diverge, ce qui n'est pas le cas des mesures.