

Chapitre IV

Simulation

IV.1 Partie I

Comme tout problème complexe, l'identification des systèmes exige une simulation avant de passer à une réalisation. Dans cette partie on va appliquer la technique d'identification série parallèle sur quatre systèmes non linéaires, et la modélisation direct de la fonction sinus et pour cela on va utiliser l'algorithme de rétropropagation du gradient qui est un algorithme d'apprentissage supervisé, ainsi que la structure neuronale utilisée dans les exemple de simulation est une structure multicouches, à un et deux couches cachées. La fonction sigmoïde est utilisée comme fonction d'activation des neurones cachés, alors que les neurones de sortie ont une fonction d'activation linéaire.

IV.1.1 Présentation des systèmes à identifier

➤ Système 1

Le système 1 à identifier est donné par l'équation récurrente suivante :

$$Y(k+1) = f[Y(k)] + u^3(k)$$

Où la fonction inconnue f est définie (pour la simulation) par :

$$f(Y(k)) = \frac{Y(k)}{1 + Y^2(k)}$$

Avec

$$u(k) = \sin(2\pi.k / 40)$$

➤ Système 2

Le système 2 à identifier est donné par l'équation récurrente suivante :

$$Y(k+1) = f[y(k-1), y(k-2), u(k), u(k-1)]$$

Où la fonction inconnue f est définie (pour la simulation) par :

$$f[y(k-1), y(k-2)u(k), u(k-1)] = \frac{0.6 \cdot [\sin(y(k-1)^2 \cdot y(k-2) \cdot (y(k-2)-1) \cdot u(k-1)) + u(k)]}{[1 + y(k-2)^2 + y(k-1)^2]}$$

Avec

$$u(k) = 0.6 \cdot (\sin(2\pi \cdot k / 250)) + 0.2 \cdot (\sin(2\pi \cdot k / 20))$$

➤ **Systeme 3**

Le système 3 à identifier est donné par l'équation récurrente suivante :

$$Y(k+1) = f[y(k)] + u(k)$$

Où la fonction inconnue f est définie (pour la simulation) par :

$$f[y(k)] = \frac{y(k)}{1 + y^2(k)}$$

Avec

$$u(k) = \sin(2\pi \cdot k / 25) + \sin(2\pi \cdot k / 50) + \sin(2\pi \cdot k / 75) + \sin(2\pi \cdot k / 100) \\ + \sin(2\pi \cdot k / 125) + \sin(2\pi \cdot k / 150) + \sin(2\pi \cdot k / 250) + \sin(2\pi \cdot k / 500)$$

➤ **Systeme 4**

Le système 4 à identifier est donné par l'équation récurrente suivante :

$$Y(k+1) = f[y(k)] + u(k)$$

Où la fonction inconnue f est définie (pour la simulation) par :

$$f[y(k)] = \frac{y(k)}{1 + y^2(k)}$$

Avec

$$u(k) = \sin(2\pi \cdot k / 250) \quad k \in [1 \dots 500] \\ u(k) = 0,6 \sin(2\pi \cdot k / 250) + 0,2 \cos(2\pi \cdot k / 25) \quad k \in [501 \dots 800]$$

IV.1.2 Résultats d'identification

IV.1.2.1 Modélisation directe

Le modèle neuronal que nous avons choisi pour la modélisation direct de la fonction sinus est :

- 01 couche cachée.
- 04 neurones dans la couche d'entrée.
- 03 neurones dans la couche cachée.
- Un seul neurone dans la couche de sortie.
- La fonction d'activation utilisée dans la couche cachée et couche d'entrée est la

fonction sigmoïde.
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire

$$f(x) = x$$

- Pas d'apprentissage $\mu = 0.009$.
- Moment $\beta = 0.001$.

Après 10000 itérations, les sorties obtenues (fonction- modèle) et l'erreur sont représentées par les figures (IV.1) et (IV.2).

IV.1.2.2 Identification série parallèle

➤ Système 1

Le modèle neuronal que nous avons choisi est :

- 01 couche cachée.
- 05 neurones dans la couche d'entrée.
- 15 neurones dans la couche cachée.
- Un seul neurone dans la couche de sortie.
- La fonction d'activation utilisée dans la couche cachée et couche d'entrée est la

fonction sigmoïde.
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire

$$f(x) = x$$

- Pas d'apprentissage $\mu = 0.01$.
- Moment $\beta = 0.00011$.

Après 13905 itérations, les sorties obtenues (système-modèle) avec validation sont représentées par la figure (IV.3) et l'erreur entre le modèle neuronal et le système représenté par la figure (IV.4).

➤ **Système 2**

Le modèle neuronal que nous avons choisi est :

- 01 couche cachée.
- 06 neurones dans la couche d'entrée.
- 03 neurones dans la couche cachée.
- Un seul neurone dans la couche de sortie.
- La fonction d'activation utilisée dans la couche cachée et couche d'entrée est

la fonction sigmoïde.
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire
$$f(x) = x$$
- Pas d'apprentissage $\mu = 0.12$
- Moment $\beta = 0.01$

Après 6580 itérations, les sorties obtenues (système-modèle) avec validation sont représentées par la figure (IV.5) et l'erreur entre le modèle neuronal et le système représenté par la figure (IV.6).

➤ **Système 3**

Le modèle neuronal que nous avons choisi est :

- 01 couche cachée.
- 07 neurones dans la couche d'entrée.
- 05 neurones dans la couche cachée.
- Un seul neurone dans la couche de sortie.

- La fonction d'activation utilisée dans la couche d'entrée et la couche cachée est la fonction sigmoïde.
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$
- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire
$$f(x) = x$$
- Pas d'apprentissage $\mu = 0.045$.
- Moment $\beta = 0.0001$.

Après 11000 itérations, les sorties obtenues (système-modèle) avec validation sont représentées par la figure (IV.7) et l'erreur entre le modèle neuronal et le système représenté par la figure (IV.8).

➤ **Systeme 4**

Le modèle neuronal que nous avons choisi est :

- 02 couches cachées.
- 07 neurones dans la couche d'entrée.
- 13 neurones dans la première couche cachée.
- 05 neurones dans la deuxième couche cachée.
- Un seul neurone dans la couche de sortie.
- La fonction d'activation utilisée dans la couche d'entrée et les deux couches cachées est la fonction sigmoïde.
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$
- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire
$$f(x) = x$$
- Pas d'apprentissage $\mu = 0.2$.
- Moment $\beta = 0.00001$.

Après 15360 itérations, les sorties obtenues (système-modèle) avec validation sont représentées par la figure (IV.9) et l'erreur entre le modèle neuronal et le système représenté par la figure (IV.10).

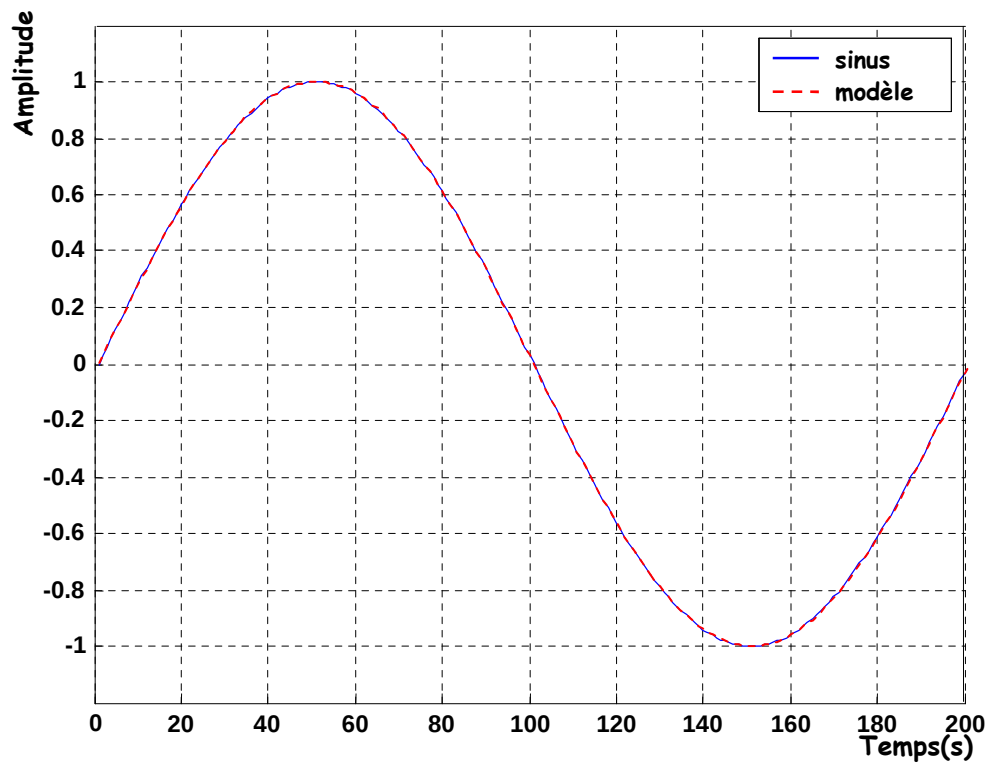


Figure (IV.1) Modélisation direct de la fonction sinus

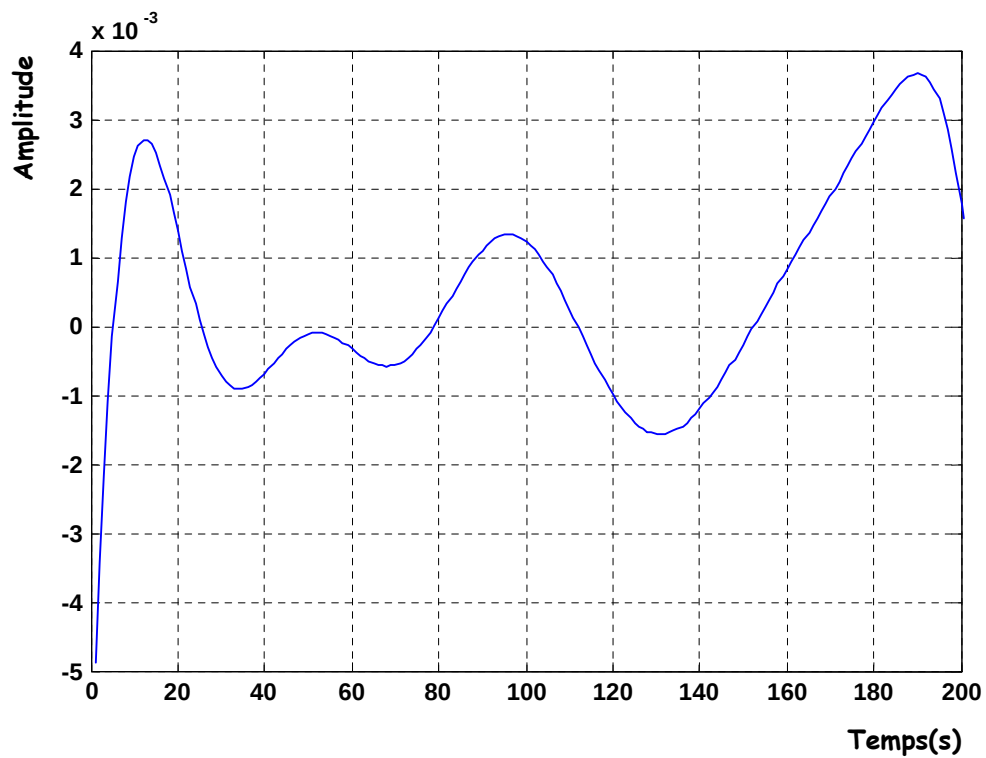
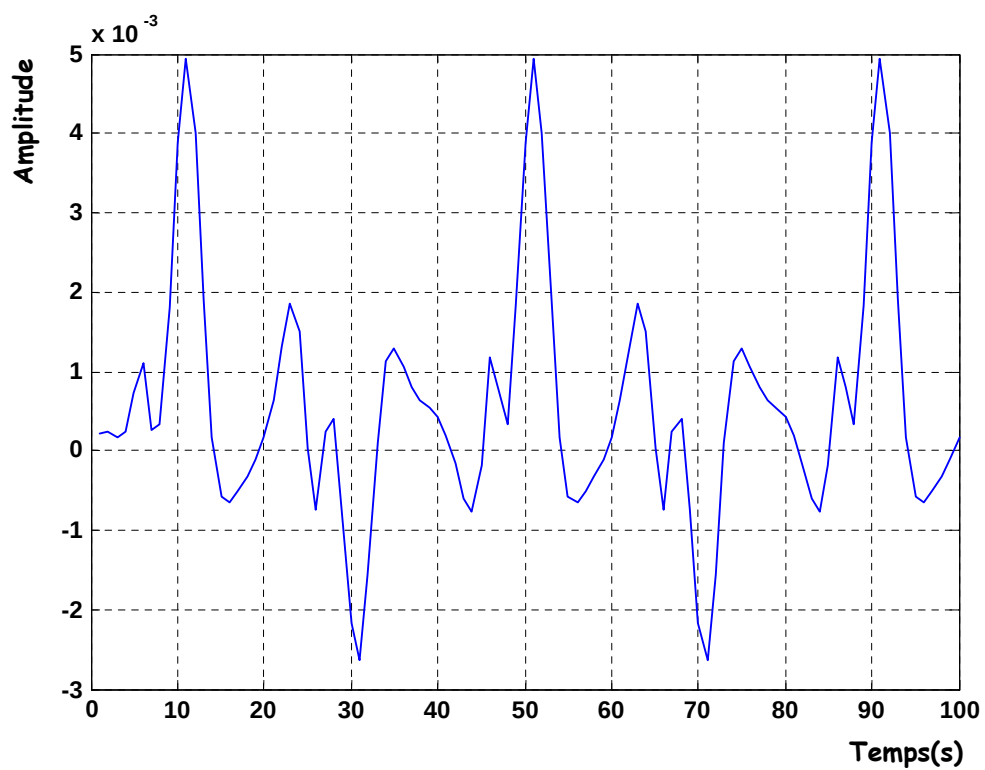
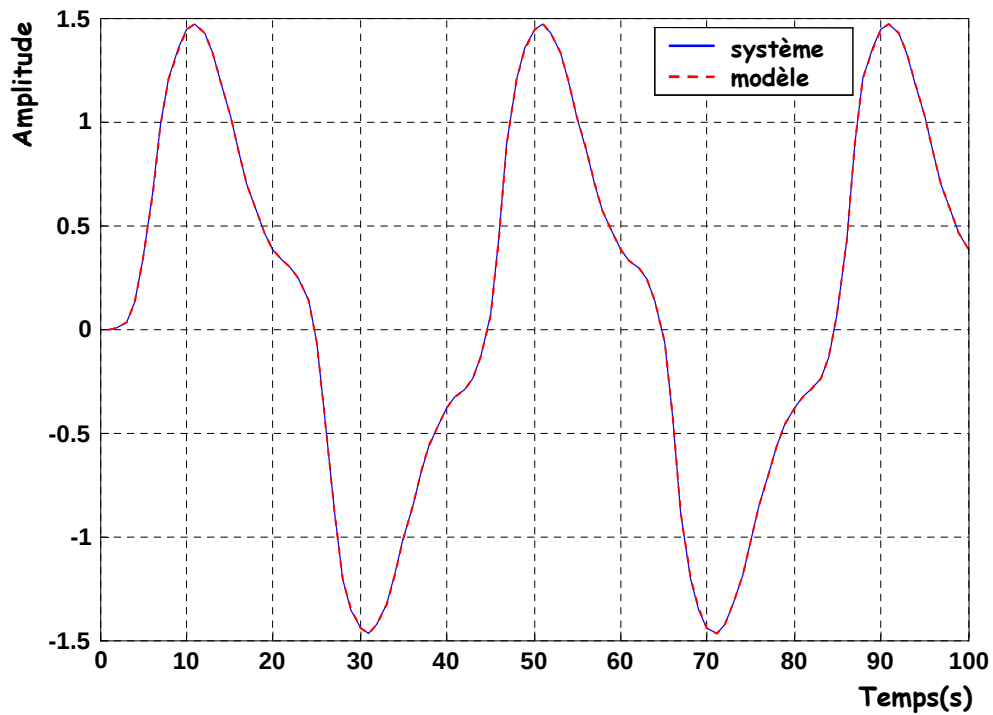


Figure (IV.2) Erreur



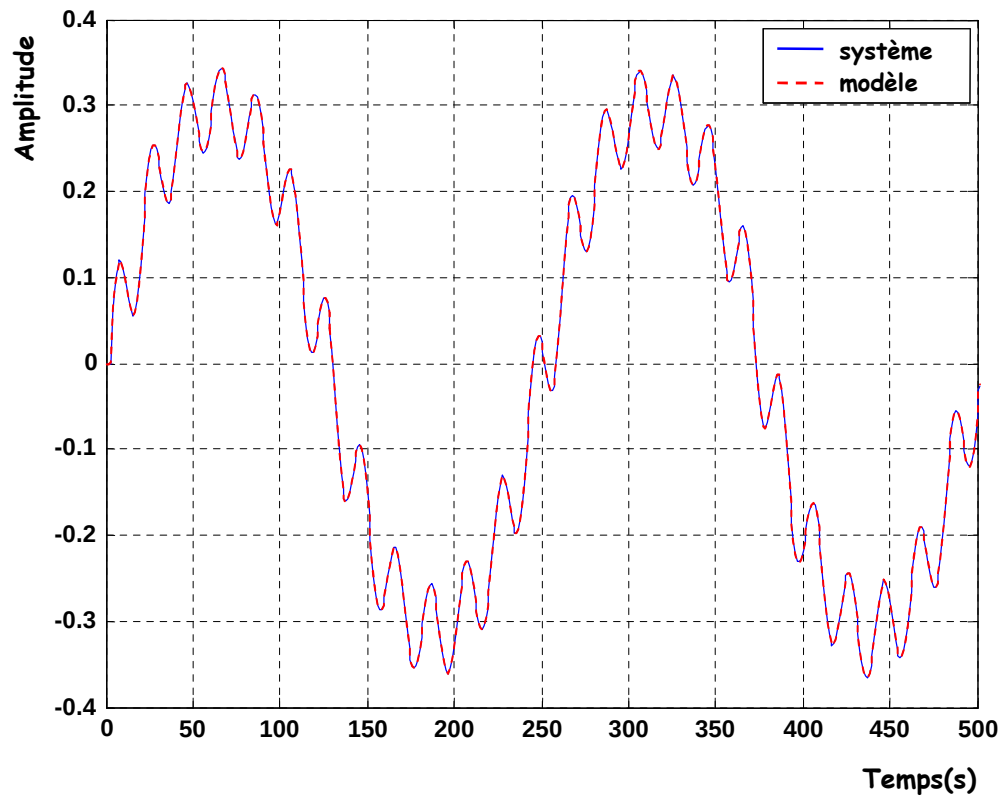


Figure (IV.5) Sortie du système 2 et de modèle neuronal

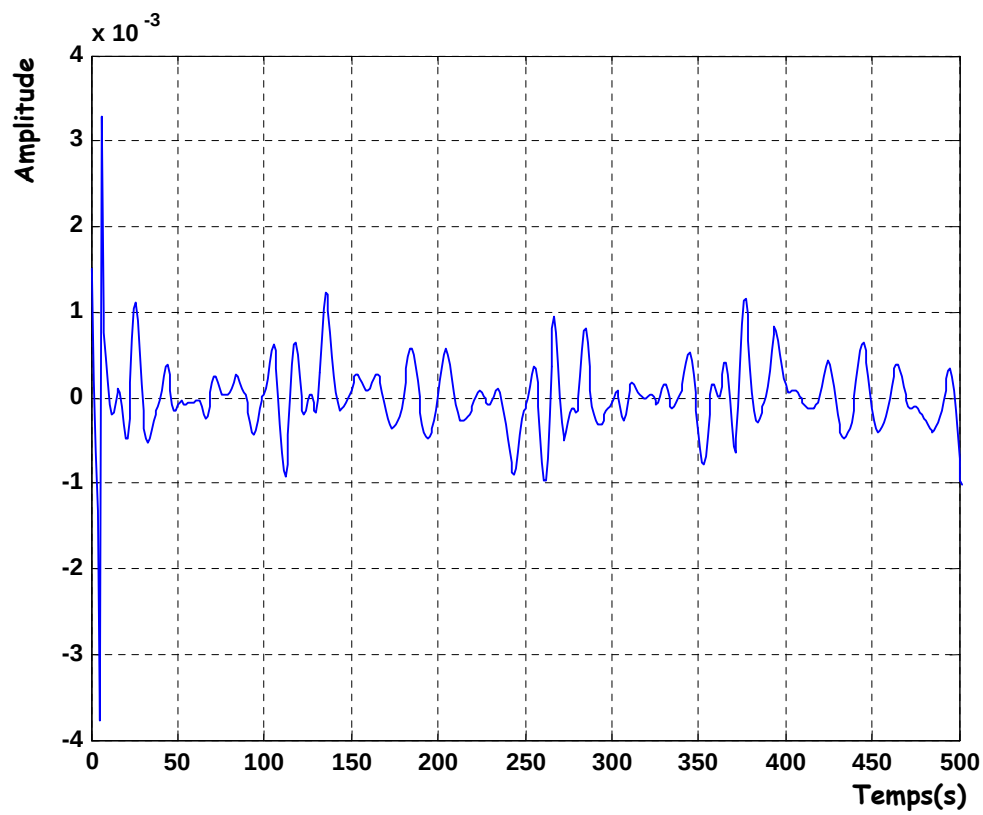


Figure (IV.6) Erreur entre le système 2 et le modèle

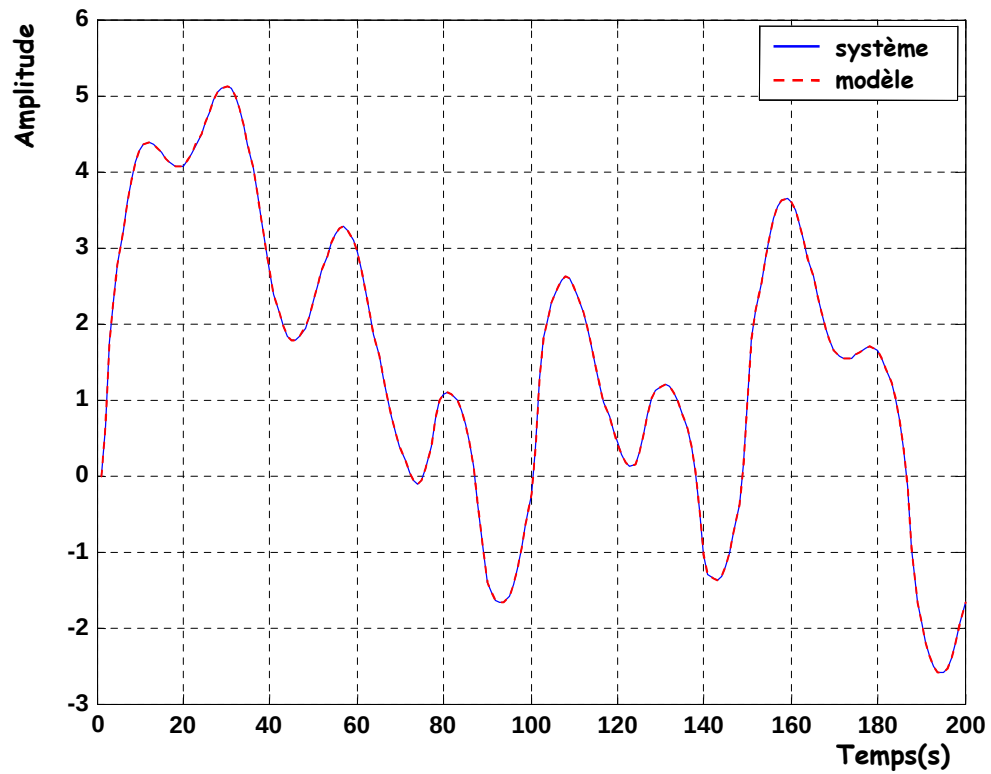


Figure (IV.7) Sortie du système 3 et de modèle neuronal

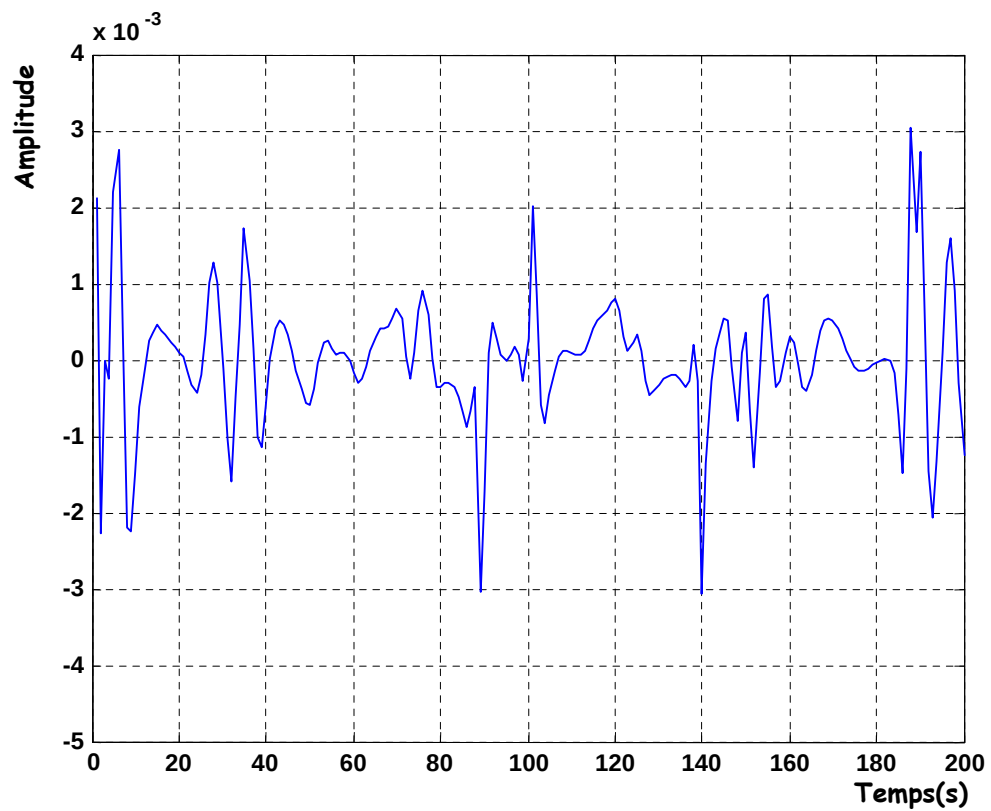


Figure (IV.8) Erreur entre le système 3 et le modèle

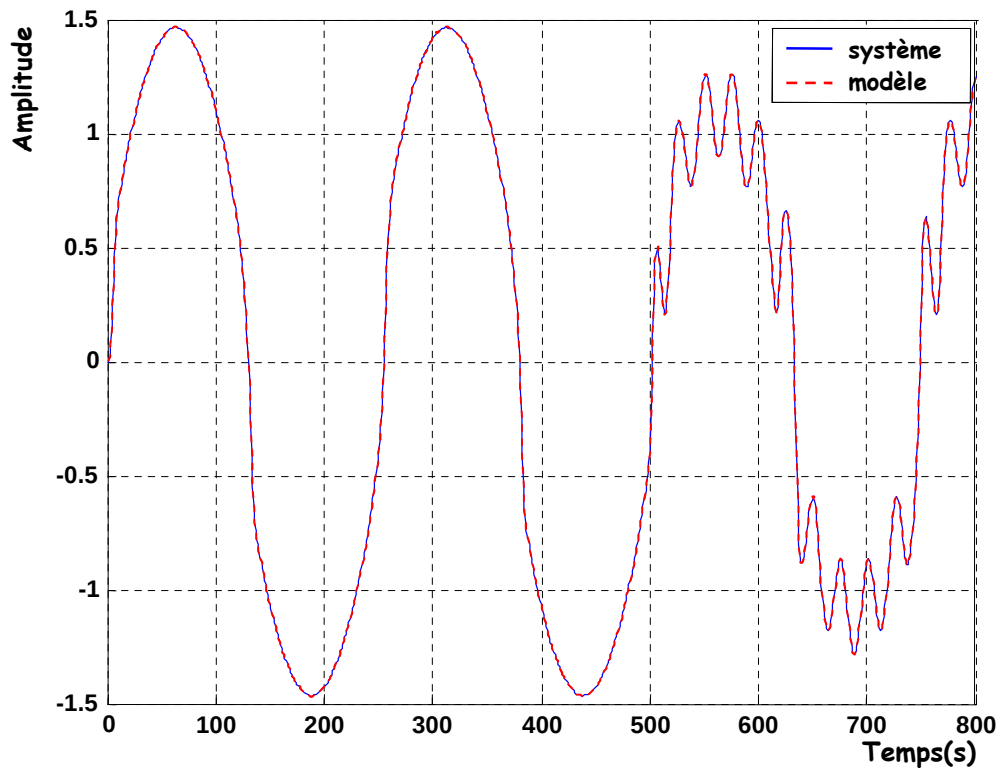


Figure (IV.9) Sortie du système4 et de modèle neuronal

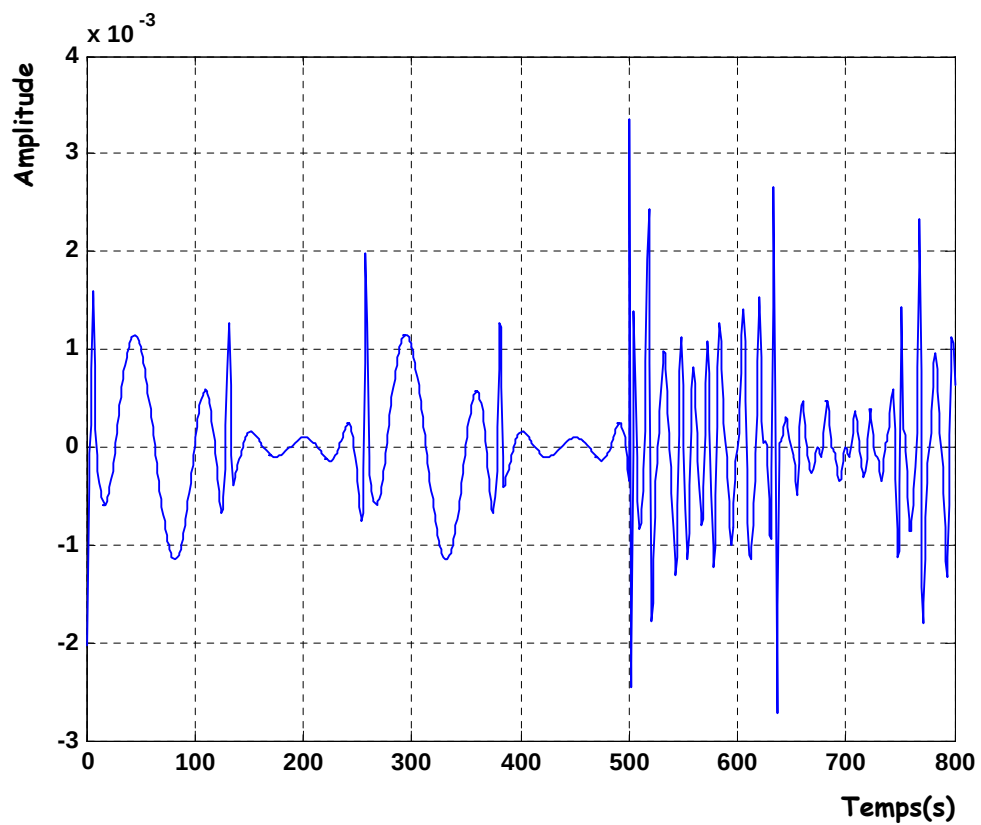


Figure (IV.10) Erreur entre le système 4 et le modèle

IV.1.3 Conclusion

Nous venons de voir dans cette partie que les réseaux de neurones peuvent être utilisés pour identifier les systèmes non linéaires. L'efficacité d'identification neuronale était démontrée par des exemples de simulation et des résultats satisfaisants ont été obtenus.

Plusieurs difficultés peuvent se présenter pendant la phase d'apprentissage car la minimisation de l'erreur n'est pas garantie, elle dépend de plusieurs paramètres : le nombre et la taille des couches cachées, l'initialisation des poids et le pas d'apprentissage μ jouent un rôle très important à la minimisation de l'erreur.

Pour les systèmes complexes, il est nécessaire d'augmenter le nombre de neurones dans la couche cachée, par conséquent l'apprentissage devient très lent.

Les programmes utilisés à l'identification sont présentés en annexe A et B.

IV.2 Partie II

Dans cette partie, on va appliquer les techniques de contrôle étudiées dans le chapitre II sur notre système (parabole de radar de poursuite).

IV.2.1 Identification du modèle direct

Pour identifier le modèle direct du système, on a utilisé la structure d'identification directe

Le modèle neuronal que nous avons choisi est :

- 01 couche cachée.
- 07 neurones dans la couche d'entrée.
- 05 neurones dans la couche cachée.
- Un seul neurone dans la couche de sortie.
- La fonction d'activation utilisée dans la couche cachée et couche d'entrée est

la fonction sigmoïde.
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire
$$f(x) = x$$
- Pas d'apprentissage $\mu = 0.05$.
- Moment $\beta = 0.0001$.

Nous disposons de 200 paires de mesures entrées-sorties (figures (IV.11) et figure (IV.12)) avec un pas d'échantillonnage de 0.1 seconde.

La sortie du modèle obtenue après 9735 itération et la sortie du système pour une entrée :

$$u(k) = -180.\sin(2.\pi.k / 7)$$

Et l'erreur entre le modèle et le système sont représentées respectivement sur la figure (IV.13) et figure (IV.14).

On remarque que l'écart entre les deux sorties est négligeable $9,7 \cdot 10^{-8}$.

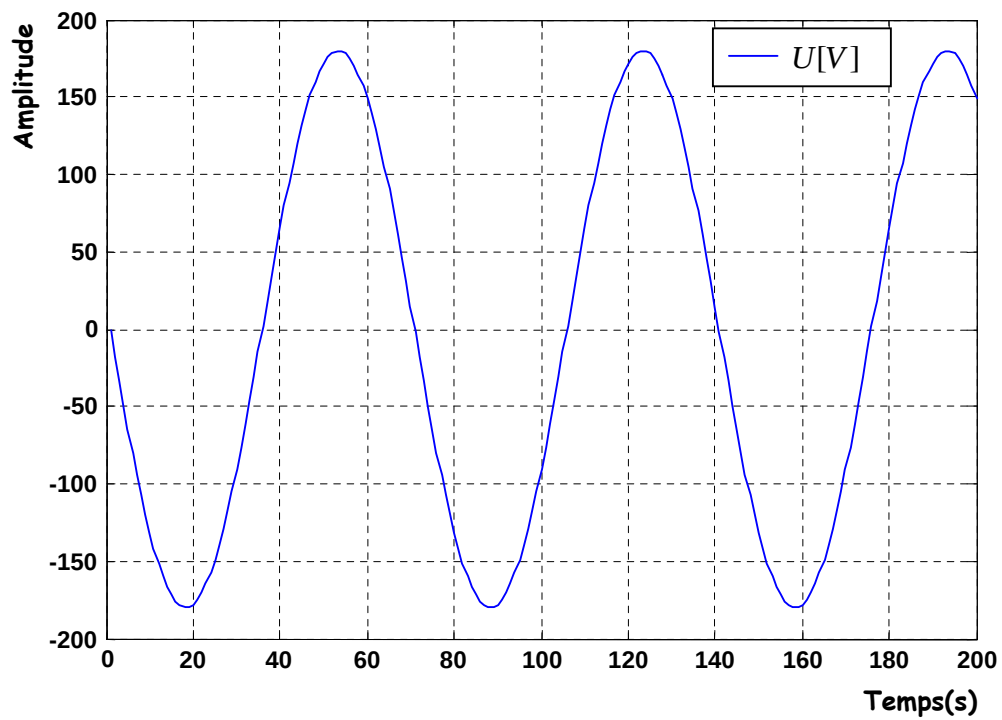


Figure (IV.11) Entrée du système

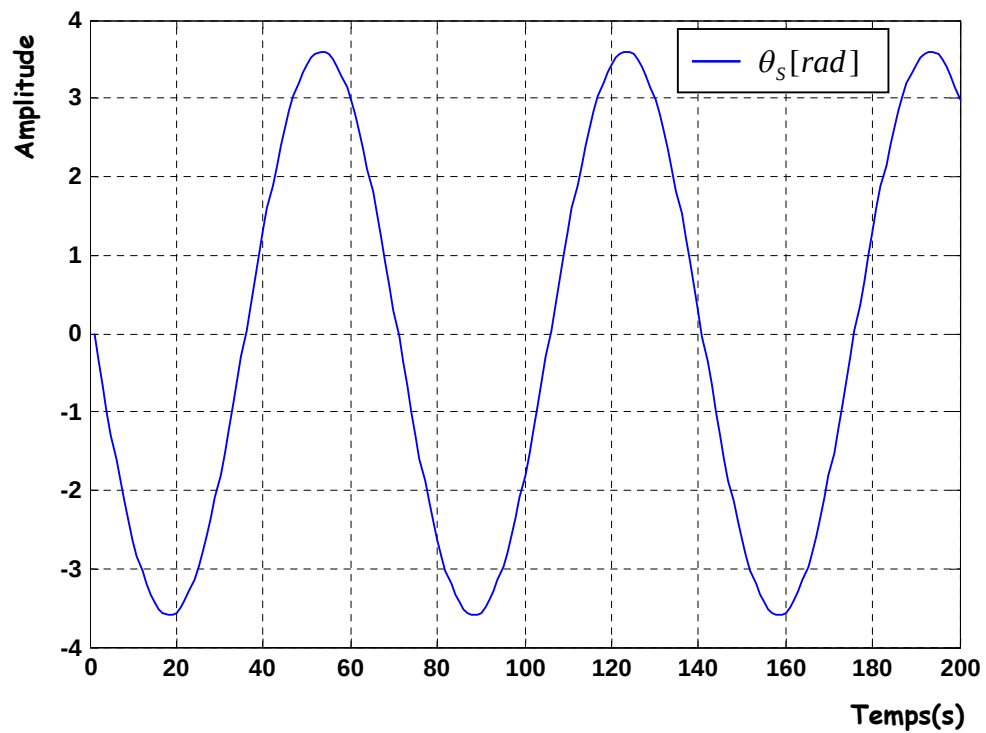


Figure (IV.12) Sortie du système

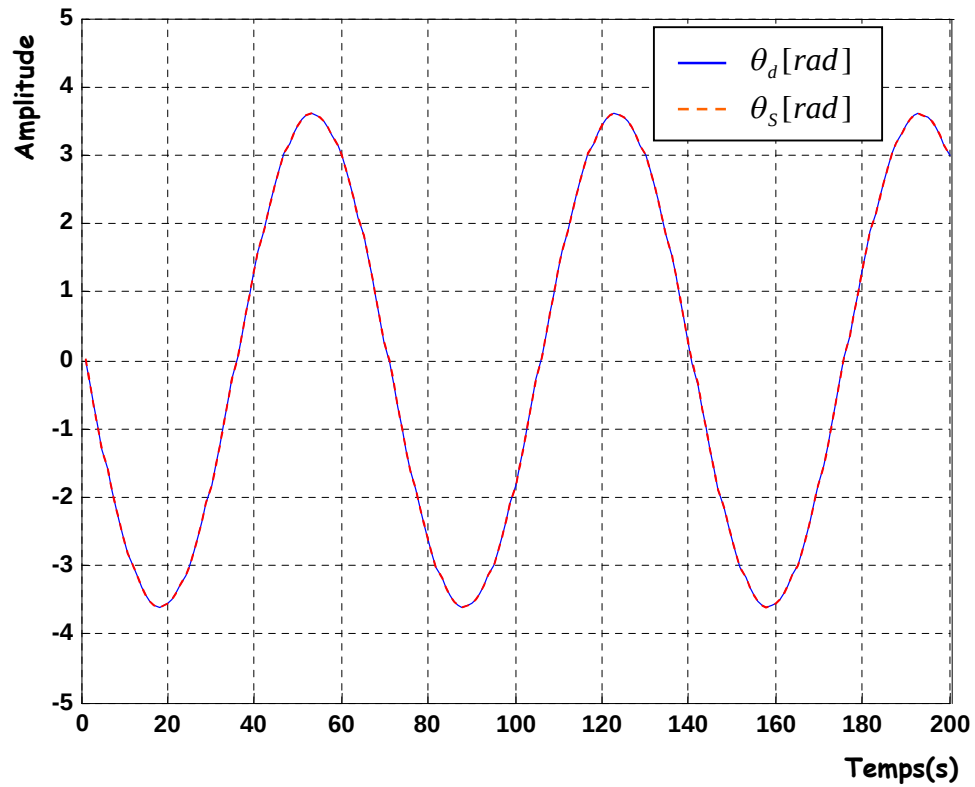


Figure (IV.13) Identification du modèle direct

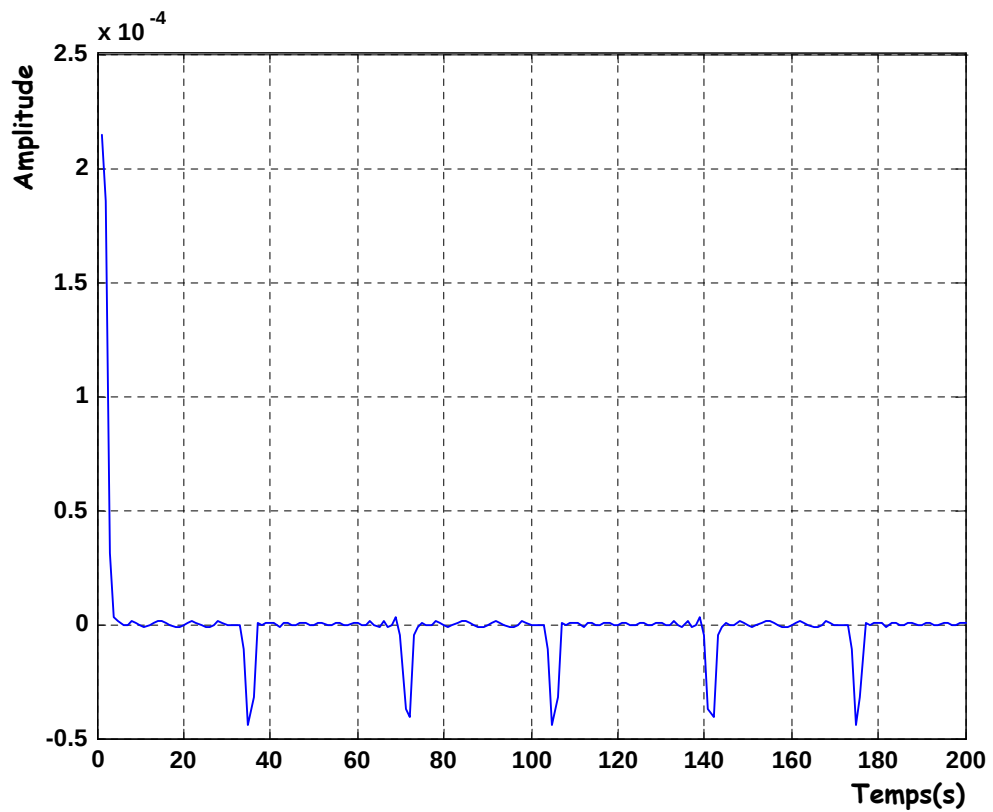


Figure (IV.14) Erreur entre le modèle et le système

➤ **test du modèle direct**

Le modèle direct obtenu par le réseau a été testé par une autre entrée tel que :

$$u_1(k) = 80.(\cos(2.\pi.k/5) + \cos(2.\pi.k/8))$$

$$u_2(k) = 100.(\sin(2.\pi.k/7) + \sin(2.\pi.k/1.5))$$

- Le résultat du test 1 ($u_1(k)$) et l'erreur sont représentés par la figure (IV.15) et figure (IV.16)
- Le résultat du test 2 ($u_2(k)$) et l'erreur sont représentés par la figure (IV.17) et figure (IV.18)

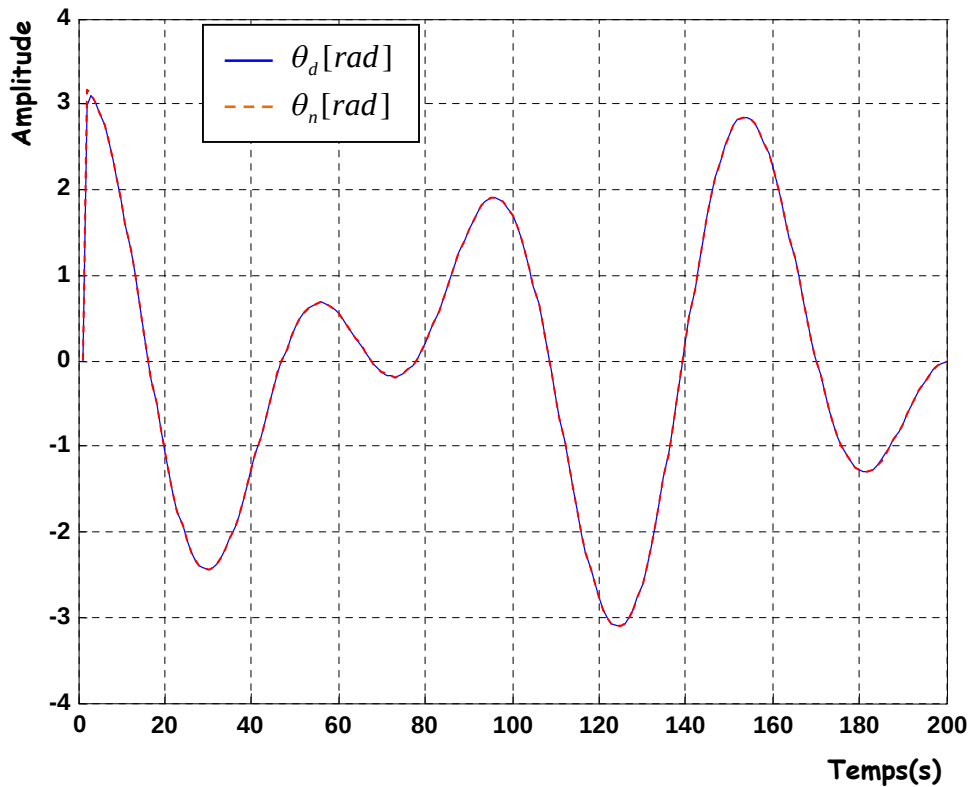


Figure (IV.15) Test 1 du modèle direct

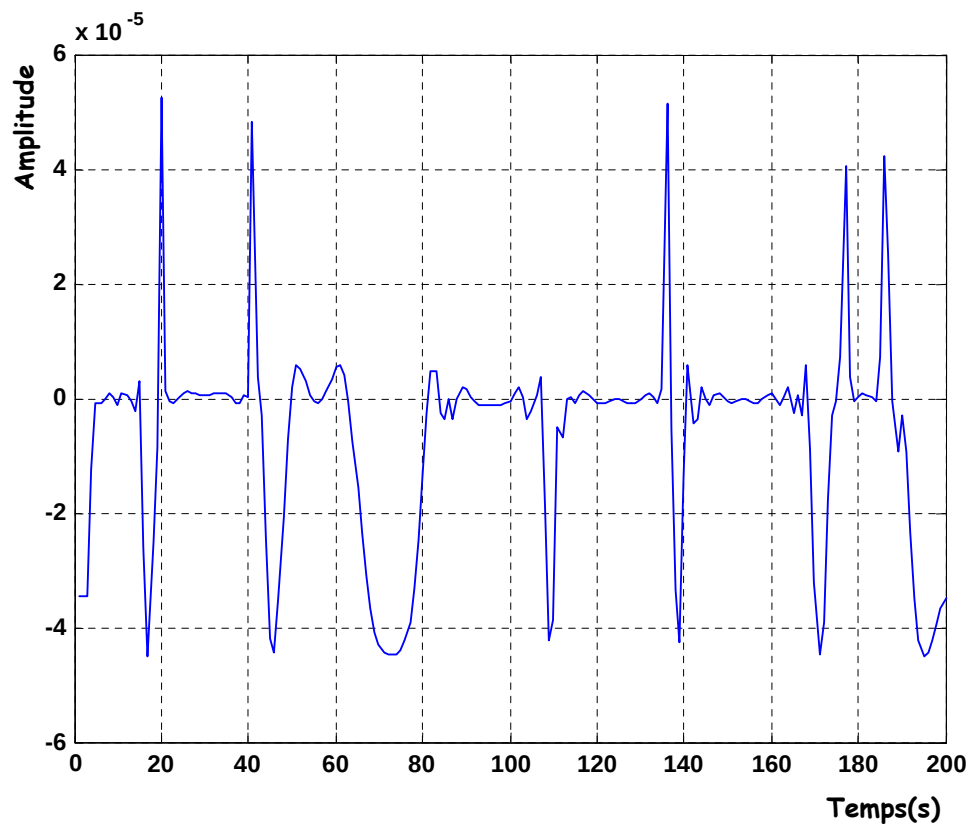


Figure (IV.16) Erreur

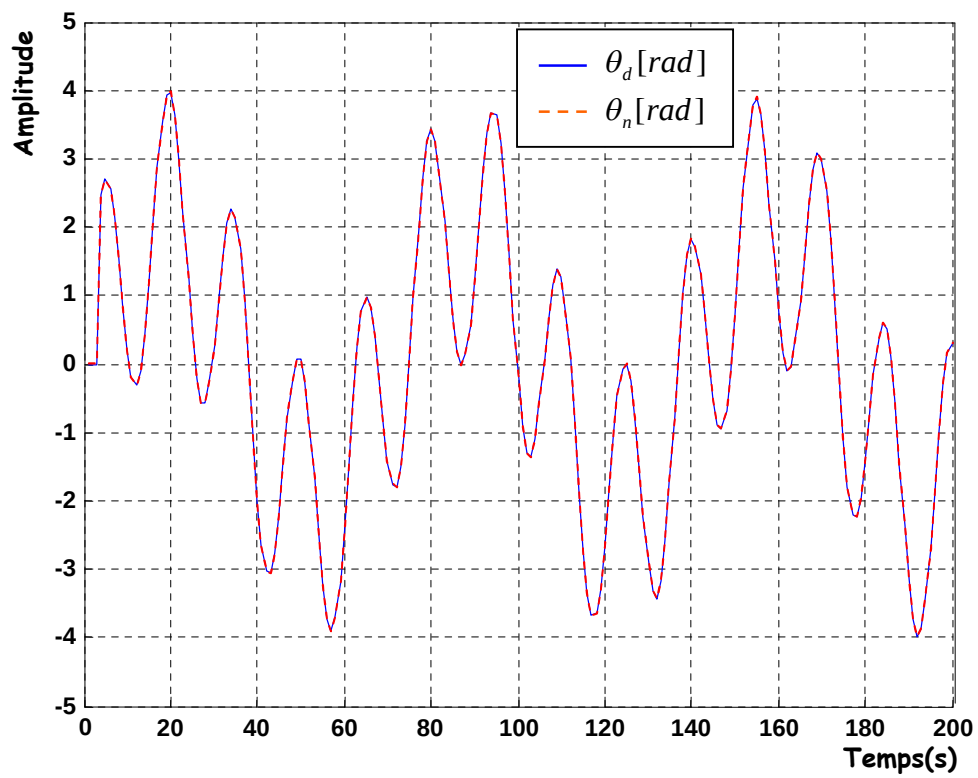


Figure (IV.17) Test 2 du modèle direct

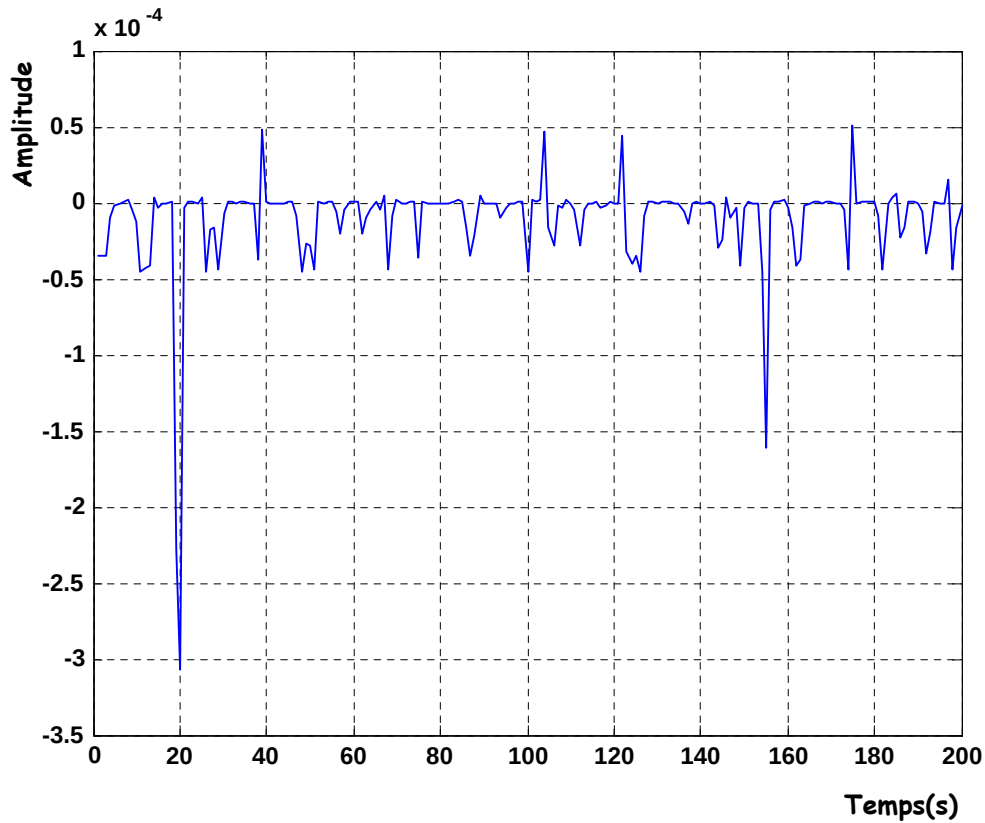


Figure (IV.18) Erreur

IV.2.2 Identification du modèle inverse

On va entraîner un réseau de neurones à approcher le modèle inverse du système $u_d(k)$, on utilise la structure d'apprentissage généralisé.

$$u_d(k) = -180 \cdot \sin(2\pi \cdot k / 7)$$

Le modèle neuronal choisi comporte :

- 02 couches cachées.
- 07 neurones dans la couche d'entrée.
- 05 neurones dans la première couche cachée.
- 03 neurones dans la deuxième couche cachée.
- Un seul neurone dans la couche de sortie.
- La fonction d'activation utilisée dans la couche d'entrée et les deux couches

cachées est la fonction sigmoïde. $f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$

- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire $f(x) = x$
- Pas d'apprentissage $\mu = 0.05$
- Moment $\beta = 0.00001$

La sortie du système est appliquée au modèle neuronal, l'erreur entre la commande et la sortie du modèle est utilisée pour l'ajustement des poids.

Après 10000 itérations, l'entrée du système et la sortie de modèle, et l'erreur entre la sortie de modèle et l'entrée de système sont représentés respectivement par la figure (IV.19) et figure (IV.20)

On remarque que l'écart entre les deux sorties est négligeable $3,19 \cdot 10^{-5}$

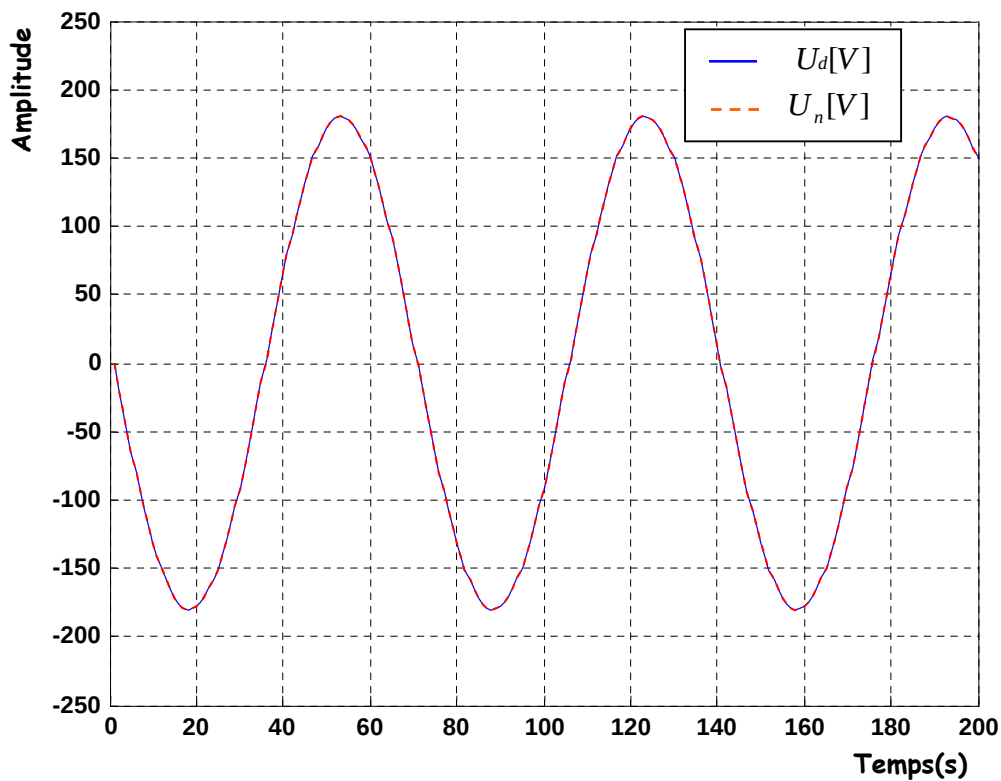


Figure (IV.19) Identification du modèle inverse

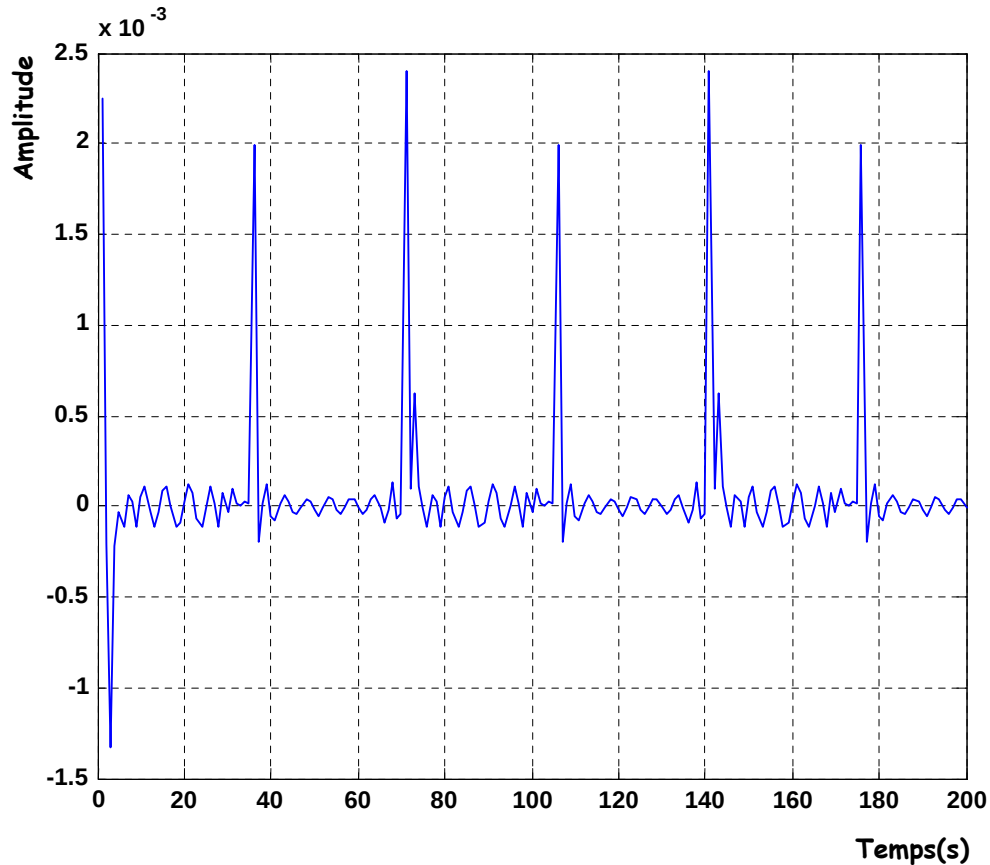


Figure (IV.20) Erreur

➤ Test de modèle inverse

Le modèle inverse a été testé, en appliquant à l'entrée du modèle la sortie du système tel que la commande appliquée au système $u(k)$:

$$u_1(k) = 80.(\cos(2.\pi.k / 5) + \cos(2.\pi.k / 8))$$

$$u_2(k) = 100.(\sin(2.\pi.k / 5) + \sin(2.\pi.k / 2.5))$$

- Le résultat du test 1 ($u_1(k)$) et l'erreur sont représentés par la figure (IV.21) et figure (IV.22).
- Le résultat du test 2 ($u_2(k)$) et l'erreur sont représentés par la figure (IV.23) et figure (IV.24).

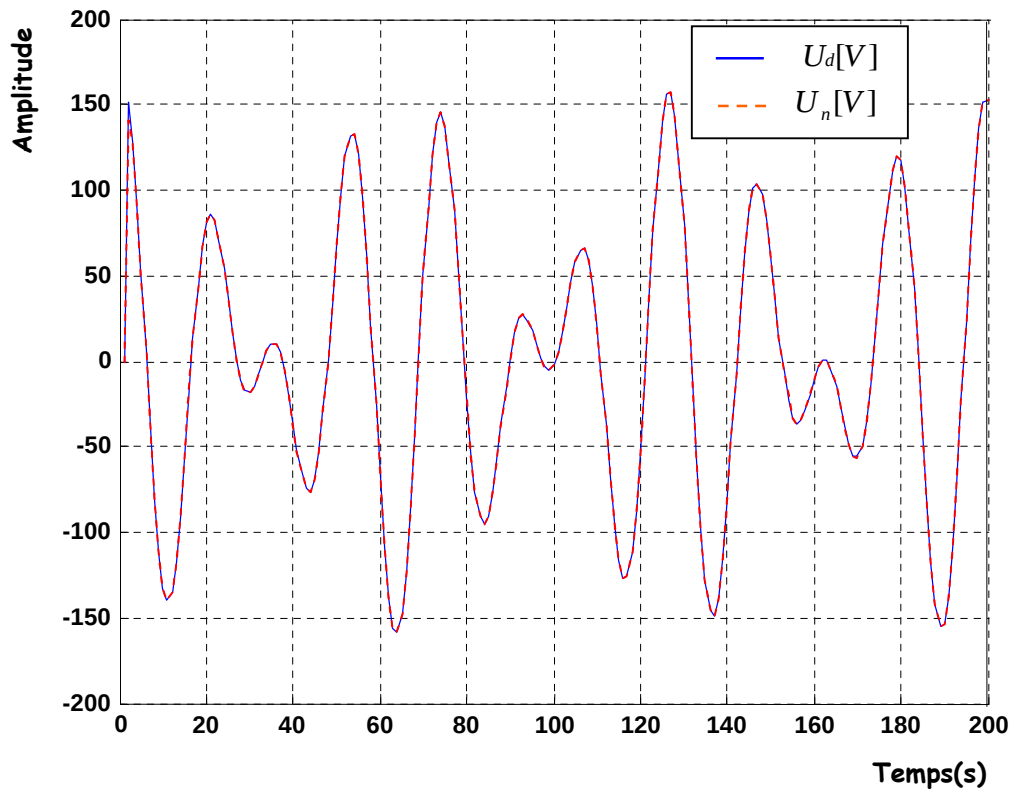


Figure (IV.21) Test 1 du modèle inverse

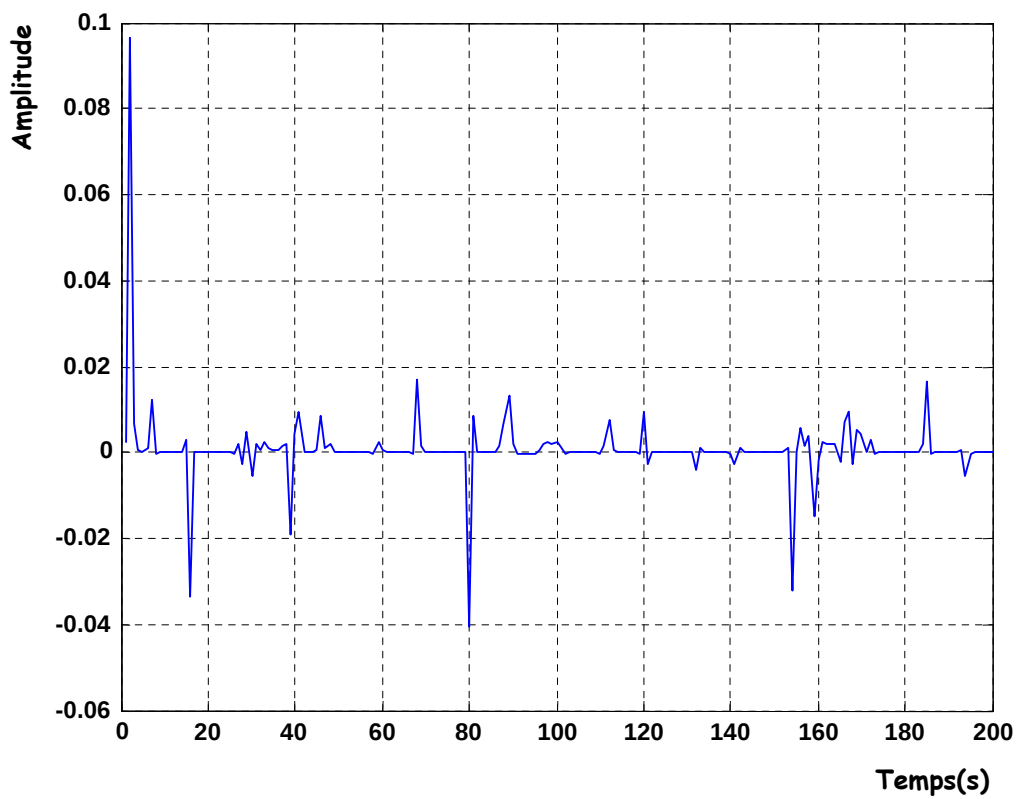


Figure (IV.22) Erreur

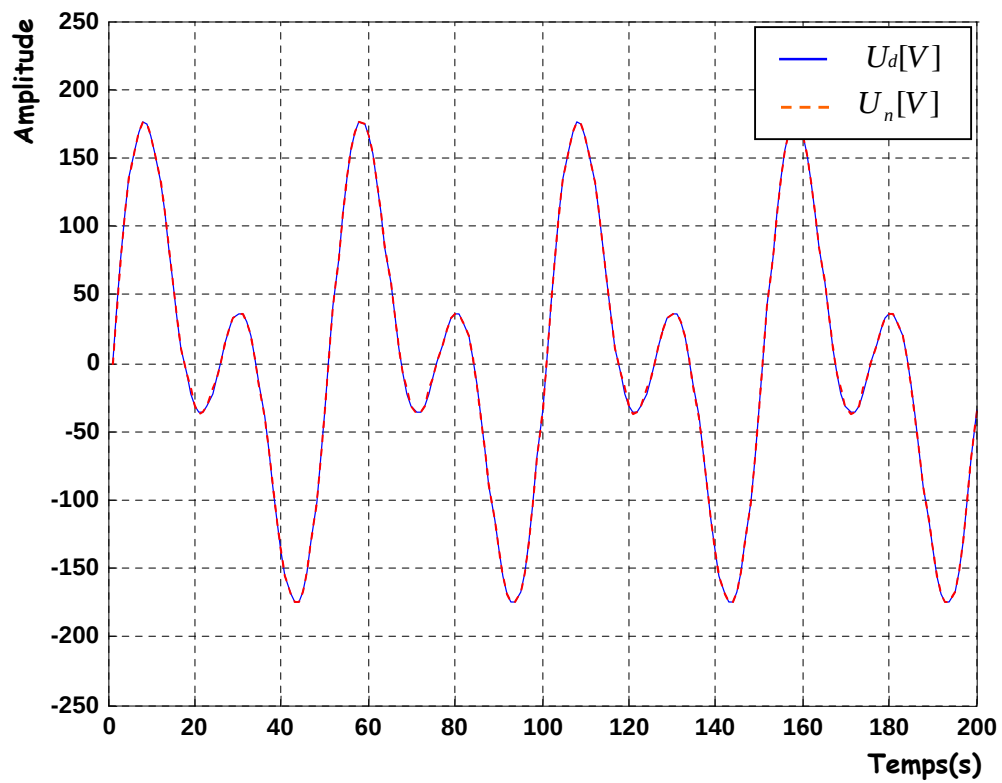


Figure (IV.23) Test 2 du modèle inverse

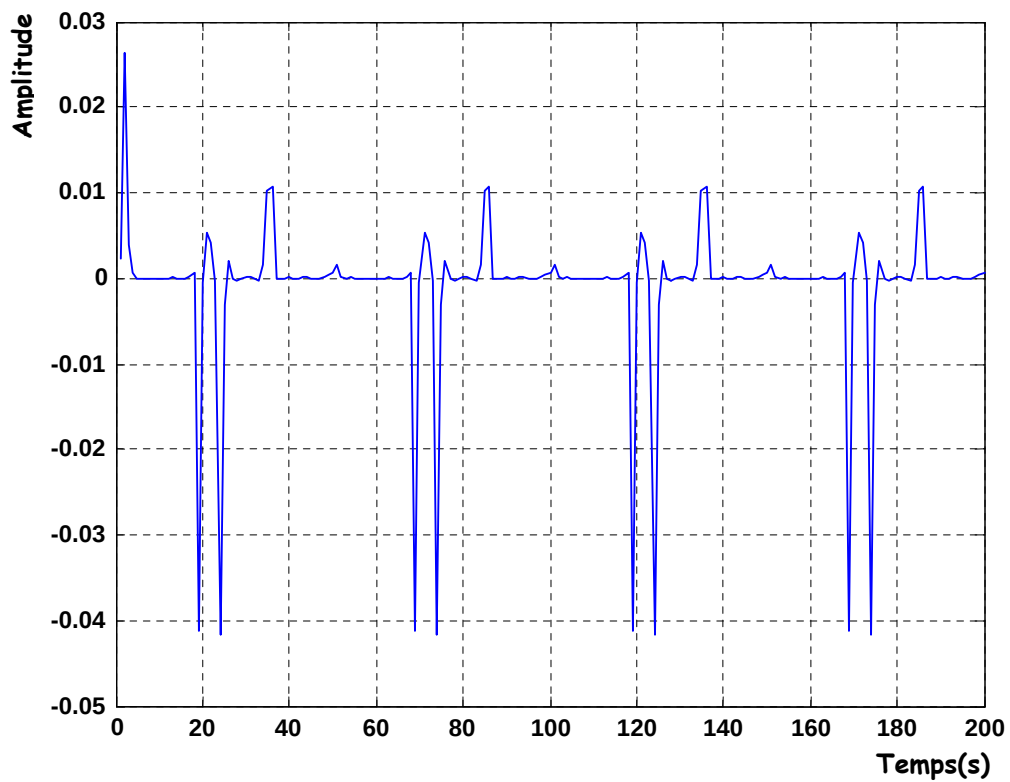


Figure (IV.24) Erreur

IV.2.3 Commande par modèle inverse

Le système peut être commandé avec la structure inverse traitée dans le chapitre II. Le modèle inverse du système obtenu dans IV.4.2 est placé en série avec le système.

La sortie de système pour des consignes échelon, rampe et $\theta_e(k)$ avec :

$$\theta_e(k) = 1.5 \sin(2\pi.k / 2.5) + 0.5 \cos(2\pi.k / 15) + 0.5 \cos(2\pi.k / 5)$$

Sont représentées par les figures, (IV.25), (IV.26), (IV.27) respectivement

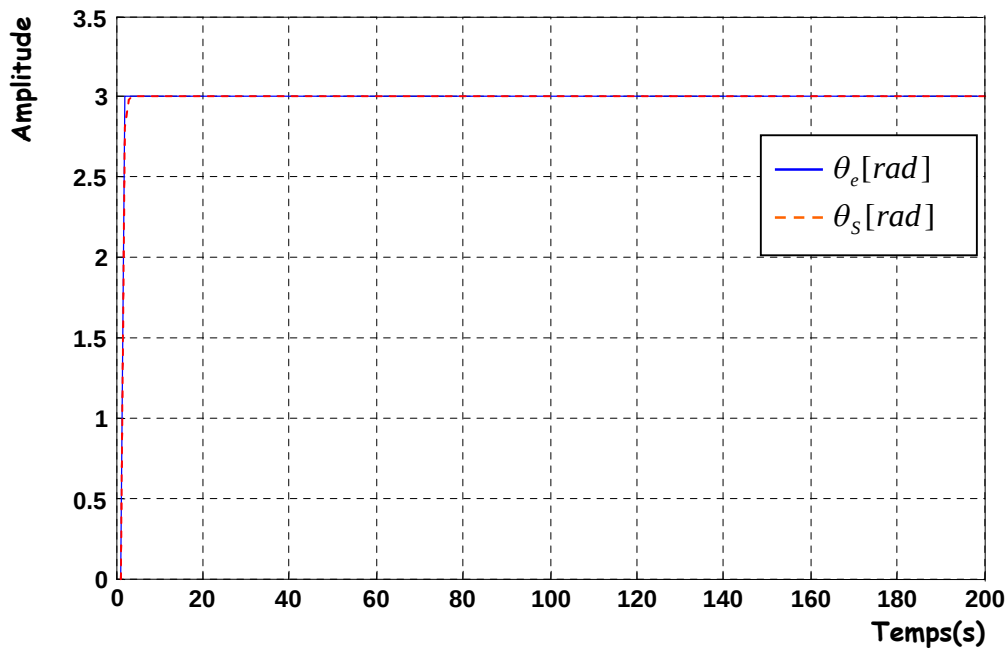


Figure (IV.25) Sortie du système pour une consigne échelon

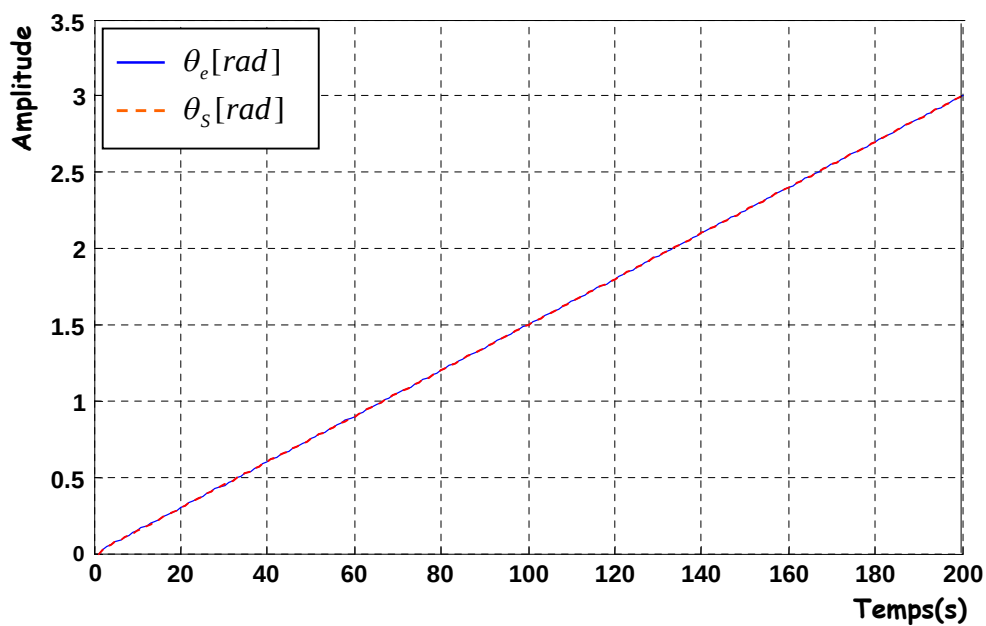


Figure (IV.26) Sortie du système pour une consigne rampe

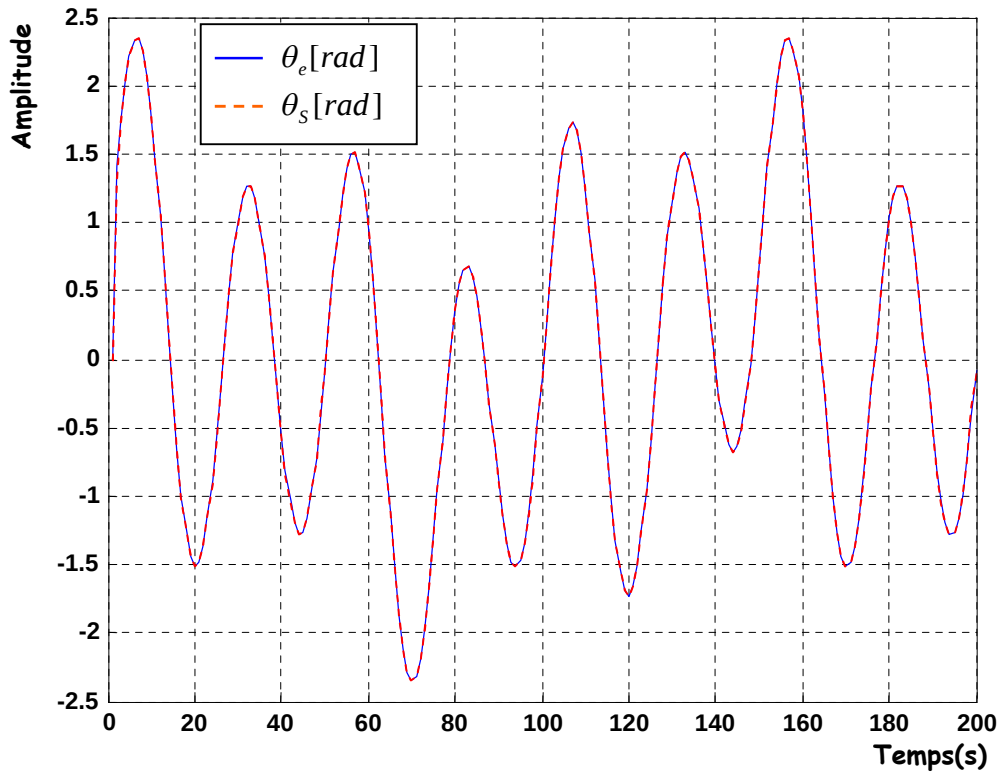


Figure (IV.27) Sortie du système pour une consigne θ_e

IV.2.4 Commande par modèle de référence

L'entrée de référence est $\theta_e(k)$.

L'objectif est de déterminer une commande $U(k)$ qui minimise l'écart entre la sortie du système et la sortie du modèle de référence.

Le contrôleur neuronal comporte :

- 01 couches cachées.
- 04 neurones dans la couche d'entrée.
- 07 neurones dans la couche cachée.
- Un seul neurone dans la couche de sortie..
- La fonction d'activation utilisée dans la couche d'entrée et les deux couches

cachées est la fonction sigmoïde.
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire
$$f(x) = x$$

Pour entraîner ce contrôleur le modèle neuronal du système obtenu dans IV.4.1 est placé en parallèle avec le système. Le rôle du modèle est la rétropropagation de l'erreur de sortie pour l'adaptation des poids du contrôleur neuronale.

L'adaptation des paramètres est effectuée pendant 14650 périodes.

La sortie du système et le modèle de référence pour des consignes échelon, rampe et $\theta_e(k)$ avec :

$$\theta_e(k) = 0.2(\sin(2.\pi.k / 2.5) + \cos(2.\pi.k / 15))$$

sont représentés par les figures, (IV.28), (IV.29), (IV.30) respectivement.

L'erreur entre la sortie du système et la consigne est de l'ordre de 10^{-4} .

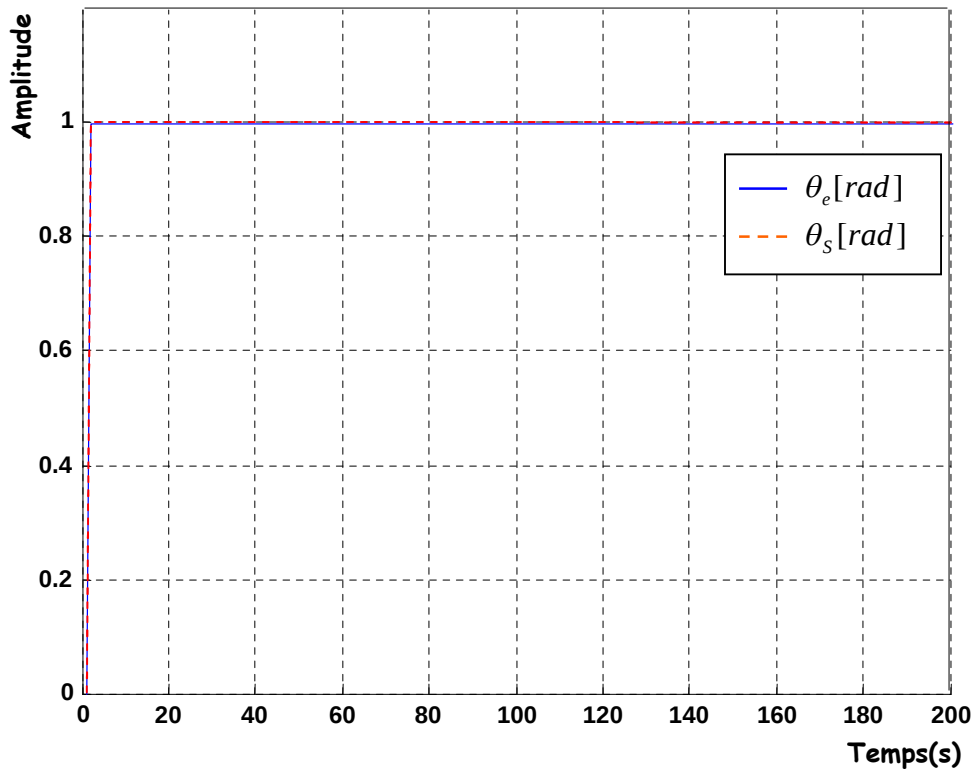


Figure (IV.28) Sortie du système pour une consigne échelon

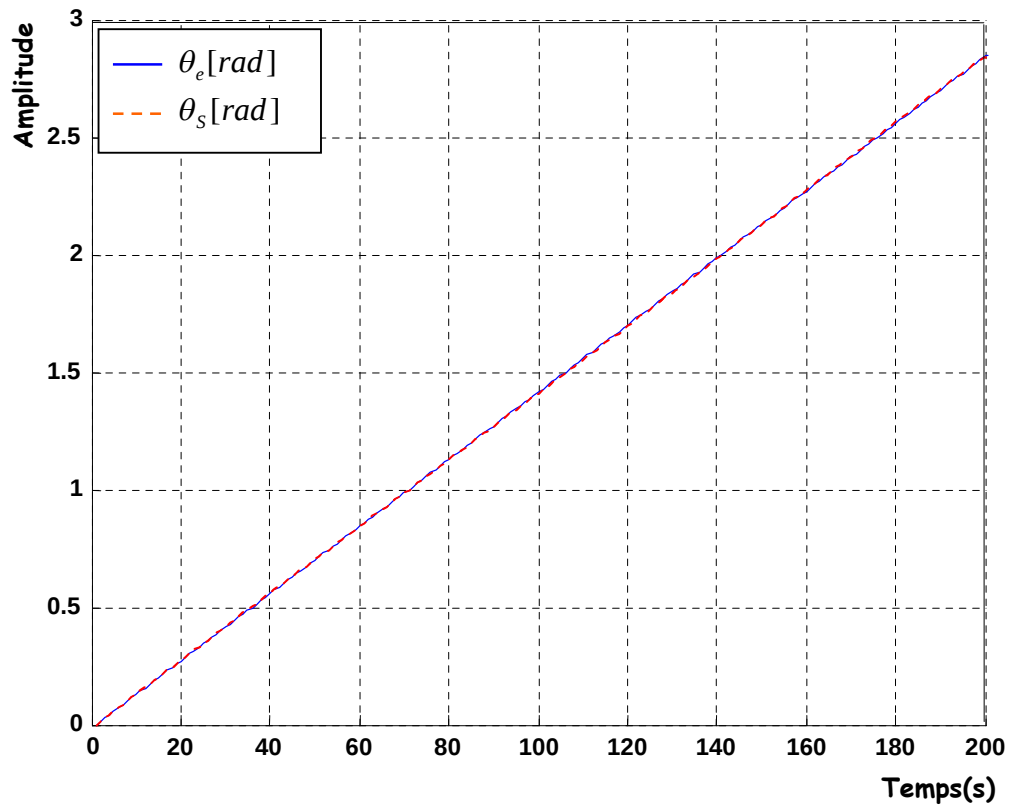
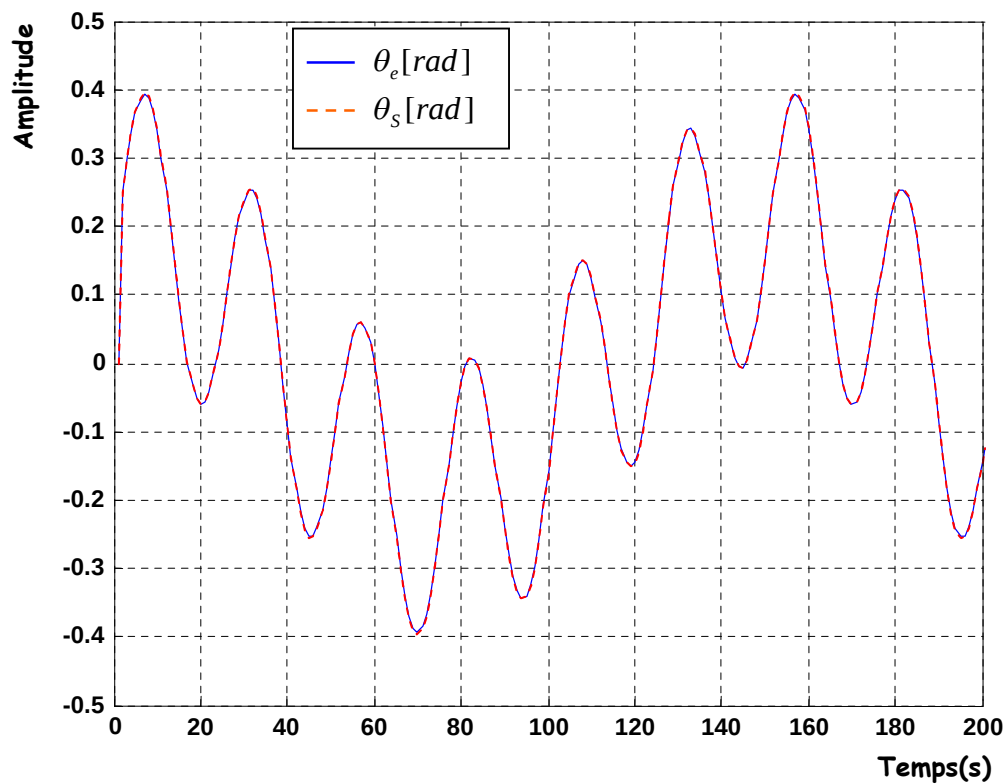


Figure (IV.29) Sortie du système pour une consigne rampe

Figure (IV.30) Sortie du système pour une consigne θ_e

IV.2.5 Conclusion

L'identification du modèle direct et inverse est obtenue après plusieurs essais car le réseau se stabilisait quelques fois à un niveau d'erreur inacceptable à cause du problème du minimum local ce qui nous obligeait d'exécuter le programme une autre fois avec un nouveau pas d'apprentissage.

Les modèles obtenus ont été testés sur un intervalle n'appartenant pas à l'ensemble d'apprentissage, les résultats obtenus confirment la capacité de la généralisation des réseaux de neurones et les structures de contrôle neuronal utilisées ont la possibilité d'engendrer les commandes nécessaires pour obtenir des réponses désirées d'une façon immédiate. Les résultats de simulation montrent que les structures de contrôle neuronal présentent de bonnes performances.