

Chapitre VI

LA COMMANDE FNN APPLIQUEE A UN BRAS MANIPULATEUR

La commande FNN appliquée à un bras manipulateur

VI.1. Introduction

Les équations dynamiques du mouvement d'un manipulateur sont hautement non linéaires et complexes et variable dans le temps [32]. Il est donc très difficile de mettre en œuvre un contrôle en temps réel basé sur un modèle dynamique détaillé d'un robot, et il n'est en aucun cas une tâche facile d'identifier les paramètres du modèle avec précision. Le problème de contrôle devient plus difficile si un contrôle adaptatif est nécessaire pour s'adapter aux conditions opérationnelles changeantes

Dans ce chapitre, nous avons développé deux réseaux neuro-flous employés dans deux régulateurs ANFIS, dont le but est de contrôler l'articulation d'un bras manipulateur à 2 D.D.L afin de poursuivre une trajectoire désirée. Les paramètres et la structure du réseau sont nécessaires dans toute modélisation visant à réaliser un modèle généralisé ; Le régulateur ANFIS utilise des techniques bien connues pour l'identification de paramètres, où la détermination du nombre optimal de règles floues sera déterminée par la structure du réseau [63] [11]. Pour la plupart des contrôleurs flous, la composition des règles est très difficile, elle nécessite la connaissance d'un expert humain dans la phase de conception [66]. Nous proposons la solution basée sur l'apprentissage supervisé, où chaque poids synaptique entre la troisième et la quatrième couche cachée de chaque réseau contrôleur représente une règle floue [61] [55]. Cependant, dans ce cas, de nouveaux poids et de nouveaux paramètres seront automatiquement adaptés et ajustés, par apprentissage en ligne, afin de réduire l'erreur de suivi de position.

La résolution de la fonction dynamique a été effectuée par deux méthodes comparatives, la méthode de R-K 4, et la méthode d'Euler. Enfin, le système contrôleur a été soumis à divers paramètres et différentes entrées, afin de caractériser les performances de suivi et de tester sa robustesse. Certains résultats sont illustrés pour montrer la validité de l'approche proposée.

VI.2. L'approche proposée

Dans cette approche, le système flou est utilisé pour exécuter le programme de contrôle, et le réseau neuronal est utilisé pour gérer et adapter les paramètres spécifiant les règles floues, en utilisant l'apprentissage et la reconnaissance des trajectoires [23], [14]. Le système flou encode les connaissances (les entrées et les sorties) par des règles qualitatives plutôt que par une description quantitative précise [23]. Cependant, l'utilisation des règles qualitatives floues est plus efficace et tend à couvrir un espace d'apprentissage plus large [67]. Le contrôleur FNN représenté sur la Fig (VI.1) ajuste la commande via une rétroaction sensorielle des sorties et emploie le couple résultant pour déplacer le bras manipulateur. En premier, nous avons un mouvement de référence, ce mouvement est défini par l'angle et la vitesse angulaire attribués à chaque articulation des deux segments. Ces données échantillonnées sont ensuite traitées de façon parallèle par deux contrôleurs FNN pour produire deux sorties, couple (τ_1) et (τ_2), appliquées au bras manipulateur qui est simulé par sa fonction dynamique. Enfin, un retour sensoriel est effectué afin de calculer l'erreur en position et en vitesse angulaire. Les paramètres spécifiant les règles floues sont mémorisés et manipulés par le FNN pour traiter une large gamme de mouvements. La figure (VI.1) présente le schéma bloc de l'approche proposée.

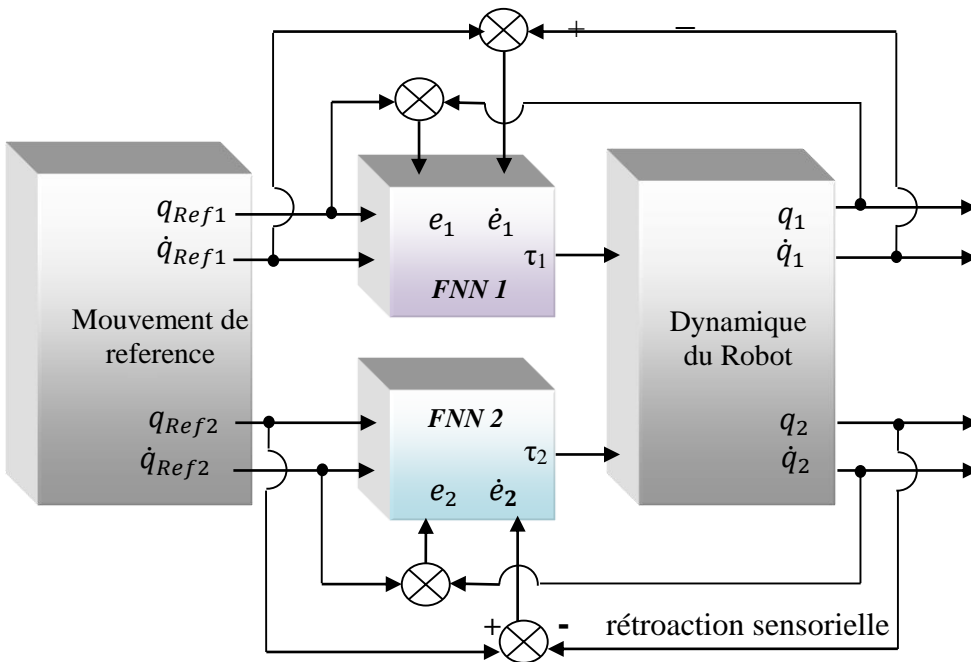


Fig. VI.1 : Schéma bloc de la commande de chaque articulation

Le bon fonctionnement du modèle proposé dépend de la conception du FNN qui gère les commandes appliquées à différents mouvements. Comme indiqué sur le schéma bloc, nous employons l'erreur en position (e), et l'erreur en vitesse (\dot{e}) entre les mouvements désirés et réels, tels que :

$$e = \frac{1}{2} \sum_{j=1}^n (q_{Refj} - q_j)^2 \quad (6.1)$$

$$\dot{e} = \frac{1}{2} \sum_{j=1}^n (\dot{q}_{Refj} - \dot{q}_j)^2 \quad (6.2)$$

La performance d'exécution du modèle proposé est évaluée tout en diminuant l'erreur entre la trajectoire désirée et la trajectoire dépitée par le bras manipulateur.

VI.3. Conception du contrôleur

Comme nous l'avons mentionné, Le contrôle d'un bras manipulateur à 2 D.D.L nécessite la conception de deux réseaux contrôleurs, là où chaque réseau contrôlera l'articulation d'un segment afin de poursuivre une des trajectoires référentielle.

ANFIS est une approche neuronale floue bien connue dans la modélisation des problèmes non linéaires complexes [61, 62]. Il faut noter que l'application de l'ANFIS ne se limite pas uniquement aux systèmes difficiles à modéliser, mais que nous pouvons également l'utiliser pour concevoir des contrôleurs beaucoup plus faciles. Les variables d'entrées représentent l'angle (q_{Ref}) et la vitesse angulaire (\dot{q}_{Ref}) du mouvement référence échantillonné, la sortie ou bien la commande représente le couple (τ) qui correspond à ces entrées. Le type de processus d'inférence flou utilisé est celui de Takagi Sugeno, dont la conséquence d'une règle floue n'est pas un ensemble flou mais une fonction des entrées [68]. Ce système d'inférence est convenable à notre processus car la sortie de l'ANFIS dépend des entrées, contrairement au système d'inférence de Mamdani là où la sortie ne dépend pas des entrées [57]. Le réseau utilisé est un réseau de neurones multi couches, composé d'une couche d'entrée passive, une couche de sortie qui correspond à la décision, et trois couches cachées comme le montre la figure (VI.2).

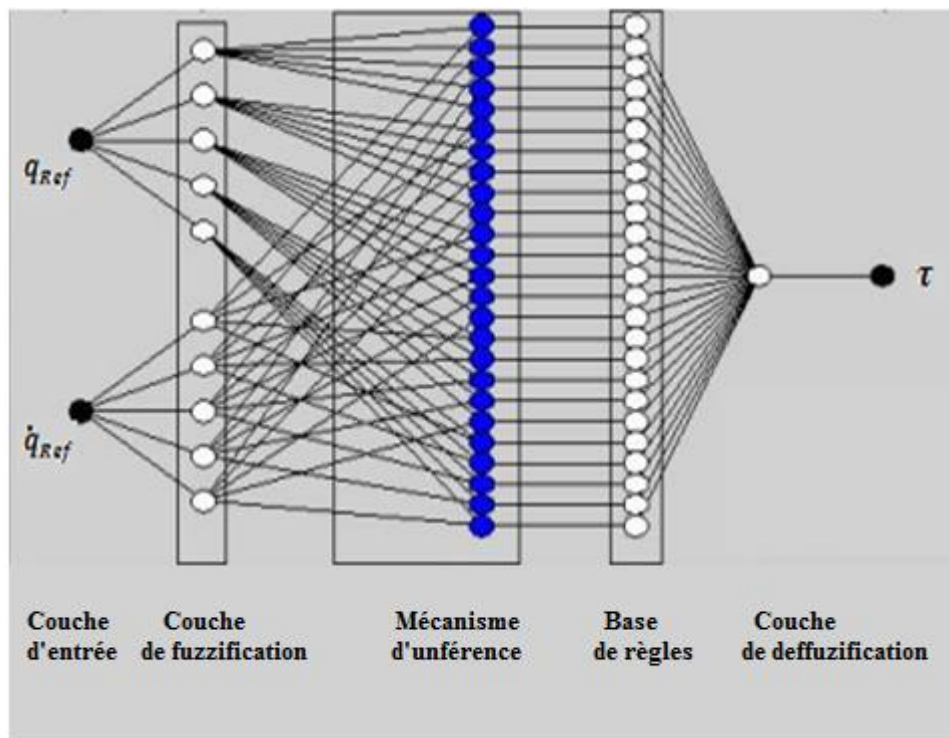


Fig. VI.2 : Structure du réseau contrôleur FNN

Cette conception nécessite le choix des paramètres suivants :

VI.2.1. Choix de la partition floue

Le choix de la partition floue est très important, il consiste à déterminer le nombre de termes dans un univers de discours pour les variables d'entrées et de sortie.

Généralement, dans la littérature nous trouvons, trois, cinq ou sept sous-ensembles flous, cela dépend du problème à traiter, s'il exige la précision ou la robustesse du système [36] [42]. Le suivi d'une trajectoire doit se faire avec une grande précision, le choix de trois sous-ensembles flous pour chaque entrée impliquera neuf règles floues, c.-à-d. neuf neurones dans la couche de mécanisme d'inférence et neuf neurones dans la couche de règles, ainsi les calculs seront rapide mais la poursuite de trajectoire manquera de précision. Tandis que le choix de sept sous-ensembles flous pour chaque entrée impliquera quarante-neuf règles floues, donc quarante-neuf neurones dans la troisième couche et dans la quatrième couche cachée, cela produira une grande précision de poursuite mais avec un temps de retard de réponse, dû aux nombres énormes de calculs effectués par le réseau, ce problème peut être surmonté en utilisant un GPU (Graphics Processing Unit)

contenant des milliers d'unités de calcul contre une dizaine pour un CPU [69]. Suite aux choix précédents, le nombre de sous-ensembles flous des deux entrées a été fixé à cinq, afin d'éviter le retard de réponse, et afin d'obtenir une poursuite satisfaisante.

Les partitions floues que nous avons choisies sont :

- NG : négative grand, NM : négative moyen, Z : zéro, PM : positive moyen, PG : positive grand, pour chaque entrée ce qui implique : $5 \times 5 = 25$ règles pour chaque réseau contrôleur.
- Sept partitions floues NG, NM, NP : négative petit, Z, PP : positive petit, PM, PG pour chaque sortie. Ce nombre n'affectera pas l'architecture du réseau et nous permettra d'obtenir une grande précision au niveau de la commande.

Pour ces choix judicieux, les FNNs proposés prennent l'appellation (557-557).

VI.2.2. Choix des fonctions d'appartenances des entrées et de la sortie

Les fonctions d'appartenances des variables d'entrées et de la sortie sont des fonctions gaussiennes, le choix de cette fonction revient à la facilité et rapidité de sa dérivée par rapport aux autres fonctions [12]. Les fonctions d'appartenances utilisées dans notre travail sont représentées dans la figure (VI.3).

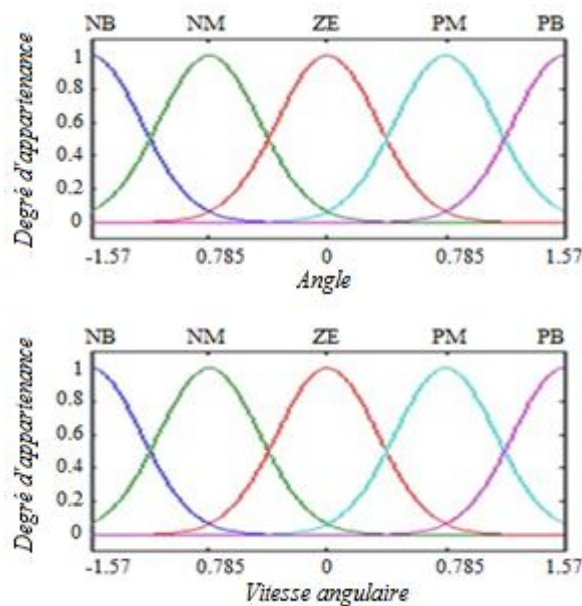


Fig. VI.3 : Formes et degrés d'activations des fonctions d'appartenances.

La valeur de l'écart type de la fonction gaussienne est de 1.57 rad. Le choix de cette valeur revient au but d'imiter l'articulation du bras humain, qui normalement varie dans l'intervalle $[-1.57\text{rad}, 1.57\text{rad}]$. Le taux de chevauchement des variables est de 50%. Ce choix revient à ce que cette valeur doit être suffisante pour permettre une description continue des variables mais pas trop importante pour limiter l'imprécision. Il est notamment préférable d'éviter que les fonctions d'appartenance de deux prédicats voisins soient simultanément égales à 1.

VI.2.3. Détermination de la base des règles

La base de règles est généralement définie par un tableau [39] [70]. Les deux bases de règles que nous avons utilisé sont représentées dans les tableaux suivants. La base de règles de chaque contrôleur a été réalisée après plusieurs tests de simulation, car réellement cette base de règles doit être concrétisée après plusieurs tests expérimentaux effectués par des experts. Les tableaux (VI.1) et (VI.2), montrent les bases de règles utilisées dans les deux réseaux contrôleurs respectivement.

\dot{q}_{Ref1}	NG	NM	ZE	PM	PG
q_{Ref1}					
NG	ZE	PP	NG	NG	NG
NM	PP	NP	ZE	PP	NM
ZE	PM	NG	PG	PM	PG
PM	PM	ZE	NM	NG	PP
PG	NP	NP	PG	PG	ZE

Tableau VI.1 : Base de règles du FNN 1

\dot{q}_{Ref2}	NG	NM	ZE	PM	PG
q_{Ref2}					
NG	NG	ZE	PG	NP	NP
NM	NP	PP	PM	NG	ZE
ZE	PM	PM	PP	PG	ZE
PM	NM	NG	PP	NP	NM
PG	PG	PM	PM	ZE	PP

Tableau VI.2 : Base de règles du FNN 2

Exemple : Si (q_{Ref2} est NG) et (\dot{q}_{Ref2} est NM) alors (τ_2 est ZE)

Comme nous l'avons mentionné précédemment, chaque réseau est composé de cinq couches de neurones, là où chacune d'entre elles exécute une étape du processus d'inférence flou, comme décrit ci-dessous.

Couche 1 : Cette couche est la couche d'entrée, d'où les entrées sont transmises à la prochaine couche directement sans aucun calcul. Dans la Figure (IV.2) nous avons deux neurones pour deux entrées.

Couche 2 : couche de fuzzification, Cette couche calcule le degré d'appartenance de chaque entrée. La sortie de chaque neurone est :

$$O_j^2 = \mu(x)$$

$$\mu(x) = \exp\left[-\left(\frac{x - \theta}{\sigma}\right)^2\right] \quad (6.3)$$

Où :

O_j^2 : la sortie du j-ème neurone de la deuxième couche du réseau

μ : degré d'appartenance de la fonction gaussienne

x : l'entrée du neurone j

θ : centre des fonctions d'appartenance

σ : écart type des fonctions d'appartenance

Couche 3 : Cette couche est prévue pour l'exécution des règles. Chaque neurone dans cette couche correspond à une règle floue. Dans cette couche, les fonctions d'appartenance sont multipliées car le « **et logique** » dans une règle floue est traduit par l'implication de Larsen [36] une multiplication en mathématique, donc la sortie de chaque neurone est :

$$O_j^3 = \mu(q_{Ref}) \cdot \mu(\dot{q}_{Ref}) \quad (6.4)$$

Couche 4 : La commande appliquée au bras manipulateur dépend des entrées précédentes, le système d'inférences approprié dans ce cas est celui de T-S [45]. L'utilisation de ce système nous permet d'obtenir une combinaison entre la sortie et les entrées [13].

Généralement la fonction de Takagi-Sugeno, possède une forme polynomiale des entrées. x_1, x_2, \dots, x_m .

$$f_j(x_1, x_2, \dots, x_m) = p_{0j} + p_{1j}x_1 + \dots + p_{mj}x_m \quad (6.5)$$

La règle à générer à partir de l'ensemble de ces données est comme suit :

$$R^i : \mathbf{Si} [x_1(k) \text{ est } A_1^i] \mathbf{et} \dots \mathbf{et} [x_{nx}(k) \text{ est } A_{nx}^i]$$

$$\mathbf{alors} \ y^i(k) = p_0^i + p_1^i x_1(k) + \dots + p_{nx}^i x_{nx}(k)$$

$$i = 0, 1, \dots, nr$$

Tel que :

R^i : i-ème règle d'inférence.

x_{nx} : n-ième variable d'entrée.

A_{nx}^i : Ensemble flou défini sur l'univers du discours de la variable x_{nx} , utilisée dans la i-ème règle d'inférence.

y^i : Sortie de l'i-ème neurone.

$p_{0i}, p_{1i}, \dots, p_{mi}$: Les coefficients de la fonction de T-S.

nr : Nombre de règles floues.

k : Valeur d'itération.

Ainsi, la sortie de chaque neurone dans cette couche est :

$$O_j^4 = O_j^3 f_j = O_j^3 \cdot (p_{0j} + p_{1j} \cdot q_{Ref} + p_{2j} \cdot \dot{q}_{Ref}) \quad (6.6)$$

Couche 5 : Pour des valeurs d'entrées $x_j(k)$, la sortie finale est déduite en prenant la moyenne pondérée des sorties de la quatrième couche [36].

$$\hat{y}(k+1) = \frac{\sum_{i=1}^{nr} K_i \cdot \mu^i(x(k)) \cdot y^i(k)}{\sum_{i=1}^{nr} \mu^i(x(k))} \quad (6.7)$$

$$\mu^i(x(k)) = \prod_{j=1}^{nx} \mu_j^i(x_j(k)) \quad (6.8)$$

$\mu^i(x(k))$ – degré d'activation de la i^{ème} règle

k_i – désigne les centres de gravité des ensembles flous de sortie associés à chaque règle.

Cela signifie que la conséquence de chaque règle floue décrit une articulation de chaque segment du bras manipulateur.

Dans cette couche, un seul neurone est nécessaire pour calculer la commande τ [46] :

$$\begin{aligned} O_j^5 = \tau &= \frac{\sum_j K_i O_j^4}{\sum_j O_j^3} \\ &= \frac{\sum_j K_i O_j^3 f_j}{\sum_j O_j^3} \end{aligned} \quad (6.9)$$

Dans chaque contrôleur FNN, nous avons deux neurones dans la couche 1, dix neurones dans la couche 2, vingt-cinq neurones dans les couches 3 et 4, et un neurone dans la couche 5. Par conséquent, un total de 125 paramètres doit être adapté pour chaque segment pendant l'apprentissage. Car entre la deuxième et la troisième couche nous avons $5 \cdot 10 = 50$ poids synaptiques, et entre la troisième et la quatrième couche nous avons 25 poids synaptiques à adapter, enfin, chaque neurone de la couche 4 a besoin de trois paramètres p_{0j} , p_{1j} et p_{2j} , où $O_j^4 = O_j^3 \cdot (p_{0j} + p_{1j} \cdot q_{Ref} + p_{2j} \cdot \dot{q}_{Ref})$, pour simplifier les calculs nous prenons $p_{0j} = 0$. Donc au total nous avons : $50 + 25 + 50 = 125$ paramètres à définir pour chaque réseau.

Généralement, pour l'adaptation des coefficients de fonction T-S, nous utilisons la méthode des algorithmes génétiques qui est un algorithme d'optimisation [51] [64], ensuite nous les injectons dans le réseau. Dans les étapes d'optimisation, qui est un domaine des mathématiques, le but est de trouver le meilleur membre selon un critère, parmi un ensemble de variables d'entrées [19]. Cette méthode comprend plusieurs étapes et prend beaucoup de temps et de calculs ; afin de surmonter ce problème une nouvelle technique est utilisée, l'idée est simple : Généralement, les poids synaptiques dans un contrôleur FNN n'ont aucune signification physique.

Dans notre travail, grâce à l'architecture du réseau, nous pouvons prendre les poids entre la deuxième et la troisième couche cachée, sous forme des paramètres de la fonction de T-S, ces paramètres sont adaptés pour être généralisés par le réseau afin de traiter un mouvement désiré; et puisque la première couche du réseau est une couche passive, cela implique que les paramètres concernés dans la phase d'apprentissage de chaque FNN, sont les poids synaptiques entre la deuxième et la troisième couche cachée, ainsi que les poids entre la troisième et la quatrième couche du réseau. Tandis que les poids entre la quatrième couche et la cinquième couche représentent les centres des fonctions d'appartenances de la sortie. Cette technique nous permet de réduire le nombre de 125 à 75 paramètres.

IV.4. La méthode d'apprentissage

Le but de l'apprentissage du contrôleur est d'adapter ces poids synaptiques, afin de générer un vecteur de contrôle τ correspondant à un mouvement échantillonné. Différentes techniques d'apprentissage ont été développées pour le réglage des entrées et des paramètres conséquents dans l'ANFIS [51] [53]. Dans cette étude, nous avons utilisé l'apprentissage supervisé, car la trajectoire dépitée par le bras manipulateur est connue à l'avance. L'algorithme adapté à cette méthode est l'algorithme de rétro-propagation [54] [55].

Le principe de cette méthode est que, de même que l'on est capable de propager un signal provenant de cellules d'entrées vers la couche de sortie, on peut, en suivant le chemin inverse, rétro-propager l'erreur commise en sortie vers les couches internes [55, 56]. Ceci permet de calculer les erreurs sur chaque neurone dans les couches cachées, et les poids sont ajustés pour but de minimiser une fonction de coût donnée par :

$$e(\omega) = \frac{1}{2} \sum_{j=1}^n (y_{Ref} - y)^2 \quad (6.10)$$

La loi d'adaptation des poids est donnée par la relation suivante :

$$W_{ij}^k(t+1) = W_{ij}^k(t) - \mu \frac{\partial e(W)}{\partial W_{ij}^k(t)} + \alpha (W_{ij}^k(t) - W_{ij}^k(t-1)) \quad (6.11)$$

$W_{ij}^k(t)$: Le poids entre le neurone j de la couche k et le neurone i de la couche $(k-1)$

t : Indice d'itération

α : Le momentum, avec $0 \leq \alpha < 1$

μ : Le pas d'apprentissage

La sortie du neurone j appartenant à la couche k est donnée par l'expression suivante :

$$y_j^k(t) = f(S_j^k(t)) \quad (6.12)$$

$$S_j^k(t) = \sum_{i=1}^{n_k} W_{ij}^k y_i^{k-1}(t) \quad (6.13)$$

f : la fonction d'activation du neurone

$S_j^k(t)$: L'entrée du neurone j

n_k : le nombre de neurones dans la couche k .

y_i^{k-1} : La sortie du neurone i dans la couche $k-1$.

Mathématiquement, la minimisation de l'erreur se fait par une approximation d'une descente de gradient, en calculant la dérivée de l'erreur quadratique par rapport à un poids donné, et cela en utilisant les règles de dérivation composé.

$$\frac{\partial e(W)}{\partial W_{ij}^k(t)} = \frac{\partial e(W)}{\partial y_j^k(t)} \cdot \frac{\partial y_j^k(t)}{\partial W_{ij}^k(t)} \quad (6.14)$$

$$\frac{\partial y_j^k(t)}{\partial W_{ij}^k(t)} = \frac{\partial f(S_j^k(t))}{\partial W_{ij}^k(t)} = f(S_j^k(t)) \cdot \frac{\partial S_j^k(t)}{\partial W_{ij}^k(t)}$$

Et :

$$\frac{\partial S_j^k(t)}{\partial W_{ij}^k(t)} = \frac{\partial \sum_{i=1}^{nk} W_{ij}^k \cdot y_i^{k-1}(t)}{\partial W_{ij}^k(t)} = y_i^{k-1}(t)$$

Donc :

$$\frac{\partial y_j^k(t)}{\partial W_{ij}^k(t)} = f(S_j^k(t)) \cdot y_i^{k-1}(t) \quad (6.15)$$

En substituant les résultats obtenus en (6.15) dans l'équation (6.14) nous obtenons :

$$\frac{\partial e(W)}{\partial W_{ij}^k(t)} = \frac{\partial e(W)}{\partial y_j^k(t)} \cdot f(S_j^k(t)) \cdot y_i^{k-1}(t) \quad (6.16)$$

avec :

$$S_j^k(t) = -\frac{\partial e(W)}{\partial y_j^k(t)} \cdot f(S_j^k(t))$$

Le deuxième terme de l'équation (6.13) satisfait :

$$\frac{\partial e(W)}{\partial W_{ij}^k(t)} = -S_j^k(t) \cdot y_i^{k-1}(t)$$

Par conséquent la loi d'adaptation des poids est donnée par la fonction suivante:

$$W_{ij}^k(t+1) = W_{ij}^k(t) + \mu S_j^k(t) \cdot y_i^{k-1}(t) + \alpha(W_{ij}^k(t) - W_{ij}^k(t-1)) \quad (6.17)$$

VI.5. Description du bras manipulateur

La figure (VI .4) représente schématiquement le prototype du bras manipulateur utilisé dans notre simulation, un système fortement non linéaire, instable et multi-variable. Comme il ressort de cette figure, le prototype est un bras avec deux liaisons reliées par des articulations rotoïdes, c'est-à-dire qu'il possède 2 D.D.L. Les deux articulations sont situées à «l'épaule» et au «coude». Il s'agit d'un mécanisme à entraînement direct, c'est-à-dire que les axes des moteurs sont connectés directement aux liaisons sans engrenages ni courroies [52] [17].

Le bras manipulateur est constitué de deux segments de longueurs l_1 et l_2 , de masses m_1 et m_2 respectivement. Les distances des axes de rotation aux centres de masse sont notées l_{c1} et l_{c2} pour les liaisons 1 et 2, respectivement. Le vecteur des positions d'articulation q est défini comme $q = [q_1 \ q_2]^T$. Enfin, I_1 et I_2 désignent les moments d'inertie des deux segments.

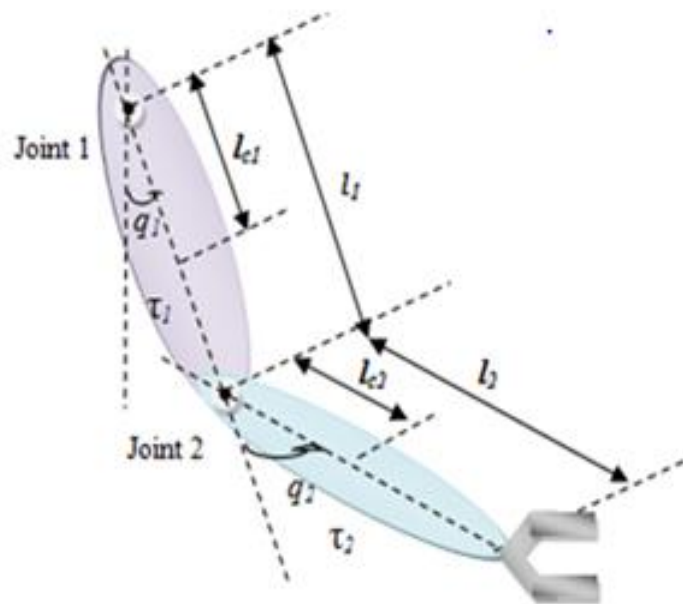


Fig. VI .4 : Structure du bras manipulateur à 2 D.D.L

L'équation dynamique de chaque segment est en fonction du couple τ , de la position q , de la vitesse angulaire \dot{q} , et de l'accélération \ddot{q} de chaque articulation [17] [67] :

$$\begin{aligned}\tau_1 &= H_{11}\ddot{q}_1 + H_{12}\ddot{q}_2 - H(\dot{q}_2^2 + 2\dot{q}_1\dot{q}_2) + G_1 \\ \tau_2 &= H_{21}\ddot{q}_1 + H_{22}\ddot{q}_2 + H\dot{q}_1^2 + G_2\end{aligned}\quad (6.15)$$

Avec :

$$\begin{aligned}H_{11} &= m_1 \cdot l_{c1}^2 + I_1 + m_2 \cdot (l_1^2 + l_{c2}^2 + 2 \cdot l_1 \cdot l_{c2} \cdot \cos q_2) + I_2 \\ H_{22} &= m_2 \cdot l_{c2}^2 + I_2 \\ H_{12} &= m_2 \cdot l_1 \cdot l_{c2} \cdot \cos q_2 + m_2 \cdot l_{c2}^2 + I_2 \\ H_{21} &= H_{12} \\ H &= m_2 \cdot l_1 \cdot l_{c2} \cdot \sin q_2 \\ G_1 &= m_1 \cdot l_{c1} \cdot g \cdot \cos q_1 + m_2 \cdot g \cdot (l_{c2}^2 + l_1 \cdot \cos q_1) \\ G_2 &= m_2 \cdot l_{c2} \cdot g \cdot \cos (q_1 + q_2)\end{aligned}\quad (6.16)$$

La résolution des équations dynamiques exige de les transformer sous forme de fonctions différentielles :

Pour la première articulation :

$$\left\{ \begin{array}{l} \frac{dq_1}{dt} = q_2 \\ \frac{dq_2}{dt} = q_3 \\ \frac{dq_3}{dt} = \frac{\tau_1 - H_{11} \cdot q_3 + H \cdot (q_3^2 + 2 \cdot q_2 \cdot q_3) - G_1}{H_{12}} \end{array} \right. \quad (6.17)$$

Pour la deuxième articulation :

$$\left\{ \begin{array}{l} \frac{dq_1}{dt} = q_2 \\ \frac{dq_2}{dt} = q_3 \\ \frac{dq_3}{dt} = \frac{\tau_2 - H_{21} \cdot q_3 + H \cdot q_2^2 - G_2}{H_{22}} \end{array} \right. \quad (6.18)$$

Ces fonctions sont dérivées en utilisant deux méthodes, la méthode d'Euler, et la méthode de R-K 4 [34, 35], Cette dernière a été bien détaillé dans le chapitre 2.

Par rapport à la méthode d'Euler, ce schéma numérique nécessite plus de calculs à chaque étape et donc un temps de calcul plus long mais avec des résultats plus précis.

L'algorithme de R-K 4 :

1. Initialisation de l'étape h , de durée T .
2. Initialisation des conditions initiales
3. Définition de la fonction $f(t, y)$
4. Tant que $t \leq T$ faire :
 - (a) Calcul à partir de $k_1 = f(t, y)$.
 - (b) Calcul à partir de $k_2 = f\left(t + \frac{1}{2}h, y + \frac{1}{2}hk_1\right)$.
 - (c) Calcul à partir de $k_3 = f\left(t + \frac{1}{2}h, y + \frac{1}{2}hk_2\right)$.
 - (d) Calcul à partir de $k_4 = f(t + h, y + hk_3)$.
 - (e) $y = y + \frac{h}{6}[k_1 + 2k_2 + 2k_3 + k_4]$; $t = t + h$.
 - (f) Enregistrer les données.

Où $f(t, y)$ est la fonction recherchée. On note h un pas basé sur la discrétisation de la variable t , dans notre cas :

$$T = 3 \text{ (s)}$$

$$h = 0.1 \text{ (s)}$$

$y = q$: Position angulaire de l'articulation

Ce qui implique 30 itérations pour la résolution de chaque fonction dynamique représentée par les fonctions (6.17) et (6.18).

Une étude comparative sera menée entre les deux méthodes afin de prouver l'efficacité de cette dernière.

VI.6. Résultats de la simulation

VI.6.1 Tests d'apprentissage

En général, plusieurs centaines de tests d'apprentissage ont été effectués afin que le bras manipulateur puisse poursuivre un mouvement prélevé. Les résultats de simulation, sont représentés dans les figures (VI.5) à (VI.18).

Les trajectoires références dépitées par chaque tige sont exprimées comme suit [71] :

$$\begin{cases} q_{Ref1}(t) = \frac{1}{2} \cdot \sin(2 \cdot \pi \cdot t) \\ q_{Ref2}(t) = -\frac{\pi}{2} + \frac{1}{2} \cdot \sin(2 \cdot \pi \cdot t) \end{cases} \quad t \leq 3 \quad (6.29)$$

Le temps de simulation t vari dans l'intervalle $[0s, 3s]$, t prend les valeurs d'échantillonnages du signal, qui sont représentées par les valeurs d'itérations dans la programmation. Dans notre cas le temps d'échantillonnage est égal à (0.1s) car :

$$\begin{aligned} t_{\text{échantillonnage}} &= \frac{t_{\text{final}} - t_{\text{initial}}}{\text{nombre d'itération}} \\ &= \frac{(3 - 0)s}{30} = 0.1s \end{aligned}$$

Les valeurs des paramètres du robot utilisées pour la simulation sont données dans le tableau (VI.3).

Variable	Valeur
L_1	0.32 m
L_2	0.24 m
L_{c1}	0.16 m
L_{c2}	0.12 m
m_1	1.8 Kg
m_2	1.2 Kg
I_1, I_2	0.02 Kg.m ²
g	9.8 N/Kg.

Tableau VI.3 : Paramètres du procédé

La figure (VI.5) présente la variation de l'angle du premier segment en utilisant la méthode de Runge-Kutta (**RK**), et la méthode d'Euler (**Eu**), afin de poursuivre une trajectoire référentielle (**Ref**).

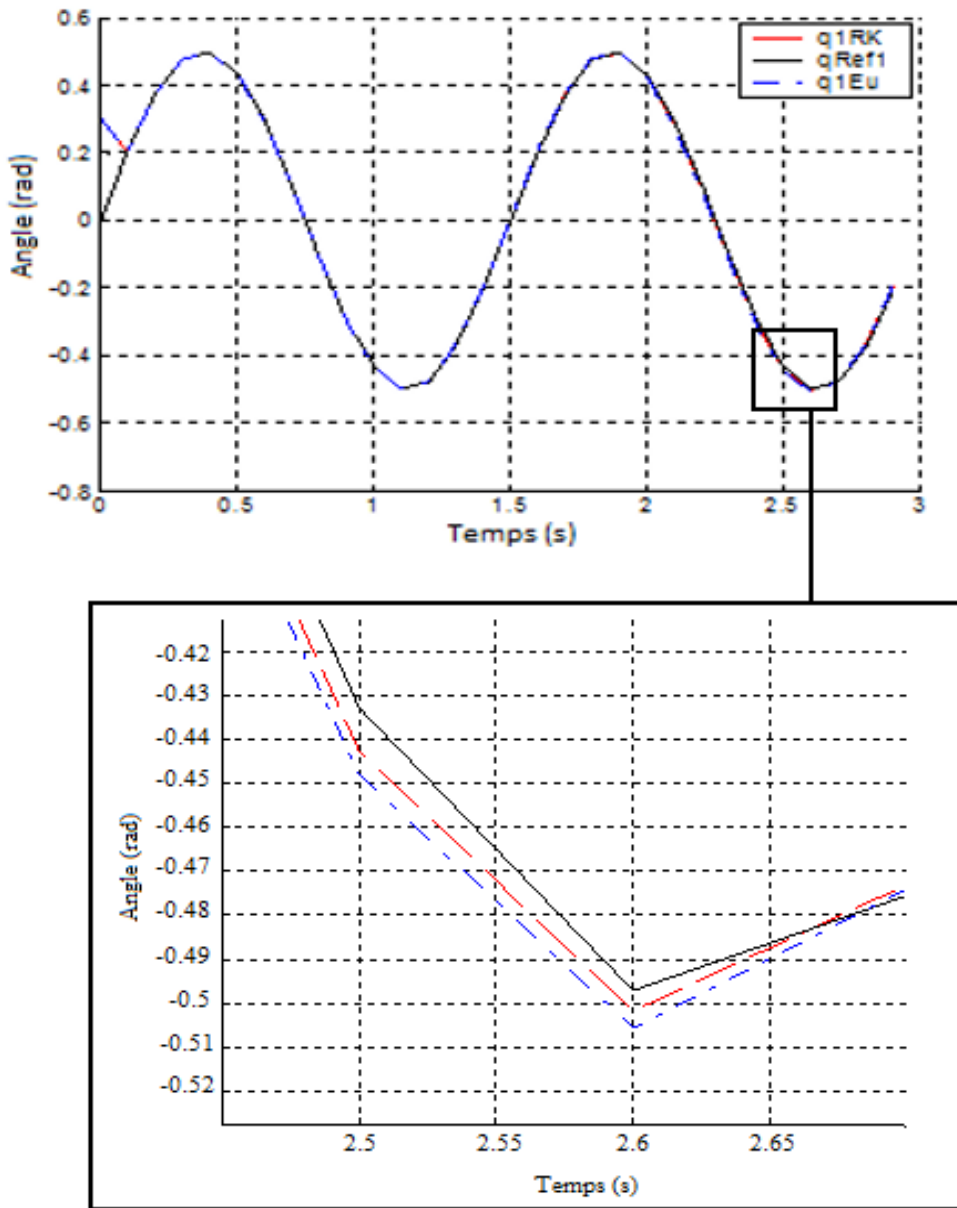


Fig VI .5 : L'articulation du segment-1 pour les conditions initiales

$$[q_1, \dot{q}_1, \ddot{q}_1, \tau]^T = [0.3, 0, 0, 0]$$

La figure (VI.5) illustre un cheminement favorable de la trajectoire désirée, exécutée par l'articulation du segment-1. Comme le montre cette figure, ce suivi de trajectoire est plus précis en utilisant la méthode de R-K 4 par rapport à la méthode d'Euler. Afin d'effectuer une étude comparative entre les deux méthodes citées précédemment, le signal d'erreur en position du premier segment est représenté dans la Figure (VI.6), et a été illustré par ses valeurs numériques dans le tableau (VI.4).

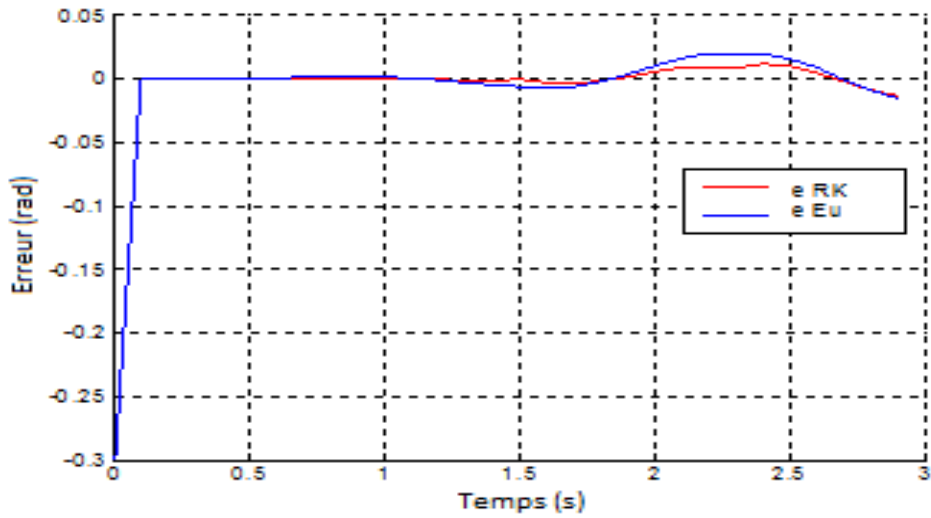


Fig. VI .6 : Erreur en position du segment 1

Temps (s)	Erreur en position avec la méthode de RK4 (e_{RK}) (rad)	Erreur en position avec la méthode d'Euler (e_{EU}) (rad)
0.1	-0.3000	-0.3000
0.2	-0.0000	-0.0000
0.3	-0.0000	-0.0001
0.4	-0.0000	-0.0001
0.5	0.0000	0.0000
0.6	0.0001	0.0004
0.7	0.0001	0.0008
0.8	0.0001	0.0013
0.9	0.0001	0.0014
1.0	0.0003	0.0016
1.1	0.0005	0.0013
1.2	0.0002	0.0004
1.3	-0.0006	0.0012
1.4	-0.0014	0.0031
1.5	-0.0020	0.0050
1.6	-0.0001	0.0059
1.7	-0.0034	0.0074
1.8	-0.0039	0.0062
1.9	-0.0021	0.0025
2.0	0.0016	0.0033
2.1	0.0056	0.0102
2.2	0.0087	0.0161
2.3	0.0078	0.0198
2.4	0.0088	0.0196
2.5	0.0117	0.0190
2.6	0.0098	0.0152
2.7	0.0046	0.0084
2.8	-0.0025	0.0014
2.9	-0.0091	0.0094
3.0	-0.0135	0.0155

Tableau VI.4 : Les valeurs numériques de l'erreur en position du segment-1

D'après les valeurs enregistrées dans le Tableau (IV.4), l'erreur moyenne en position du segment-1 en utilisant la méthode R-K 4 est de $3,10 \cdot 10^{-3}(\text{rad})$, et $(-6,33) \cdot 10^{-3}(\text{rad})$ en utilisant la méthode d'Euler.

La variation de la vitesse angulaire du segment-1 est représentée dans la figure suivante.

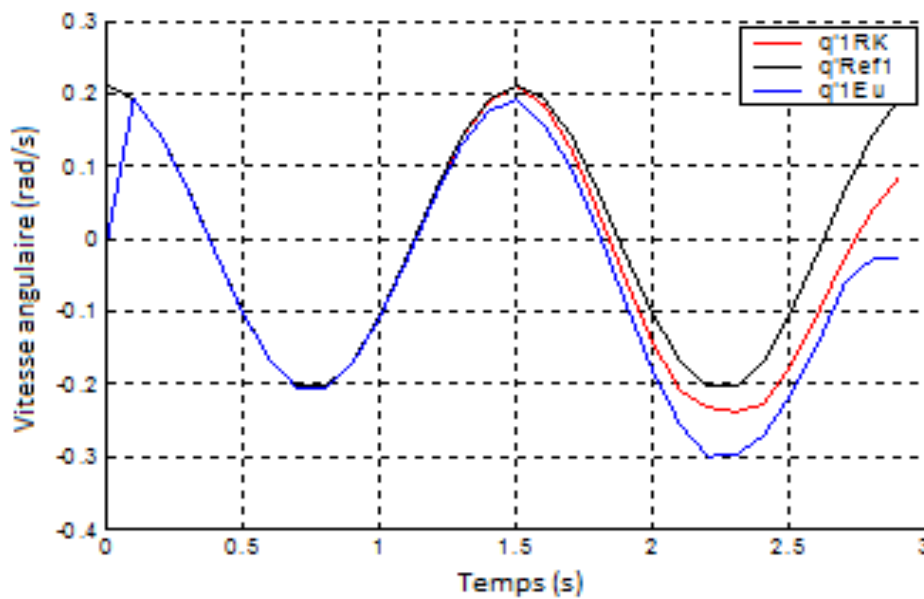


Fig VI.7 : Variation de la vitesse angulaire du segment-1

En raison de la flexibilité du bras manipulateur, des oscillations se produisent près des extrémités du segment [9] [22], comme le montre la figure (VI.7). Il faut noter que cette flexibilité influe plus sur la vitesse angulaire du segment que sur sa position. Cela est dû au principe d'inertie (première loi de Newton) [67], où le segment a tendance à conserver sa vitesse initiale.

Les figures (VI .8), (VI .9) et (VI .10) suivantes, présentent la variation de l'angle et de l'erreur angulaire, et de la vitesse angulaire du deuxième segment respectivement.

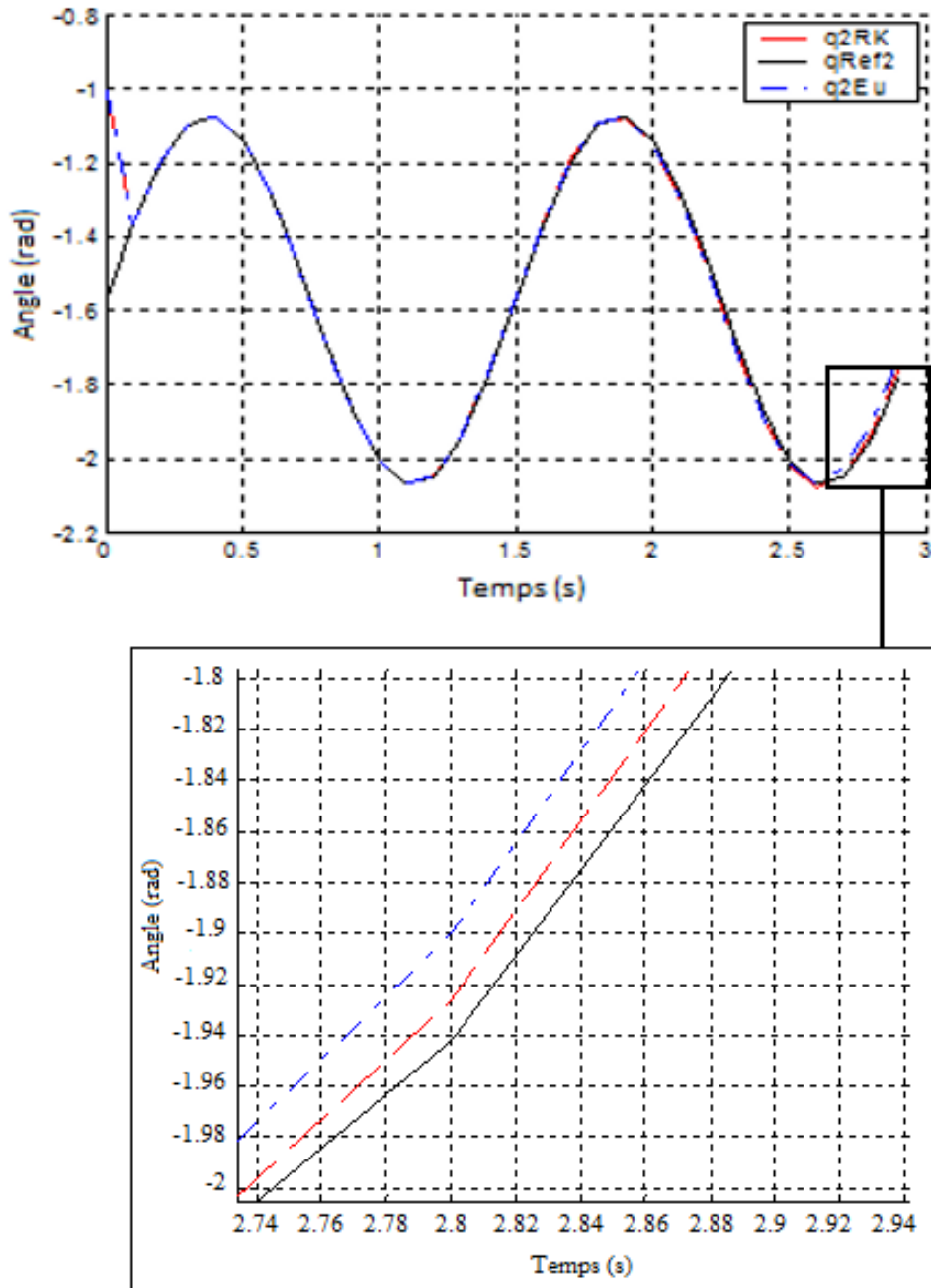


Fig VI .8 : L'articulation du segment-2 pour les conditions initiales

$$[q_2, \dot{q}_2, \ddot{q}_2, \tau]^T = [-1, 0, 0, 0]$$

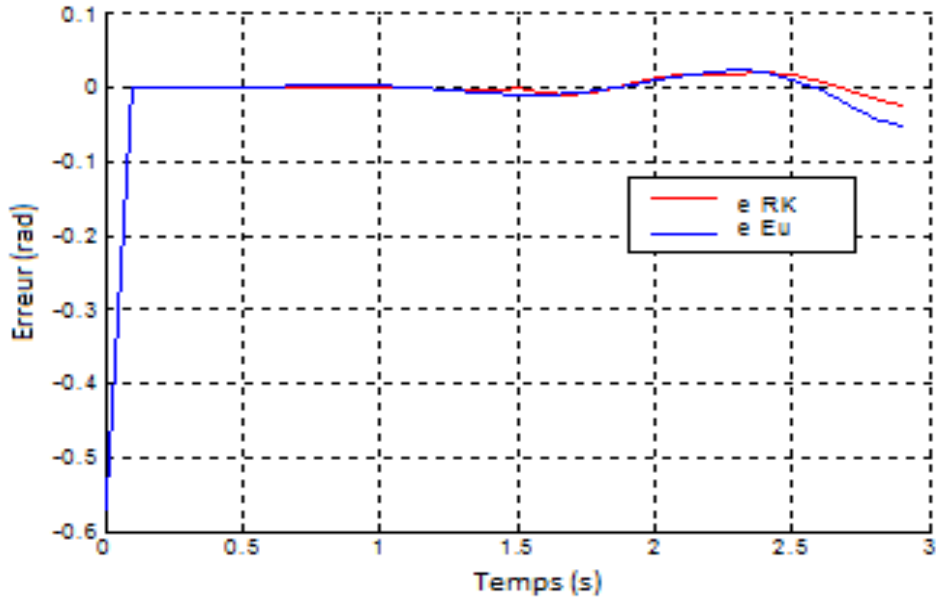


Fig VI .9 : Erreur en position du segment-2

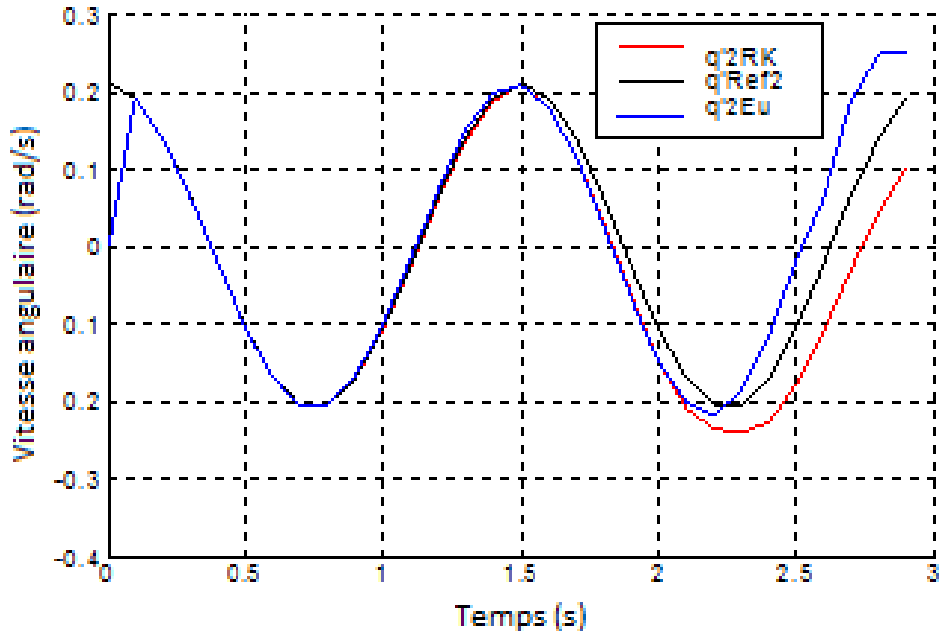


Fig VI .10 : Variation de la vitesse angulaire du segment-2

D'après les figures (VI.8), (VI.9) et (VI.10), le deuxième segment du bras manipulateur arrive à dépasser la deuxième trajectoire référentielle représentée par la fonction (6.29), avec une erreur proche de zéro qui varie quasiment dans l'intervalle $[10^{-5}, 10^{-3}]$ (rad).

par contre cette erreur est plus remarquable mais acceptable concernant la vitesse angulaire qui varie quasiment entre $[10^{-5}, 10^{-1}]$ (rad/s), car en plus la flexibilité, cette dernière rend le bras manipulateur vulnérable au principe d'inertie comme nous l'avons expliqué précédemment. Il faut noter que ce phénomène se produit dans tous les robots flexibles.

La figure (VI .8) illustre aussi que les résultats obtenus par la méthode de R-K4 son plus précis par rapport à la méthode d'Euler.

La figure (Fig VI .11) représente la variation des couples appliqués aux deux segments.

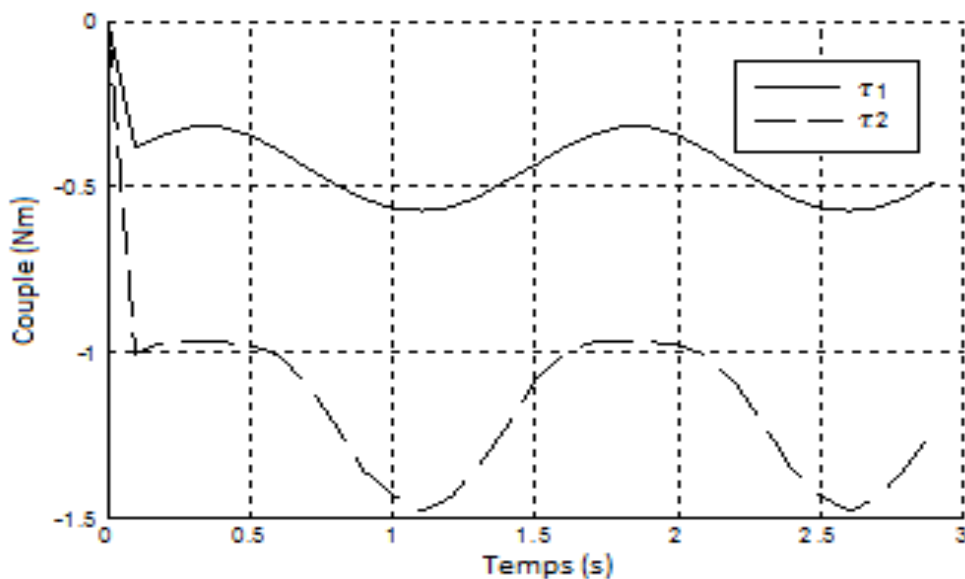


Fig VI .11 : Sorties des FNNs

Le couple moteur est la force du mouvement de rotation du moteur responsable de l'articulation angulaire du segment. Il est lié au régime moteur (la vitesse de rotation du moteur, exprimée en rad/s). Ainsi, plus le régime moteur sera élevé et plus le couple sera à son maximum et vice-versa. Il est exprimé en Newton Mètre (Nm) et se calcule en fonction de la force du mouvement de rotation et de la distance parcourue soit :

$$\text{Couple (Nm)} = \text{Force (N)} \times \text{Distance (m)}$$

VI.7.2 Test de la robustesse

Afin de tester la robustesse des deux contrôleurs conçus, un autre mouvement sera exécuté, avec des changements sur les paramètres internes du bras manipulateur, sans effectuer un nouvel apprentissage du mouvement et sans aucun changement au niveau des contrôleurs. La nouvelle trajectoire à suivre est la suivante :

$$\begin{cases} q_{Ref3}(t) = -\frac{\pi}{2} + \frac{1}{4} \cdot \left[2 \cdot \pi \cdot \frac{t}{3} - \sin\left(2 \cdot \pi \cdot \frac{t}{3}\right) \right] \\ q_{Ref4}(t) = \frac{1}{4} \cdot \left[2 \cdot \pi \cdot \frac{t}{3} - \sin\left(2 \cdot \pi \cdot \frac{t}{3}\right) \right] \end{cases} \quad t \leq 3 \quad (6.30)$$

Les nouvelles valeurs des paramètres du robot que nous avons choisi sont représentées dans le Tableau (IV.5).

Variable	valeur
L_1	0.40 m
L_2	0.30 m
L_{c1}	0.20 m
L_{c2}	0.15 m
m_1	2.0 Kg
m_2	1.5 Kg
I_1, I_2	0.028 Kg.m ²
g	9.8 N/Kg.

Tableau VI.5 : Paramètres du procédé

La figure (VI .12), (VI .13) et (VI .14) représente la variation angulaire du premier segment exécutant l'articulation représentée par la fonction (6.30), l'erreur en position, et la variation de la vitesse angulaire respectivement.

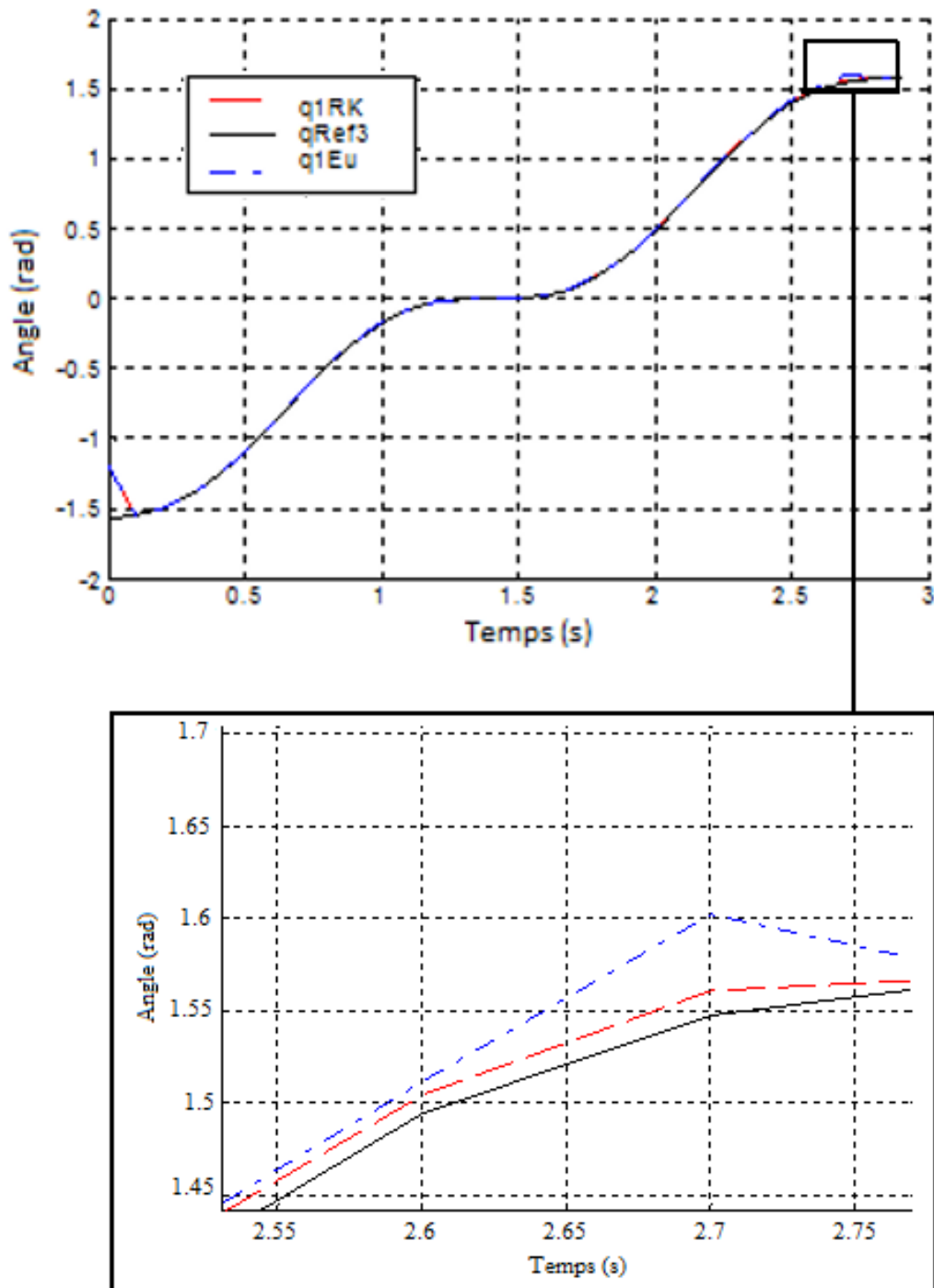


Fig. VI .12 : L'articulation du segment-1 pour les conditions initiales

$$[q_1, \dot{q}_1, \ddot{q}_1, \tau]^T = [-1.2, 0, 0, 0]$$

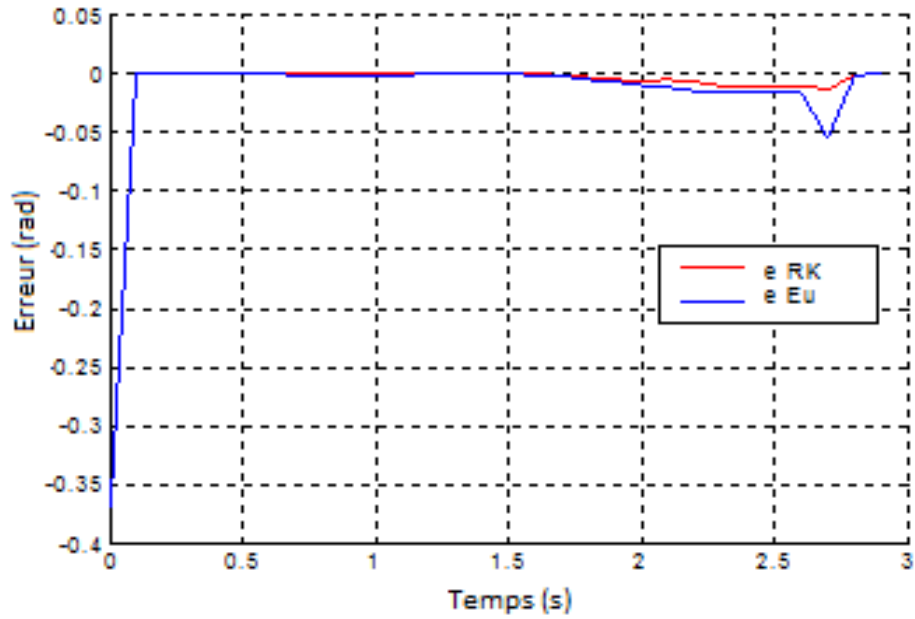


Fig. VI .13 : Erreur en position du segment-1

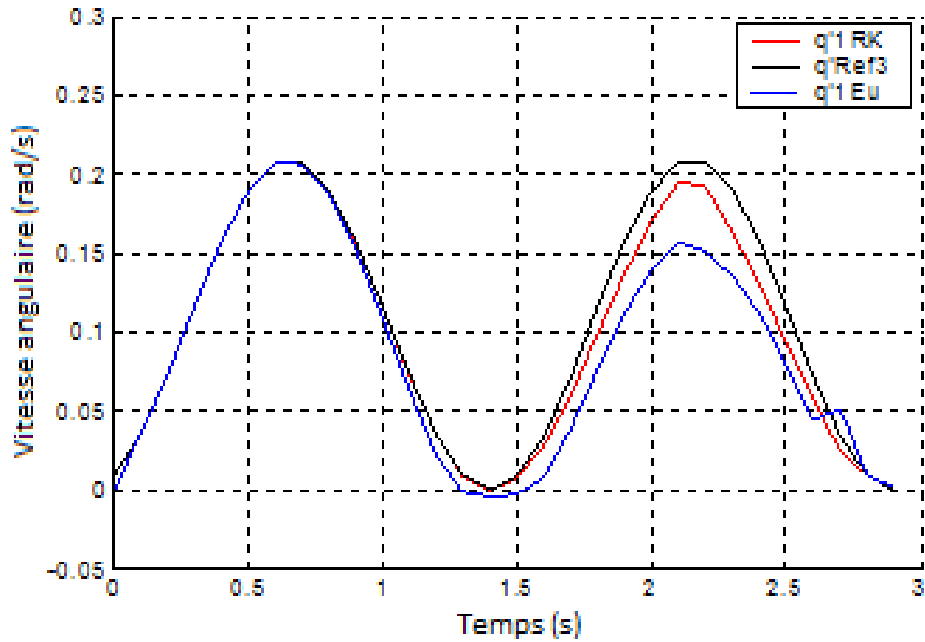


Fig VI .14 : Variation de la vitesse angulaire de segment-1

Les figures (VI.15), (VI.16), (VI.17) et (VI.18) représentent la variation angulaire du deuxième segment, l'erreur en position, et la variation de la vitesse angulaire, et les sorties des FNNs respectivement.

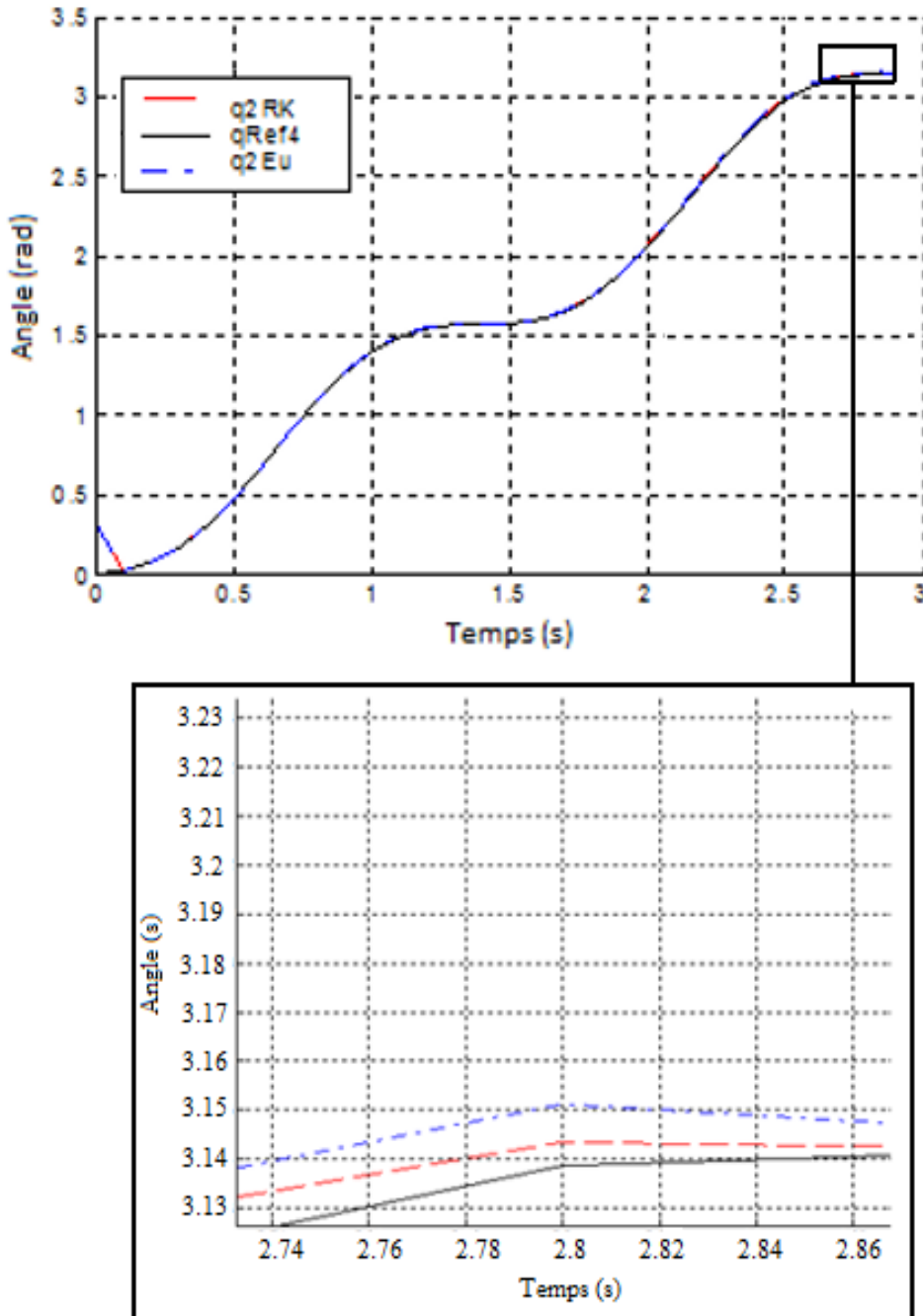


Fig VI.15 : L'articulation du segment-2 pour les conditions initiales $[q_2, \dot{q}_2, \ddot{q}_2, \tau]^T = [0.3, 0, 0, 0]$

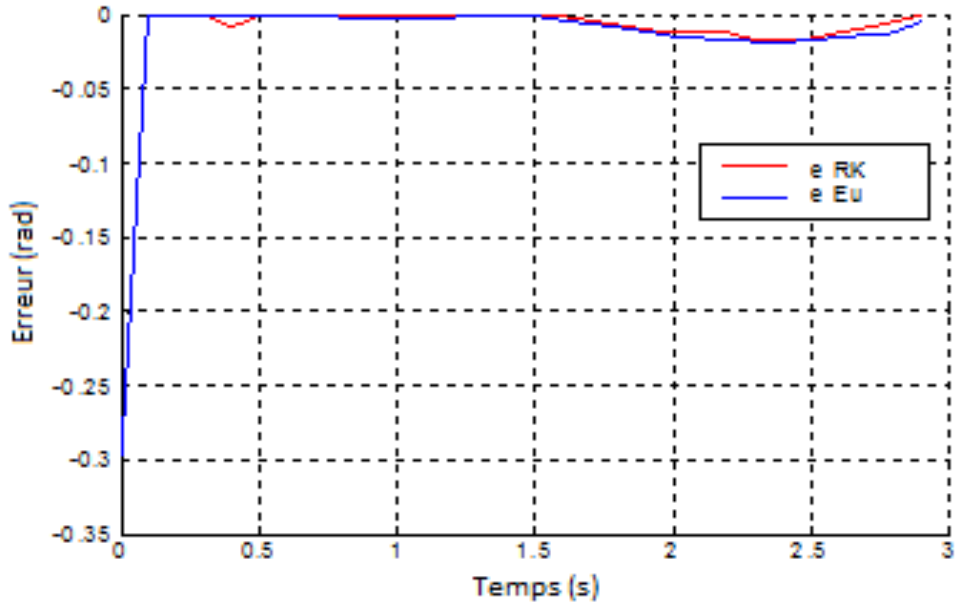


Fig VI .16 : Erreur en position du segment-2

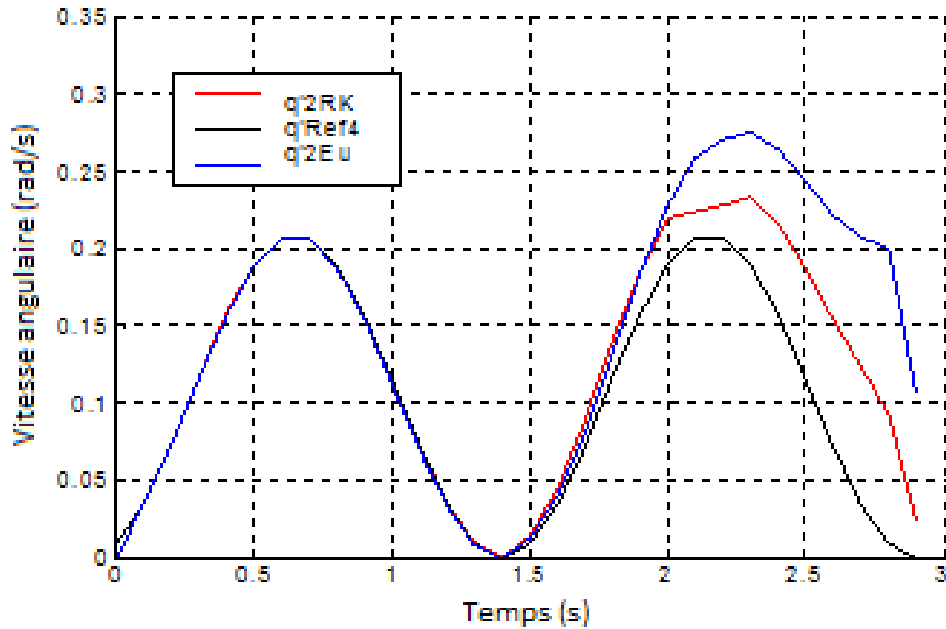


Fig VI .17 : Variation de la vitesse angulaire du segment-2

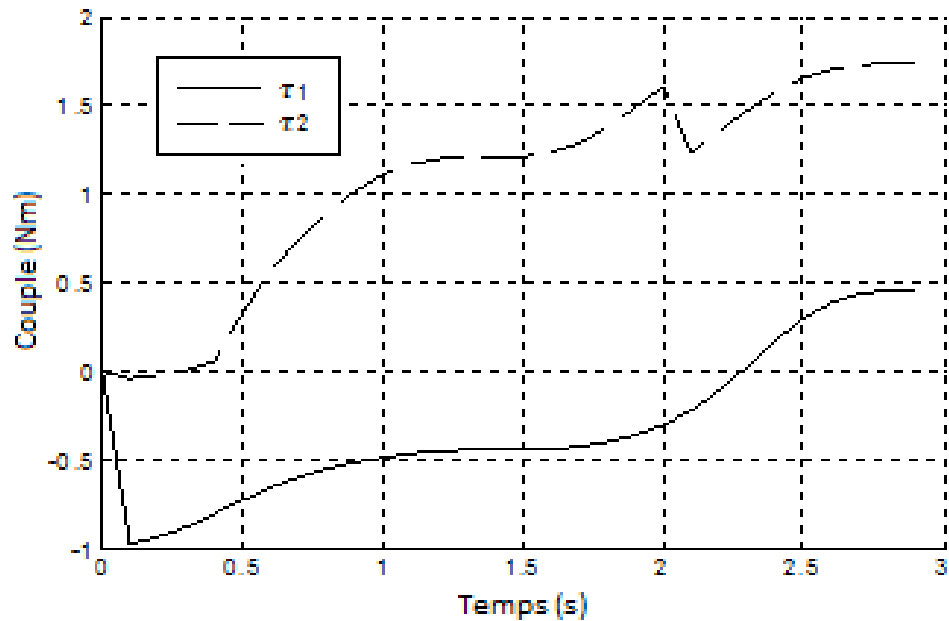


Fig. VI .18 : Sorties des FNNs

Les résultats de la simulation montrent les réponses des articulations en positions et en vitesses angulaires. Ainsi que la sortie des FNNs qui est représentée par le couple, moment de force. Ces résultats sont obtenus avec de nouvelles conditions initiales, et avec des modifications sur le poids et la longueur de chaque segment du bras manipulateur.

Lorsque les équations dynamiques sont dérivées par la méthode de R-K 4, il est clair que l'effet de contrôle du système est meilleur et l'erreur de suivi de l'articulation est plus petite.

Nous remarquons que l'erreur augmente près des extrémités des trajectoires, comme le montre plus précisément la figure (VI .17). Cette erreur se produit parce que le procédé n'a pas accompli un freinage saillant à la fin du suivi de mouvement. Ce phénomène a été expliqué précédemment, et pourrait être diminué avec davantage d'étude sur la gestion du mouvement. De manière analogue, l'humain démontre également un comportement similaire lors de l'exécution de mouvements rapides : un dépassement à la fin du mouvement [9] [22] [67]. Finalement, Comme nous pouvons le voir, les résultats de la simulation vérifient l'efficacité de la solution proposée.

VI.7. Conclusion

Le contrôle d'un robot manipulateur est une tâche difficile est ardue. Pour résoudre ce problème, nous avons développé un contrôleur neuro-flou en tirant parti des avantages du système flou et des réseaux de neurones. Les paramètres et les identifications de structure du réseau ont été effectués en utilisant un régulateur ANFIS. L'un des problèmes majeurs dans la mise en œuvre du contrôleur, est l'adaptation de ses poids synaptiques qui a été résolu par un apprentissage supervisé. La fonction dynamique du bras manipulateur est complexe et non linéaire. Pour résoudre ce problème, une étude comparative a été menée entre deux méthodes, la méthode d'Euler et la méthode de Runge Kutta d'ordre 4. D'après Les résultats de la simulation on peut déduire que l'approche proposée a un bon effet de suivi, et que l'erreur de poursuite des trajectoires est réduite en utilisant la méthode de RK-4 par rapport à la méthode d'Euler.

Finalement, l'objectif fixé par notre étude a été atteint, et les résultats de la simulation montrent l'efficacité et la robustesse de cette approche vis-à-vis des changements des paramètres internes du bras manipulateur, et au nouveau mouvement d'exécution.