

Modélisation et commande d’un bras manipulateur redondant avec sept dégrées de liberté « IIWA 14 »

Présenté par :

*Iheb BOUARICHE

*Maroua LAROUSSI

## Sommaire

[Introduction 1](#_TOC_250011)

[Modélisation du robot KUKA LBR IIWA 14 3](#_TOC_250010)

1. [Modèle Géométrique Direct (MGD) 3](#_TOC_250009)

Tableau de Denavit-Hartenberg 3

Les valeurs relatives à la géométrie du robot 3

La matrice de Denavit-Hartenberg 3

1. [Modèle Géométrique Inverse (MGI) 4](#_TOC_250008)

Algorithme de BFGS 5

1. [Modèle Dynamique Direct (MDD) 5](#_TOC_250007)

Les équations du modèle dynamique direct calculé par la méthode d’Euler Lagrange 5

1. [Modèle Dynamique Inverse (MDI) 6](#_TOC_250006)
2. Valisation du modèle dynamique direct (SimScape-CAO) par le modèle dynamique inverse 6

[Commande du robot IIWA 14 8](#_TOC_250005)

1. [Commande PID 8](#_TOC_250004)

Application de la commande sur IIWA 14 10

1. [Commande à Couple Calculé 12](#_TOC_250003)

Conception du contrôleur à couple calculé 12

Stabilité de la commande 14

Application de la commande sur IIWA 14 14

1. [Commande par mode glissant 17](#_TOC_250002)

Application de la commande sur IIWA 14 20

Le phénomène de réticence (Chattering) 22

[Génération de la trajectoire avec KUKA LBR IIWA 14 26](#_TOC_250001)

Les quatre points désirés 26

Les solutions des articulations 26

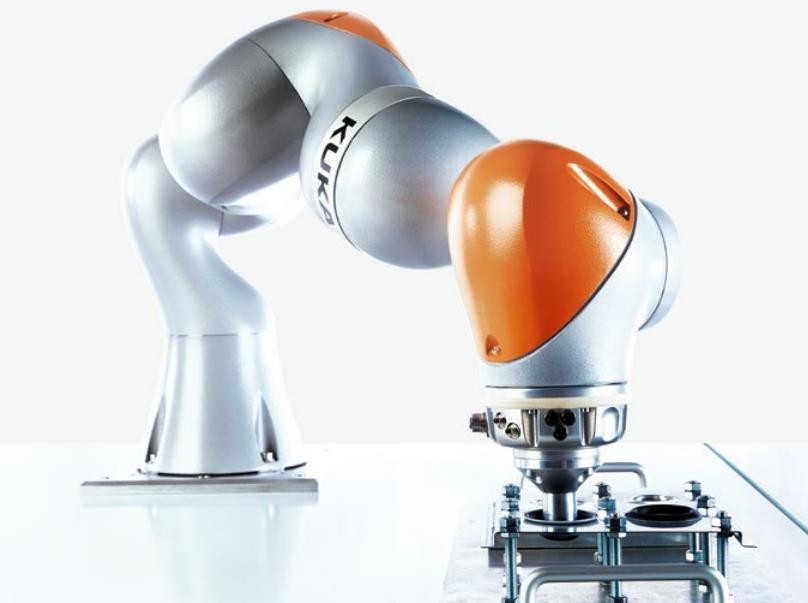
* 1. Trajectoire commandée par la commande PID 27
  2. Trajectoire commandée par la commande à Couple Calculé 30
  3. Trajectoire commandée par la commande par mode glissant 33

[Conclusion 36](#_TOC_250000)

## Introduction

La robotique est une des réalités fascinantes de l’histoire de l’humanité. De la conception des premiers automates jusqu’à la création des robots humanoïdes, les chercheurs ont toujours fait preuve de beaucoup de créativité. L’homme a toujours réfléchi à des manières d’améliorer ses conditions de vie. Ainsi, il a eu l’idée de construire des robots afin de s’affranchir de nombreuses tâches répétitives, dangereuses, ou nécessitant un certain degré de précision.

La robotique est une activité multidisciplinaire visant l'étude, la conception et la construction de robots ou plus simplement de machines automatiques. Sa pratique réunie des savoir-faire techniques et des connaissances scientifiques des domaines, de l’automatique, de l'électronique, de l'informatique et de la mécanique. Il est incontestable que la technologie robotique ne cesse d’évoluer jour après jour et qu’elle devient de plus en plus dominante dans de nombreux domaines tels que l’industrie (chaînes de montage), la médecine (diagnostic et le traitement de certaines maladies), l’aérospatial (l’exploration de l’espace), la défense militaire (les transports des armes et matériels), des laboratoires (les recherches scientifiques), la sécurité pour la surveillance, etc. Elle influence presque tous les aspects de la vie car elle a le potentiel de la transformer, d'améliorer les niveaux d'efficacité et de sécurité, de relever les niveaux de service et de créer des emplois.



**Figure 1-1 :** robot manipulateur « IIWA14 »

Un robot est un assemblage complexe de pièces mécaniques, électromécanique ou pièces électroniques. L'ensemble est piloté par un contrôleur numérique qui peut inclure : une simple séquence d'automatisme, [logiciel informatique](https://fr.wikipedia.org/wiki/Logiciel) ou une [intelligence artificielle](https://fr.wikipedia.org/wiki/Intelligence_artificielle) suivant le degré de complexité des tâches à accomplir. Il existe plusieurs types de robots : robots mobiles, bras manipulateur, robots humanoïde, cobots, robots sous-marins etc…

Parmi les types de robots les plus utilisés les bras manipulateurs qui est constitué d’une chaine de corps liées entre eux par des articulations peuvent être des actionneurs à mouvement rotatif ou prismatique. Ils sont également équipés de capteurs et d’un contrôleur. Ce type de robot est programmable et peut être autonome ou contrôlé manuellement. Il peut être aussi utilisé dans le but d’effectuer une variété de tâches avec une grande précision.

Notre projet portera sur le robot LBR IIWA. Il s’agit d’un un bras manipulateur possédant sept degrés de liberté créé par la boîte allemande KUKA en 2016. IIWA a une masse légère, peut atteindre un espace de travail important et effectuer des tâches compliquées avec précision grâce à sa géométrie ainsi que son poids. Ce robot est capable de lever jusqu’à 14Kg. IIWA est le premier robot ayant été développé afin de travailler en collaboration avec l’être humain avec sa capacité d’apprentissage et son intelligence. Remplissant ces critères ainsi que d’autres, il a pu dominer une partie très importante dans les applications de l’industrie 4.0, éducation, domaine médical, recherche scientifique etc…

**Figure 1-2 :** IIWA14 en collaboration avec un humain.

## Modélisation du robot KUKA LBR IIWA 14

Dans cette partie, une représentation mathématique de la position de l’effecteur par rapport aux positions des actionneurs du bras manipulateur IIWA sera donnée. Ensuite, on fera une étude mathématique du comportement dynamique du système.

### Modèle Géométrique Direct (MGD) :

Le modèle géométrique direct d'un robot est la relation permettant de calculer les coordonnées opérationnelles du robot, c.à.d. la position de l’organe terminal dans l’espace opérationnel en fonction des coordonnées articulaires.

Tableau de Denavit-Hartenberg :

On suppose que les positions des repères des articulations verticales A1, A3, A5 sont à la même position avec les articulation horizontales A2, A4 et A6 respectivement afin de pouvoir simplifier le calcul du modèle géométrique direct et vérifier les règles de D-H.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Links** | **a(mètre)** | **α(rad)** | **d(mètre)** | **θ(rad)** |
| **1** | 0 | α1 | d1+d2 | θ1 |
| **2** | 0 | α2 | 0 | θ2 |
| **3** | 0 | α3 | d3+d4 | θ3 |
| **4** | 0 | α4 | 0 | θ4 |
| **5** | 0 | α5 | d5+d6 | θ5 |
| **6** | 0 | α6 | 0 | θ6 |
| **7** | 0 | α7 | d7 | θ7 |

Les valeurs relatives à la géométrie du robot **:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **d1**=0.1575 | **d2**=0.2025 | **d3**=0.2045 | **d4**=0.2155 | **d5**=0.1845 | **d6**=0.2155 | **d7=**0.1260 |
| **α1** = -π/2 | **α2** = π/2 | **α3** = π/2 | **α4** = -π/2 | **α5** = -π/2 | **α6** = π/2 | **α7** = 0 |

La matrice de Denavit-Hartenberg :

𝑐𝜃𝑖 −𝑠𝜃𝑖 ∗ 𝑐𝛼𝑖

𝐴(𝑖) = [𝑠𝜃𝑖 𝑐𝜃𝑖 ∗ 𝑐𝛼𝑖

0 𝑠𝛼𝑖

0 0

𝑠𝜃𝑖 ∗ 𝑠𝛼𝑖 𝑎𝑖 ∗ 𝑐𝜃𝑖

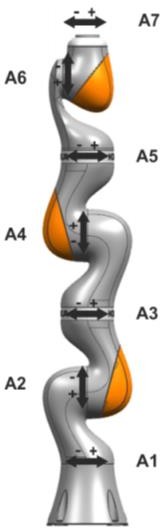
−𝑐𝜃𝑖 ∗ 𝑠𝛼𝑖 𝛼𝑖 ∗ 𝑠𝜃𝑖]

𝑐𝛼𝑖 𝑑𝑖

0 1

𝑇7 = 𝐴1 ∗ 𝐴2 ∗ 𝐴3 ∗ 𝐴4 ∗ 𝐴5 ∗ 𝐴6 ∗ 𝐴7

0

**Remarque :** En raison de la taille importante des équations symboliques représentant les éléments de la matrice homogène 𝑇7 pour un robot à 7 degrés de liberté, le calcul de celle-ci s’avère long et complexe. C’est pourquoi, il a été fait à l’aide de MATLAB qui utilise des calculateurs numériques.

0

**Figure 2-1 :** Robot KUKA LBR IIWA14

### Modèle Géométrique Inverse (MGI) :

A l’inverse du MGD, le modèle géométrique inverse représente la relation permettant d’obtenir les coordonnées articulaires du robot, c.à.d. les positions angulaires de chacune de ses articulations en fonction des coordonnées opérationnelles.

IWAA est un robot à sept degrés de liberté, ce qui le rend redondant car son degré de liberté est supérieur au degré de liberté de l’espace opérationnel 3D limité à six degrés de liberté.

Possédant un nombre de degré de liberté supérieur à celui de l’espace opérationnel limité à 6, IIWA est considéré comme un robot redondant. Par conséquent, il existe une infinité de solutions pouvant résoudre le MGI. De ce fait, on fera appel à une des méthodes d’optimisation numérique, « quasi- Newton ». Il s’agit de l’algorithme de BFGS (Broyden–Fletcher–Goldfarb–Shanno) qui prend des valeurs initiales pour les articulations et les fait converger vers la solution la plus proche.

Le problème à résoudre par les méthodes Quasi-Newtoniennes est de trouver la direction de

convergence d’une fonction d’erreur, entre la position désirée et la position courante, vers zéro afin de pouvoir mettre à jour les positions des articulations.

L’algorithme de la méthode de BFGS présenté ci-dessous permet d’approximer la matrice Hessienne et le gradient de la fonction d’erreur afin de pouvoir calculer la direction puis mettre à jours les articulations.

Algorithme de BFGS :

Pour N nombre d’itérations :

1. Initialisation des articulations θi et la matrice Hessienne initiale B(0)
2. Calcul de la valeur de la direction 𝑃(𝑘) = −𝐵(𝑘)−1 ∗ ∇𝑓(θ(𝑘)), tel que 𝑓(θ(𝑘)) est une fonction d’erreur entre la position désirée et la position courante.
3. Mise à jour des valeurs articulations 𝜃𝑖(𝑘 + 1) = 𝜃𝑖 + 𝛼 ∗ 𝑃(𝑘) , sachant que α est un facteur qui représente le taux de changement pour chaque pas.

4- Calcul de 𝑦(𝑘) = ∇𝑓(θ(𝑘 + 1)) − ∇𝑓(𝜃(𝑘))

5- Mise à jour de la matrice Hessienne : 𝐵(𝑘 + 1) = 𝐵(𝑘) + 𝑦(𝑘)∗𝑦(𝑘)𝑇

𝑇

𝑦(𝑘) ∗𝑠(𝑘)

− 𝐵(𝑘)∗𝑠(𝑘)∗𝑠(𝑘)𝑇∗𝐵(𝑘)𝑇

𝑠(𝑘)𝑇∗𝐵(𝑘)∗𝑠(𝑘)

Fin.

### Modèle Dynamique Direct (MDD) :

Le modèle dynamique direct représente la relation entre les positions et les vitesses articulaires ainsi que les couples à appliquer aux articulations du robot pour avoir les accélérations de ces dernières.

On a utilisé SolidWorks pour relever les paramètres du robot. Quant à leur validation, elle s’est faite à l’aide de Matlab en utilisant Simulink et Robotics Systems Toolbox.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Corps** | **Masse(kg)** | **Gx(m)** | **Gy(m)** | **Gz(m)** | **Ixx** | **Iyy** | **Izz** | **Iyz** | **Ixz** | **Ixy** |
| **0** | 5 | 0 | -0.03 | 0.12 | NaN | NaN | NaN | NaN | NaN | NaN |
| **1** | 4 | 0.0003 | 0.059 | 0.1995 | 0.0745 | 0.1345 | 0.08 | 0 | 0.035 | 0 |
| **2** | 4 | 0 | 0.13 | 0.39 | 0.1612 | 0.1476 | 0.0236 | 0.0144 | 0 | 0 |
| **3** | 3 | 0 | 0.067 | 0.5985 | 0.0710 | 0.0251 | 0.0579 | -0.001 | 0 | 0 |
| **4** | 2.7 | 0.0001 | -0.076 | 0.8010 | 0.1334 | 0.1257 | 0.0127 | -0.0117 | 0 | 0 |
| **5** | 1.7 | 0 | -0.0006 | 0.9649 | 0.0452 | 0.0131 | 0.0411 | -0.0062 | 0 | 0 |
| **6** | 1.8 | 0 | 0.02 | 1.18 | 0.0306 | 0.0278 | 0.0057 | -0.0027 | 0 | 0 |
| **7** | 0.3 | 0 | 0 | 1.281 | 0.005 | 0.0036 | 0.0047 | 0 | 0 | 0 |

Les équations du modèle dynamique direct calculé par la méthode d’Euler Lagrange :

On commence par le calcul de l’énergie cinétique consommé par les mouvements articulaires des corps du robot :

𝟕

𝑲 = 𝟏 ∑ 𝒎 𝒗𝑻𝒗

+ 𝑚𝑻𝑰 𝑚

𝟐

𝒊=𝟏

𝒊 𝒊 𝒊

𝒊 𝒊 𝒊

Ensuite, on calcule l’énergie potentielle due à la force de gravité sur les articulations du robot :

𝟕

𝑼 = ∑ 𝒎𝒊 𝒈 𝒉𝒊

𝒊=𝟏

On termine par le calcul des équations de Lagrange et les équations différentielles qui vont définir la dynamique couplée des articulations du robot en termes d’inertie, effet de Coriolis et centrifuge, de la gravité et les couples générés par les actionneurs :

𝑳 = 𝑲 − 𝑼

# 𝒅 𝑳

𝒅𝒕 ( 𝒒̇ ) −

𝒊

𝑳

𝒒𝒊

# = 𝟎

𝒒̈ = 𝑴−𝟏(𝒒) ∗ (𝑟 − 𝑪(𝒒, 𝒒̇ )𝒒̇ − 𝒈(𝒒))

### Modèle Dynamique Inverse (MDI) :

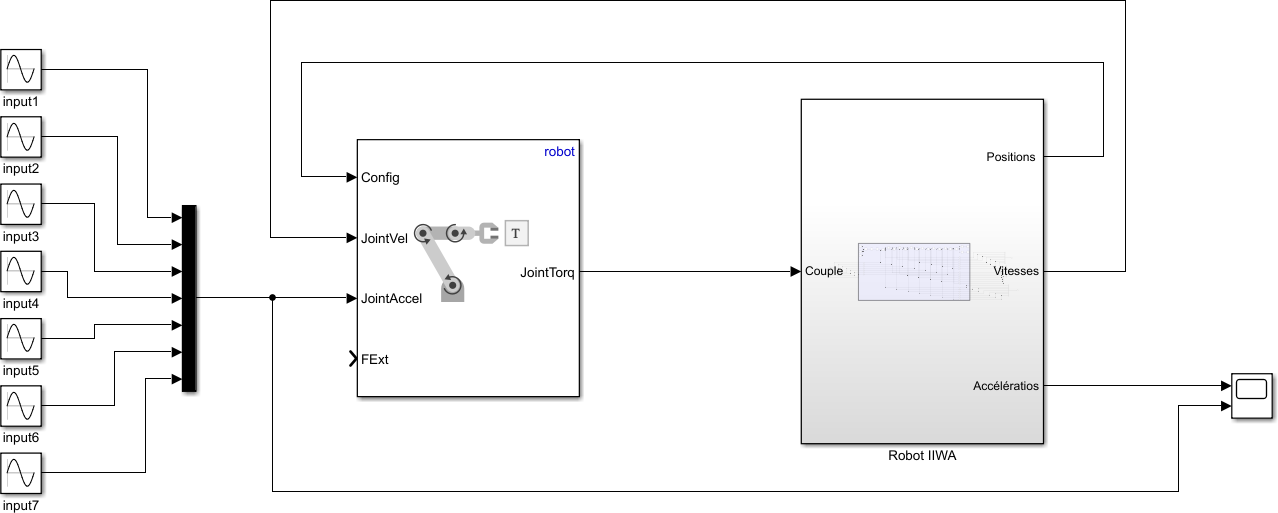
Le modèle dynamique inverse calcule les couples nécessaires à appliquer au niveau des articulations du robot en fonction des positions, vitesses et accélérations articulaires.

# 𝑴(𝒒)𝒒̈ + 𝑪(𝒒, 𝒒̇ )𝒒̇ + 𝒈(𝒒) = 𝑟

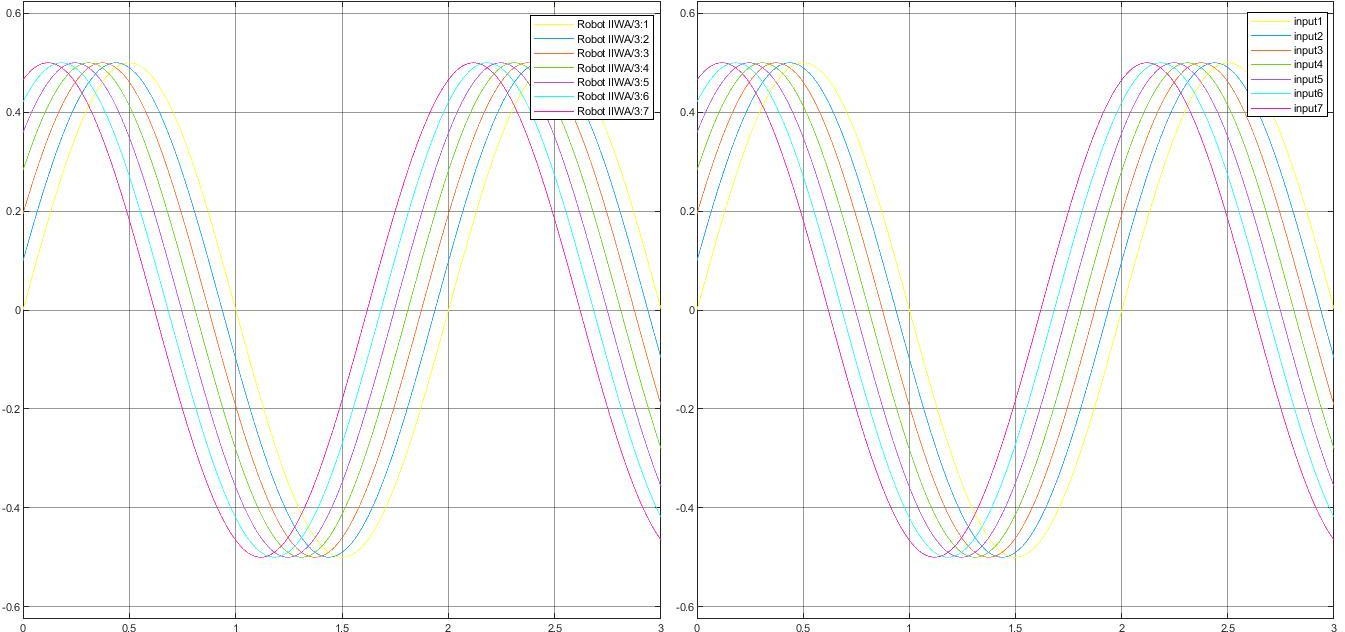
### Validation du modèle dynamique direct (SimScape – CAO) par le modèle dynamique inverse :

Pour une bonne simulation d’un système mécanique, on a utilisé SolidWorks pour la modélisation et l’assemblage du robot. On a également eu recours à Matlab, Simulink, Simscape et Robotics Systems Toolbox pour simplifier la résolution du modèle, effectuer des calculs numériques et réaliser une simulation et animation 3D.

Pour valider notre modèle dynamique direct (modèle en 3D définit en Simscape), on génère des entrés sinusoïdales déphasées les unes par rapport aux autres de 0.2 radian dans les entrées qui sont des accélérations du model dynamique inverse. Ensuite on vérifie la sortie du modèle dynamique direct. Les accélérations en sortie du modèle dynamique direct et les entrées générées doivent être confondues.



**Figure 2-2 :** Schéma de validation du robot IIWA14 par le MDI



**Figure 2-3 :** (à gauche) Signal de sortie du modèle (SimScape-CAO) | (à droite) les signaux d’entrée

On peut conclure que l'on a bien modéliser notre robot, et on peut aller vers la commande de notre système en se basant sur les paramètres calculés.

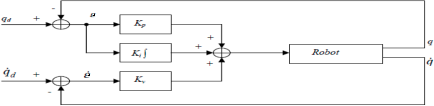
## Commande du robot IIWA 14

De nos jours, la commande des robots manipulateurs constitue un des axes dominants de la recherche en robotique. Cela est dû aux difficultés posées par les dynamiques hautement non linéaires et fortement couplées du robot et éventuellement des actionneurs (les effets inertiels, gravitationnels, centrifuges et de Coriolis, ainsi que les frottements) et la variation des paramètres au cours du temps (par exemple du fait du changement de charge). En effet, les problèmes rencontrés aussi bien en théorie qu’en pratique résident dans le fait que ces systèmes ne disposent pas d’outils et de techniques généraux permettant de synthétiser leurs commandes. La commande des robots manipulateurs a donc fait l’objet de nombreux travaux et est devenue un domaine de recherche très important et très vaste qui ne cesse d’évoluer. Au cours de ces dernières décennies, plusieurs approches de commandes ont été établies. Parmi ces dernières, on trouve celles qui ne sont pas basées sur le modèle, à savoir la commande de type PD et PID. D’autre commandes prennent en compte le modèle dynamique du robot manipulateur notamment la commande par couple calculé et la commande par mode glissant.

### Commande PID

Bien que de nombreux algorithmes de commande soient développés dans le but de satisfaire les objectifs exigés par les cahiers des charges, on constate une certaine réticence à adopter des techniques de contrôle non linéaires jugées difficiles à maîtriser, compliquée à mettre en œuvre et dont l’analyse des performances se révèle complexe. Pour cette raison, on a recours à des méthodes de commande linéaires et classique afin de traiter des problèmes non linéaires similaires à ceux des robots manipulateurs où ces derniers sont considérés comme des systèmes linéaires et chacune de leurs articulations est asservie par une commande décentralisée de type PID à gain constants. Il est donc possible de constater des dépassements de consigne dans la réponse temporelle du robot variant selon sa configuration et une mauvaise précision de suivi dans les mouvements rapides. Dans de nombreuses applications, ces inconvénients ne représentent pas un gros handicap.

En pratique, une telle commande est réalisée selon le schéma suivant :



**Figure 3-1 :** schéma synoptique de la commande PID.

La loi de commande est donnée par

Γ = 𝐾𝑝(𝑞𝑑 − 𝑞) + 𝐾𝑑(𝑞𝑑̇ − 𝑞̇) + 𝐾𝐼∫ (𝑞𝑑 − 𝑞)𝑑𝜏

Où 𝑞𝑑 et 𝑞𝑑̇ désignent les positions et les vitesses désirées dans l’espace articulaire et 𝐾𝑝, 𝐾𝑑 et 𝐾𝐼 sont des matrices diagonales définies positives de dimension (nxn).

Le calcul des gains est effectué en considérant le modèle de l’articulation j représenté par le système linéaire du deuxième ordre à coefficients constants suivant :

Γ𝑗 = 𝑚𝑗𝑞𝑗̈ + 𝐹𝑣𝑗𝑞𝑗̇ + 𝛾𝑗

Equation dans laquelle 𝑚𝑗 = 𝑀𝑗𝑗𝑚𝑎𝑥 désigne la valeur maximale de l’élément 𝑀𝑗𝑗 de la matrice d’inertie du robot et 𝛾𝑗 représente un couple perturbateur.

La fonction de transfert en boucle fermée pour 𝛾 = 0 est alors donnée par :

𝑞𝑗(𝑠)

𝑞𝑑(𝑠) =

𝑗

𝐾𝑑𝑗𝑠2 + 𝐾𝑝𝑗𝑠 + 𝐾𝐼𝑗

𝑚𝑗𝑠3 + (𝐾𝑑𝑗 + 𝐹𝑣 ) 𝑠2 + 𝐾𝑝𝑗𝑠 + 𝐾𝐼𝑗

𝑗

Et l’équation caractéristique s’écrit :

Δ(𝑠) = 𝑚𝑗𝑠3 + (𝐾𝑑𝑗 + 𝐹𝑣 ) 𝑠2 + 𝐾𝑝𝑗𝑠 + 𝐾𝐼𝑗

𝑗

La solution la plus courante en robotique consiste à choisir les gains de manière à obtenir un pôle triple réel négatif, ce qui donne une réponse rapide sans oscillations. Par conséquent, l’équation caractéristique se factorise de la façon suivante :

Δ(𝑠) = 𝑚𝑗(𝑠 + 𝜔𝑗)3

Avec

𝜔𝑗 > 0

On en déduit les gains :

Dans notre cas, 𝐹𝑣𝑗 = 0.

𝐾𝑝𝑗 = 3𝑎𝑗𝜔𝑗²

{𝐾𝑑𝑗 + 𝐹𝑣𝑗 = 3𝑎𝑗𝜔𝑗

𝐾𝐼𝑗 = 𝑎𝑗𝜔𝑗3

Remarques :

- 𝜔𝑗 est choisi le plus grand possible ; toutefois, cette pulsation ne doit pas être supérieure à la pulsation de résonance 𝜔𝑟𝑗 correspondant aux modes de vibration mécanique afin de ne pas

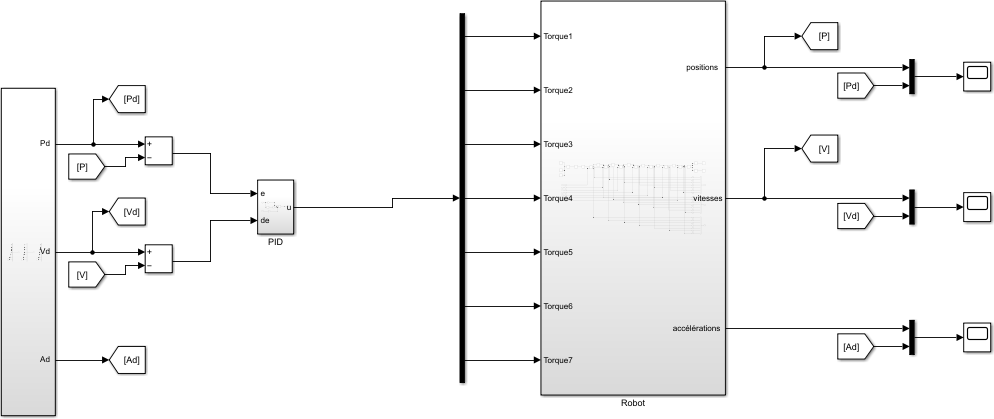
déstabiliser le système. Une valeur 𝜔𝑗

= 𝜔𝑟𝑗 représente un bon compromis ;

2

* En l’absence du terme intégral, une erreur statique due à la force de gravité et aux frottements peut subsister autour de la position finale.
* Le fait de choisir un pôle triple négatif assure la stabilité du système en boucle fermée. Elle peut également être vérifiée à l’aide du critère de Ruth (commande linéaire).

Application de la commande sur IIWA 14 :

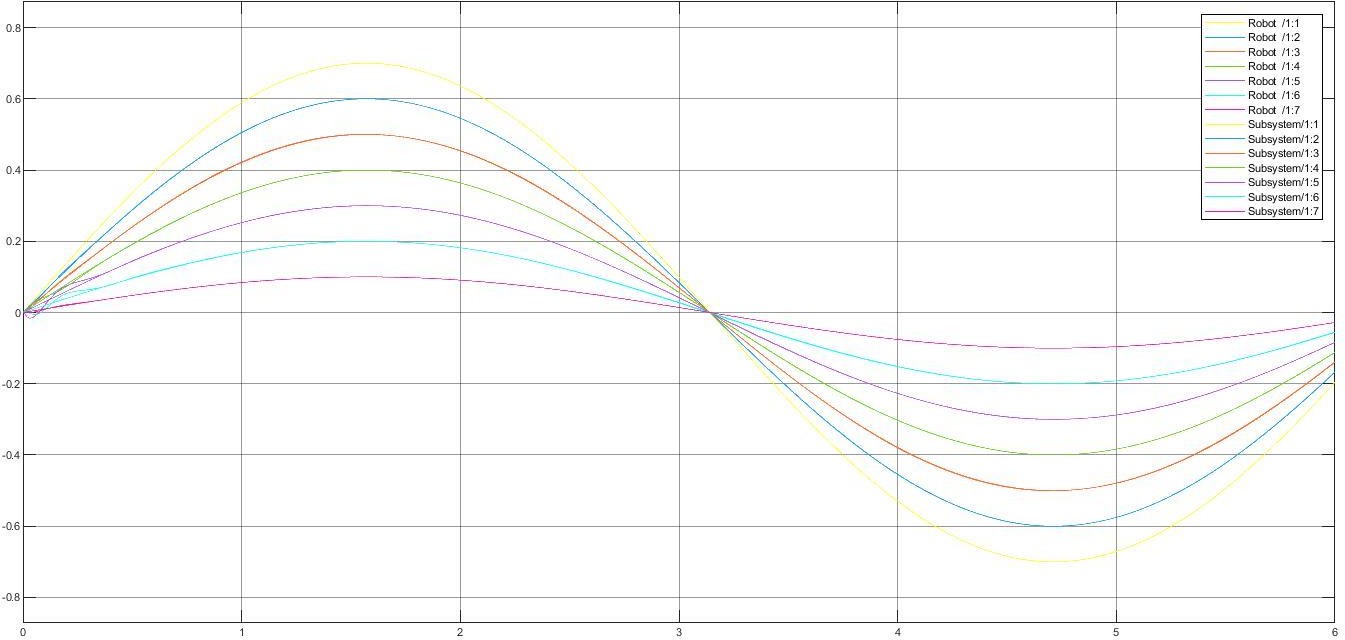


**Figure 3-2 :** Schéma de la commande PID (Simulink).

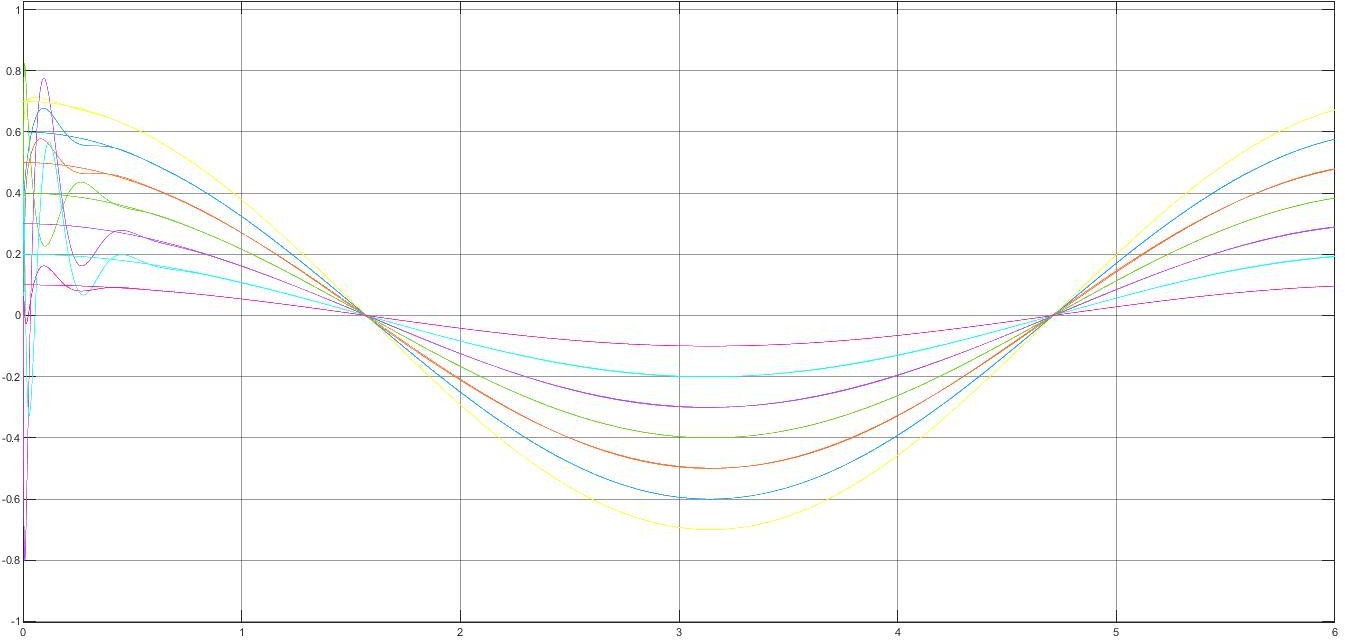
Les éléments 𝑚𝑗 = 𝑀𝑗𝑗𝑚𝑎𝑥 de la matrice d’inertie ont été déterminés graphiquement.

𝑚1 = 0.835, 𝑚2 = 3.4155, 𝑚3 = 0.112, 𝑚4 = 0.545, 𝑚5 = 0.010525, 𝑚6 = 0.0075, 𝑚1 = 0.001.

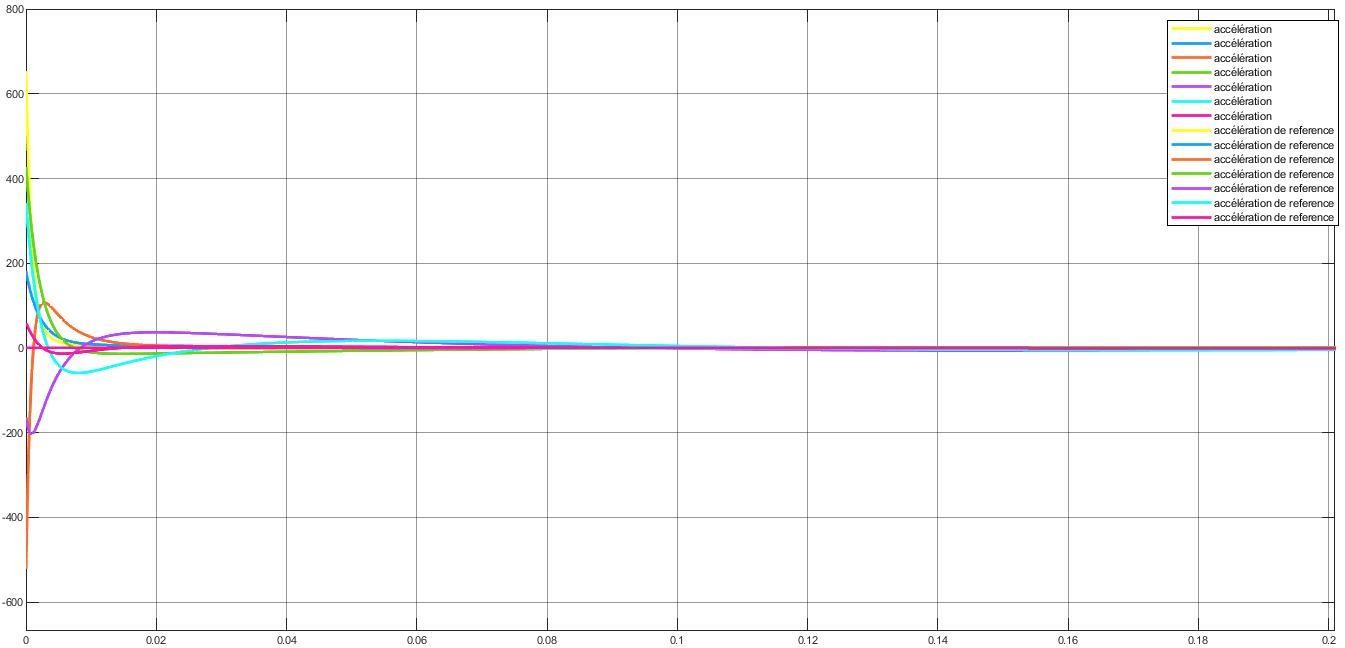
A chaque valeur 𝑚𝑗 (chaque articulation) correspond un 𝐾𝑝𝑗, 𝐾𝑑𝑗 et 𝐾𝐼𝑗. Pour un choix de 𝜔𝑗 =, on aboutit au résultat suivant :



**Figure 3-3 :** Signaux des positions articulaires mesurées et désirées après l’application de la commande PID.



**Figure 3-4 :** Signaux des vitesses articulaires mesurées et désirées après l’application de la commande PID.



**Figure 3-5 :** Signaux des accélérations articulaires mesurées et désirées après l’application de la commande PID.

### Interprétation :

On constate que le contrôleur PID se révèle assez performant en termes de poursuite. On peut aisément voir que les positions, vitesses et accélérations articulaires mesurée suivent leurs consignes dès la première seconde et L’erreur statique se trouve aussi annulée grâce à l’action intégrale tout en garantissant la stabilité du système grâce à l’action dérivée.

### Commande à Couple Calculé :

Lorsque l'application exige des évolutions rapides du robot et une grande précision dynamique, il est nécessaire de concevoir un système de commande plus sophistiqué qui prend en compte tout ou une partie des forces d'interaction dynamiques. L'utilisation de la commande à couple calculée constitue une bonne approche en ce sens. Il s’agit d’un contrôleur non linéaire puissant dont le principe consiste à linéariser un système en éliminant les non linéarités de ce dernier, puis lui appliquer la théorie du contrôle linéaire.

Conception du contrôleur à couple calculé

La commande à couple calculé repose sur l’annulation, partielle ou totale, des termes non linéaires par le choix du couple à appliquer au système pour que celui-ci soit linéarisé. Le modèle dynamique du manipulateur donné par :

𝑀(𝑞)𝑞̈ + 𝐶(𝑞, 𝑞̇)𝑞̇ + 𝐹(𝑞 ) + 𝐺(𝑞) = 𝑟

est donc exploité. Pour un choix de couple de la forme :

𝑟 = 𝑀(𝑞)𝑢 + 𝐶(𝑞, 𝑞̇)𝑞̇ + 𝐹(𝑞 ) + 𝐺(𝑞)

On aboutit au résultat suivant :

𝑀(𝑞)(𝑞̈ − 𝑢) = 0

Le problème se ramène donc à la commande de n systèmes linéaires, invariants, découplés et du second ordre (doubles intégrateurs). Plusieurs choix peuvent être envisagés pour 𝑢. Dans ce qui suit, on

considère le cas d’une commande spécifiée en termes d’accélération 𝑞̈𝑑 et obtenue par un compensateur de type Proportionnel Dérivée (PD)

Soit : 𝑢 = 𝑞̈𝑑 − 𝐾𝑑𝑒̇ − 𝐾𝑝𝑒

Où : 𝑒 = 𝑞 – 𝑞𝑑 et 𝑒̇ = 𝑞̇ – 𝑞̇𝑑 sont respectivement l’erreur de position et l’erreur de vitesse du robot manipulateur.

𝐾𝑑 et 𝐾𝑝 sont des matrices diagonales définies positives représentant les gains du contrôleur.

L’équation 𝑀(𝑞)(𝑞̈ − 𝑢) = 0 est équivalente à 𝑞̈ − 𝑢 = 0. En remplaçant 𝑢 par son expression, on trouve l’équation qui régit la dynamique de l’erreur en boucle fermée :

𝑒̈(𝑡) + 𝐾𝑑𝑒̇ + 𝐾𝑝𝑒 = 0

Les matrice de gain 𝐾𝑑 et 𝐾𝑝 sont choisies pour imposer à l’erreur de chaque axe j la dynamique désirée d’un système du second ordre, d'amortissement 𝜉𝑗 et de pulsation 𝜔𝑗 quelle que soit la configuration du robot. Les matrice 𝐾𝑑 et 𝐾𝑝 s’écrivent alors de la manière suivante :

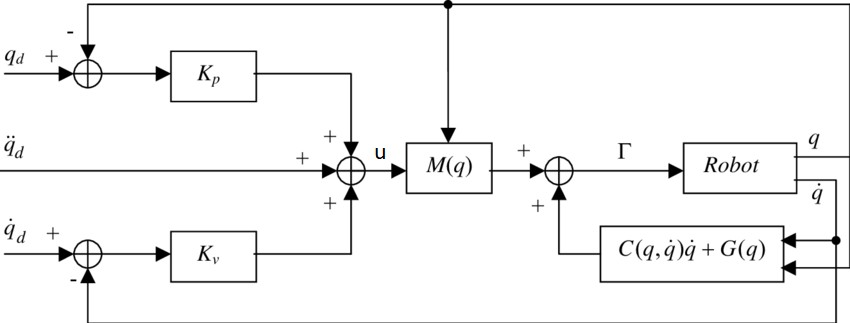
𝐾𝑑 = 𝑑𝑖𝑎𝑔{𝐾𝑑𝑗} et 𝐾𝑝= 𝑑𝑖𝑎𝑔{𝐾𝑝𝑗}

Tel que : 𝐾𝑑𝑗 = 2 𝜉𝑗 𝜔𝑗 et 𝐾𝑝𝑗 = 𝜔𝑗²

Afin d’éviter que les dépassement et oscillations n’entachent la réponse du robot, il est souhaitable d’opter pour un amortissement 𝜉𝑗 = 1.

Finalement, le couple que les actionneurs doivent fournir est :

𝑟 = 𝑀(𝑞)𝑞̈𝑑 − 𝑀(𝑞)(𝐾𝑑𝑒̇ + 𝐾𝑝𝑒) + 𝐶(𝑞, 𝑞̇)𝑞̇ + 𝐹(𝑞 ) + 𝐺(𝑞)



-

**Figure 3-6 :** Schéma synoptique de la commande à couple calculé de type PD.

Stabilité de la commande :

L’équation 𝑒̈(𝑡) + 𝐾𝑑𝑒̇ + 𝐾𝑝𝑒 = 0 peut être prendre la forme d’une représentation d’état en posant :

𝑥1 = 𝑒  𝑥̇1 = 𝑒̇ = 𝑥2

{𝑥2 = 𝑒̇  𝑥̇2 = 𝑒̈ = −𝐾𝑑𝑒̇ − 𝐾𝑝𝑒 = −𝐾𝑑 𝑥2 − 𝐾𝑝𝑥1

{ 𝑥̇1 = 𝑥2

𝑥̇2 = −𝐾𝑑 𝑥2 − 𝐾𝑝𝑥1

Ce système peut s’écrire :

Avec : 𝑋̇ = [ 𝑥̇1 𝑥̇2]𝑇 = [𝑒̇ 𝑒̈]𝑇

𝑋 = [𝑥1 𝑥2]𝑇 =[𝑒 𝑒̇]𝑇

0 𝐼

𝐴 = [−𝐾𝑝 −𝐾𝑑 ]

𝑋̇ = 𝐴𝑋

Pour étudier la stabilité du point d’équilibre, on choisit la fonction candidate de Lyapunov :

𝑉(𝑥) = 𝑋𝑇𝑃𝑋

Tel que 𝑃 = 𝑃𝑇 > 0 ( 𝑠𝑦𝑚é𝑡𝑟𝑖𝑞𝑢𝑒 𝑑é𝑓𝑖𝑛𝑖𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒) est la solution de l’équation de Lyapunov :

Avec 𝑄 > 0

𝐴𝑇𝑃 + 𝑃𝐴 = −𝑄

𝑉̇ (𝑥) = 𝑋̇ 𝑇𝑃𝑋 + 𝑋𝑇𝑃𝑋̇

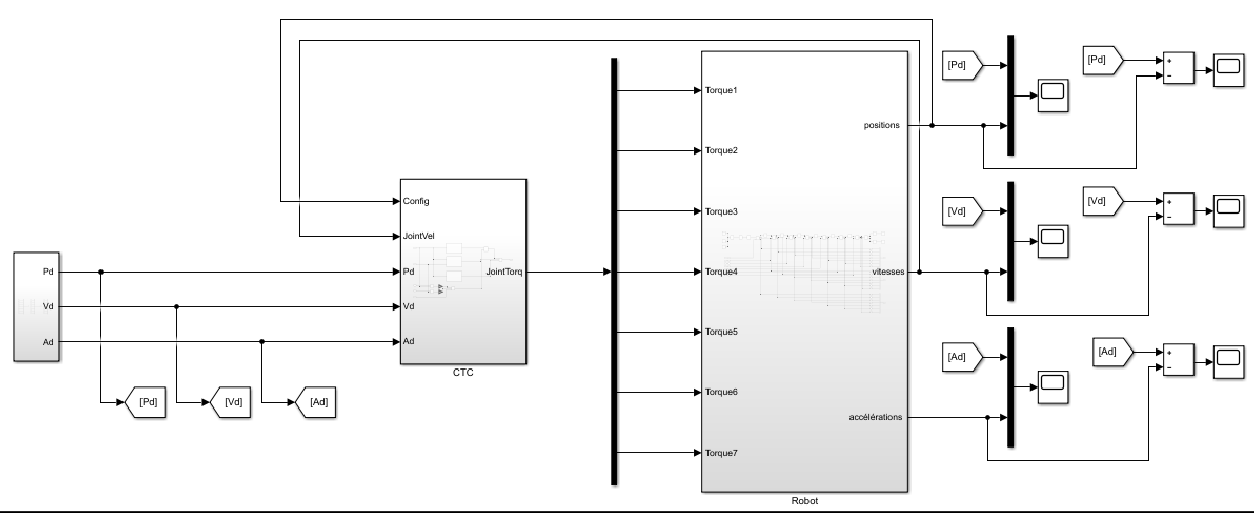
𝑉̇ (𝑥) = 𝑋𝑇𝐴𝑇𝑃𝑋 + 𝑋𝑇𝑃𝐴𝑋

𝑉̇ (𝑥) = 𝑋𝑇(𝐴𝑇𝑃 + 𝑃𝐴)𝑋

𝑉̇ (𝑥) = 𝑋𝑇(−𝑄)𝑋

𝑉̇ (𝑥) = −𝑋𝑇𝑄𝑋 < 0

On en conclut que la stabilité uniforme asymptotique globale du point d’équilibre est ainsi garantie. Application de la commande sur IIWA 14



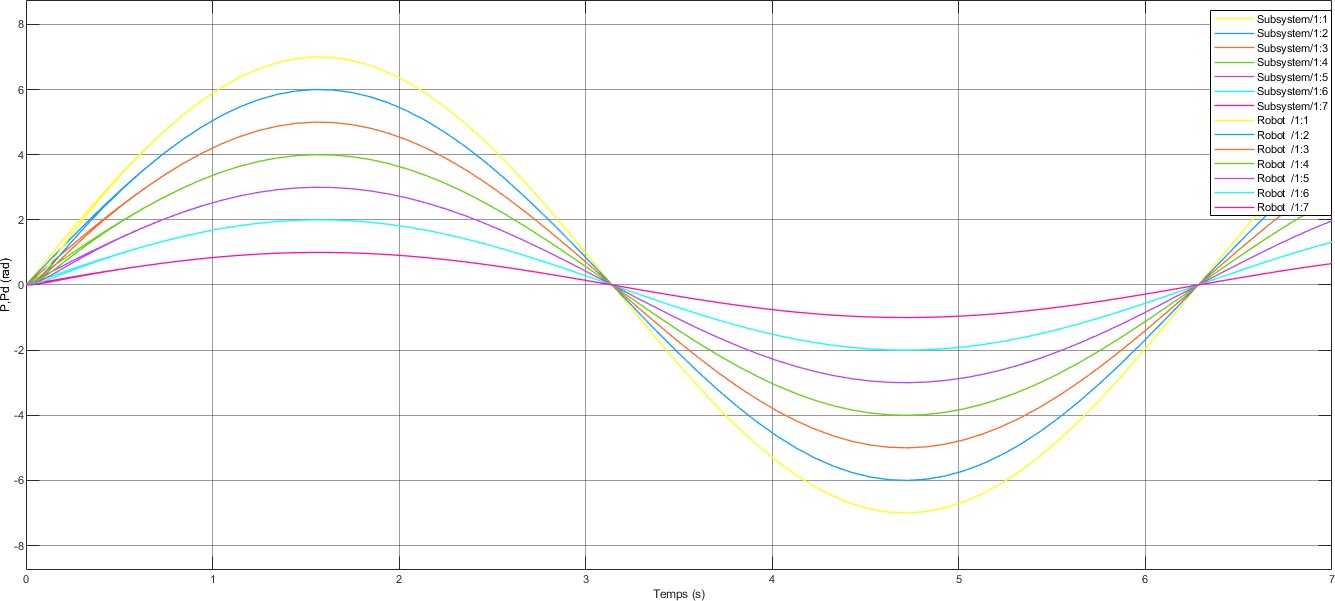
**Figure 3-7 :** Schéma de la commande à couple calculé de type PD (Simulink).

On désire asservir ce système afin que l’erreur suive la dynamique définie par :

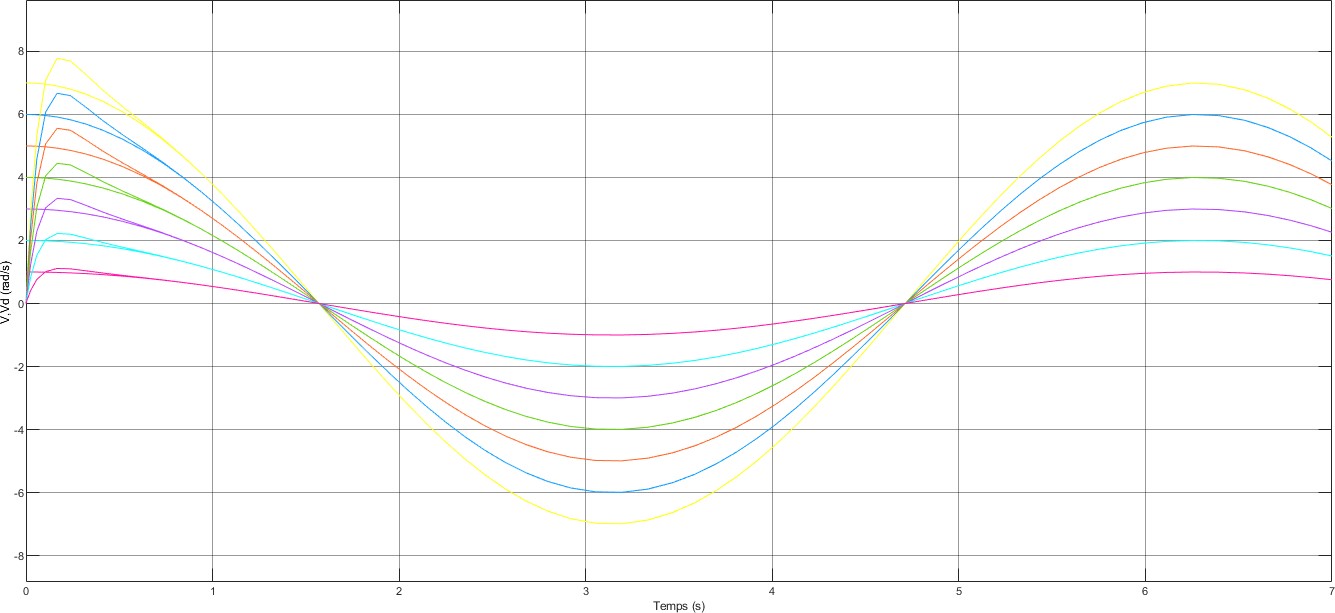
Δ = 𝑒̈(𝑡) + 20 𝑒̇ + 100𝑒

Avec 𝜉𝑗 = 1 et 𝜔𝑗 = 10𝑟𝑎𝑑/𝑠, ce qui implique que 𝐾𝑑𝑗 = 20 et 𝐾𝑝𝑗 = 102 = 100

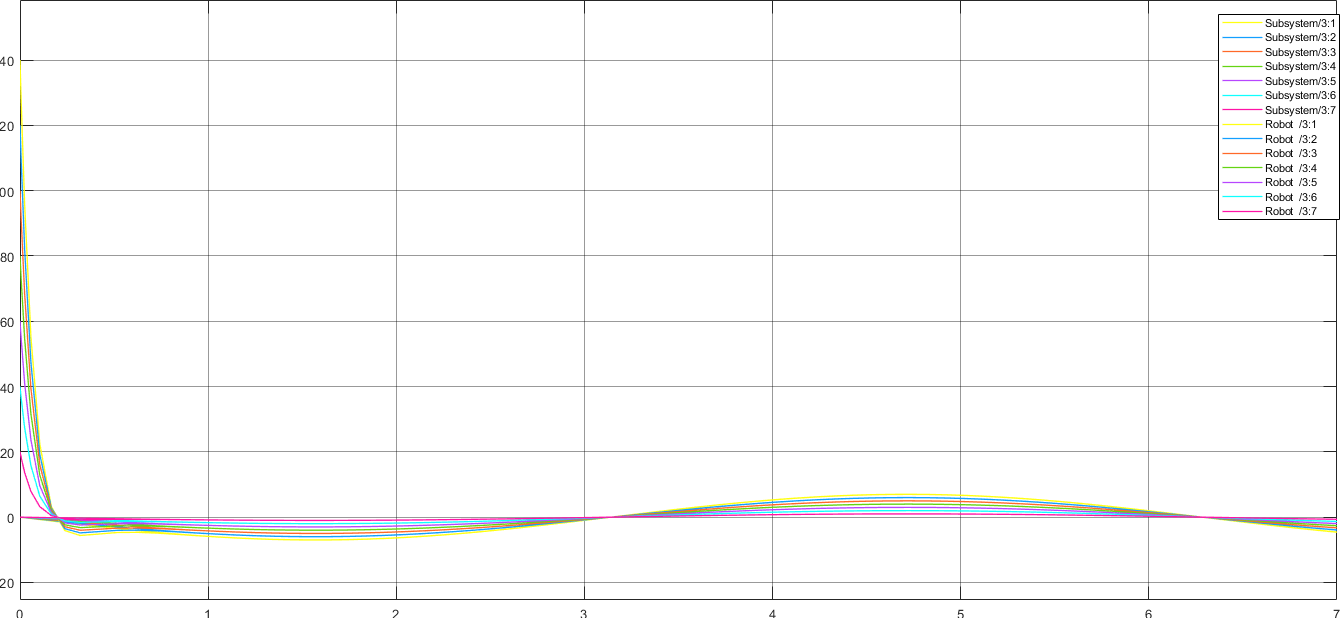
Pour un choix de 𝐾𝑑 = 𝑑𝑖𝑎𝑔{20} et 𝐾𝑝= 𝑑𝑖𝑎𝑔{100}, on obtient donc les résultats suivants :



**Figure 3-8 :** Signaux des positions articulaires mesurées et désirées après l’application de la commande CTC.



**Figure 3-9 :** Signaux des vitesses articulaires mesurées et désirées après l’application de la commande CTC.



**Figure 3-10 :** Signaux des accélérations articulaires mesurées et désirées après l’application de la commande CTC.

### Interprétation :

D’après les résultats de simulation durant 7s, on remarque que dès la première seconde les positions, les vitesses et les accélérations articulaire suivent leurs références respectives avec une erreur tendant vers 0. Ainsi, la commande par calcul de couple s’avère très performante dans ce cas (elle donne de bons résultats dans le cas où les paramètres du bras sont exactement connus).

### Commande par mode glissant :

Le contrôle des systèmes dynamiques, notamment les robots, présentant des non-linéarités et incertitudes de modélisation et subissant des perturbations, tels que les paramètres dynamiques (l'inertie et conditions de charge utile), les effets dynamiques (frottements non linéaires complexes) ainsi que la dynamique non modélisée est devenu un des problèmes les plus préoccupants à traiter lors de

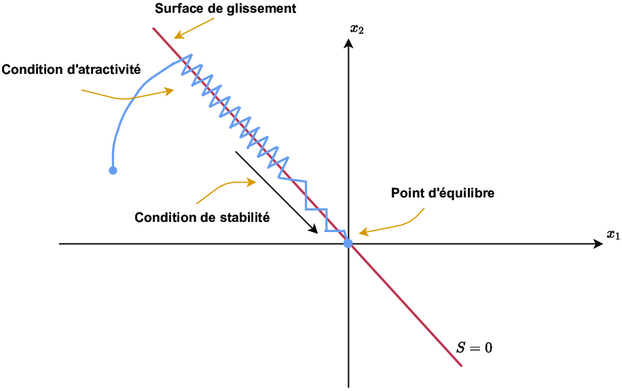
l’étude des installations industrielles. Par conséquent, des progrès considérables ont été atteint en ayant recours à des techniques de contrôle robustes, notamment la commande par mode glissant (Sliding Mode Control ou SMC).

Le mode de glissement est un mode de fonctionnement particulier des systèmes à structure variable. Le principe de cette technique de contrôle consiste à amener, quelles que soient les conditions initiales, la trajectoire de la variable d’état du système sur une hyper surface, dite surface de glissement et qui représente le comportement dynamique désiré, dans le plan de phase et de la maintenir sur cette surface jusqu’à ce qu’elle tende vers l’origine de ce plan. La poursuite de la trajectoire désirée se fait en deux phases : l’approche et le glissement. Ainsi, la commande utilisée dans ce cas se compose de deux parties : 𝑢(𝑡) = 𝑈𝑒𝑞(𝑡) + 𝑈𝑛(𝑡)

où

- U𝑒𝑞(𝑡) un terme continu appelé contrôle équivalent.

- U𝑛 (𝑡) un terme discontinu appelé commande de commutation.



**Figure 3-11 :** Surface de glissement.

Le choix typique de la surface de glissement est :

𝑆(𝑥) = 𝐾𝑒 + 𝑒̇

Avec

𝑒 = 𝑞 − 𝑞𝑑

𝑒̇ = 𝑞̇ − 𝑞𝑑̇

K : coeffcient positif.

L’objectif de la commande est de maintenir la surface 𝑆(𝑥) à zéro. Dans ce cas, cette dernière est une équation différentielle linéaire

𝑆(𝑥) = 𝐾𝑒 + 𝑒̇ = 0 𝐾 > 0

Dont l’unique solution est 𝑒 = 𝑒̇ = 0

Le modèle dynamique du robot manipulateur est :

𝑟 = 𝑀(𝑞)𝑞̈ + 𝐶(𝑞, 𝑞̇ )𝑞̇ + 𝐺(𝑞)

On en tire :

En posant {𝑥1 = 𝑞  𝑥1̇ = 𝑞̇



𝑥2 = 𝑞̇ 𝑥2̇ = 𝑞

𝑞̈ = 𝑀(𝑞)−1(𝑟 − 𝐶(𝑞, 𝑞̇)𝑞̇ − 𝐺(𝑞))

Le système peut se mettre sous la forme d’une représentation d’état :

{ 𝑥1̇ = 𝑥2

𝑥2̇ = 𝑀(𝑞)−1𝑟 − 𝑀(𝑞)−1(𝐶(𝑞, 𝑞̇)𝑥2 + 𝐺(𝑞))

Equivalente à la forme

Où 𝑢= 𝑟

𝑔(𝑥) = 𝑀(𝑞)−1

{ 𝑥1̇ = 𝑥2

𝑥2̇ = 𝑓(𝑥) + 𝑔(𝑥)𝑢

Et on a

𝑓(𝑥) = −𝑀(𝑞)−1(𝐶(𝑞, 𝑞̇)𝑥2 + 𝐺(𝑞))

𝑒 = 𝑞 − 𝑞𝑑 = 𝑥1 − 𝑥𝑑

1

𝑒̇ = 𝑞̇ − 𝑞̇𝑑 = 𝑥2 − 𝑥𝑑

2

On a donc :

𝑆(𝑥) = 𝐾𝑒 + 𝑒̇ = 𝐾(𝑥1 − 𝑥𝑑) + 𝑥2 − 𝑥𝑑

1 2

On en déduit :

𝑆̇(𝑥) = 𝐾𝑒̇ + 𝑒̈ = 𝐾(𝑥1̇ − 𝑥̇𝑑) + 𝑥̇2 − 𝑥̇𝑑

1 2

𝑆̇(𝑥) = 𝐾(𝑥2 − 𝑥𝑑) + 𝑓(𝑥) + 𝑔(𝑥)𝑢 − 𝑞̈𝑑

2

𝑆̇(𝑥) = 𝐾𝑒̇ + 𝑓(𝑥) + 𝑔(𝑥)𝑢 − 𝑞̈𝑑

Afin d’évaluer la stabilité de cette équation, on utilise la stailité de Lyapunov qui consiste à choisir une fonction condidate de Lyapunov. On a donc :

𝑉(𝑠) =

1 𝑆2

2

𝑉 𝑑é𝑓𝑖𝑛𝑖𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒

On a une stabilité asymptotique si : {𝑉̇ 𝑑é𝑓𝑖𝑛𝑖𝑒 𝑛é𝑔𝑎𝑡𝑖𝑣𝑒

Etant donnée que 𝑉(0) = 0 et 𝑉(𝑆) > 0 ∀𝑆 ≠ 0 et donc la première condition est vérifiée

𝑉̇ (𝑆) = 𝑑𝑉(𝑆) = 𝑑𝑉(𝑆) 𝑑𝑆 = 𝑆𝑆̇

𝑑𝑡

𝑑𝑆

𝑑𝑡

𝑉̇ est définie négative si :

𝑉̇ (𝑆) = (𝐾𝑒̇ + 𝑒̈) = 𝑆(𝐾𝑒̇ + 𝑓(𝑥) + 𝑔(𝑥)𝑢 − 𝑞̈𝑑)

< 0 𝑝𝑜𝑢𝑟 𝑆 > 0

𝑆̇(𝑥) = 𝐾𝑒̇ + 𝑓(𝑥) + 𝑔(𝑥)𝑢 − 𝑞̈𝑑 = {= 0 𝑝𝑜𝑢𝑟 𝑆 = 0

> 0 𝑝𝑜𝑢𝑟 𝑆 < 0

𝑆̇(𝑥) = 0

𝐾𝑒̇ + 𝑓(𝑥) + 𝑔(𝑥)𝑢 − 𝑞̈𝑑 = 0

𝑢 = 𝑔−1(𝑥) (𝑞̈𝑑 − 𝐾𝑒̇ − 𝑓(𝑥)) = 𝛽(𝑥)

La loi de commande équivalente est définie par 𝑈𝑒𝑞 = 𝛽(𝑥)

𝑢 < 𝛽(𝑥) 𝑝𝑜𝑢𝑟 𝑆 > 0

𝑉̇ est définie négative si :{𝑢 = 𝛽(𝑥) 𝑝𝑜𝑢𝑟 𝑆 = 0

𝑢 > 𝛽(𝑥) 𝑝𝑜𝑢𝑟 𝑆 < 0

Ceci est valable pour 𝑢 = 𝛽(𝑥) − 𝑔−1(𝑥)𝜆 𝑠𝑖𝑔𝑛(𝑆)

avec 𝜆 > 0

1 𝑝𝑜𝑢𝑟 𝑆 > 0

𝑠𝑖𝑔𝑛(𝑆) = { 0 𝑝𝑜𝑢𝑟 𝑆 = 0

−1 𝑝𝑜𝑢𝑟 𝑆 < 0

Et 𝑔(𝑥) = 𝑀(𝑞)−1 où 𝑀(𝑞) est la matrice d’inertie et est symétrique définie positive (SDP), donc son inverse l’est également.

Pour ce choix, on a :

𝑉̇ (𝑆) = 𝑆𝑆̇

𝑉̇ (𝑆) = 𝑆(𝐾𝑒̇ + 𝑓(𝑥) + 𝑔(𝑥)𝑢 − 𝑞̈𝑑)

𝑉̇ (𝑆) = 𝑆(𝐾𝑒̇ + 𝑓(𝑥) − 𝑞̈𝑑) + 𝑆𝑔(𝑥)𝑢

𝑉̇ (𝑆) = 𝑆(𝐾𝑒̇ + 𝑓(𝑥) − 𝑞̈𝑑) + 𝑆𝑔(𝑥)( 𝛽(𝑥) − 𝑔−1(𝑥)𝜆 𝑠𝑖𝑔𝑛(𝑆))

𝑉̇ (𝑆) = 𝑆(𝐾𝑒̇ + 𝑓(𝑥) − 𝑞̈𝑑) + 𝑆𝑔(𝑥) ( 𝑔−1(𝑥) (𝑞̈𝑑 − 𝐾𝑒̇ − 𝑓(𝑥)) − 𝑔−1(𝑥)𝜆 𝑠𝑖𝑔𝑛(𝑆))

𝑉̇ (𝑆) = 𝑆(𝐾𝑒̇ + 𝑓(𝑥) − 𝑞̈𝑑) + 𝑆 (𝑞̈𝑑 − 𝐾𝑒̇ − 𝑓(𝑥)) − 𝑆𝑔(𝑥)𝑔−1(𝑥)𝜆 𝑠𝑖𝑔𝑛(𝑆)

𝑉̇ (𝑆) = 𝑆(𝐾𝑒̇ + 𝑓(𝑥) − 𝑞̈𝑑) − 𝑆 (−𝑞̈𝑑 + 𝐾𝑒̇ + 𝑓(𝑥)) − 𝑆𝜆 𝑠𝑖𝑔𝑛(𝑆))

𝑉̇ (𝑆) = −𝑆𝜆 𝑠𝑖𝑔𝑛(𝑆))

𝑉̇ (𝑆) = −𝜆|𝑆| < 0 (Stabilité Vérifiée)

La commande globale proposée comporte donc deux termes :

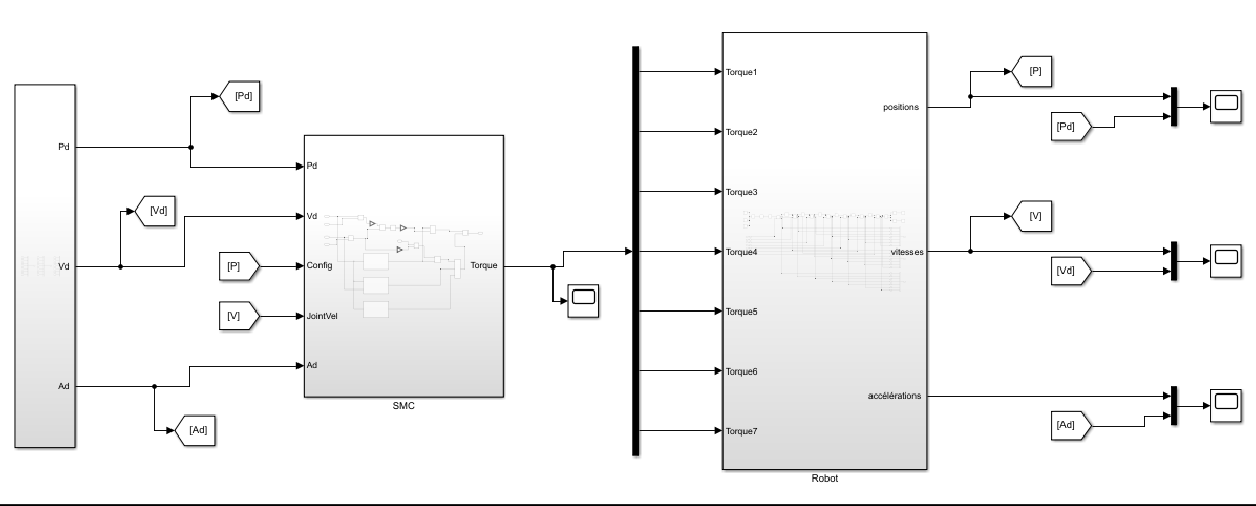
Finalement

𝑢 = 𝑈𝑒𝑞 + 𝑈𝑛 = 𝛽(𝑥) − 𝑔−1(𝑥)𝜆 𝑠𝑖𝑔𝑛(𝑆)

𝑢 = 𝑔−1(𝑥) (𝑞̈𝑑 − 𝐾𝑒̇ − 𝑓(𝑥)) −𝑔−1(𝑥)𝜆 𝑠𝑖𝑔𝑛(𝑆)

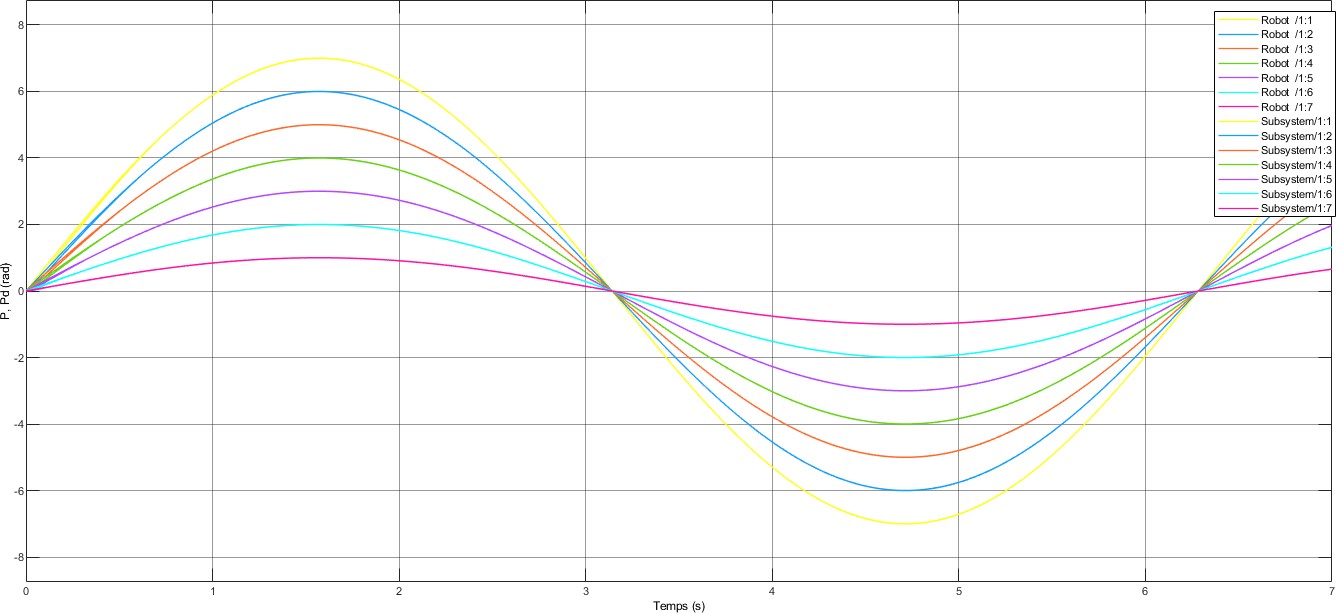
𝑢 = 𝑀(𝑞) (𝑞̈𝑑 − 𝐾𝑒̇ − 𝑓(𝑥)) − 𝑀(𝑞)𝜆 𝑠𝑖𝑔𝑛(𝑆)

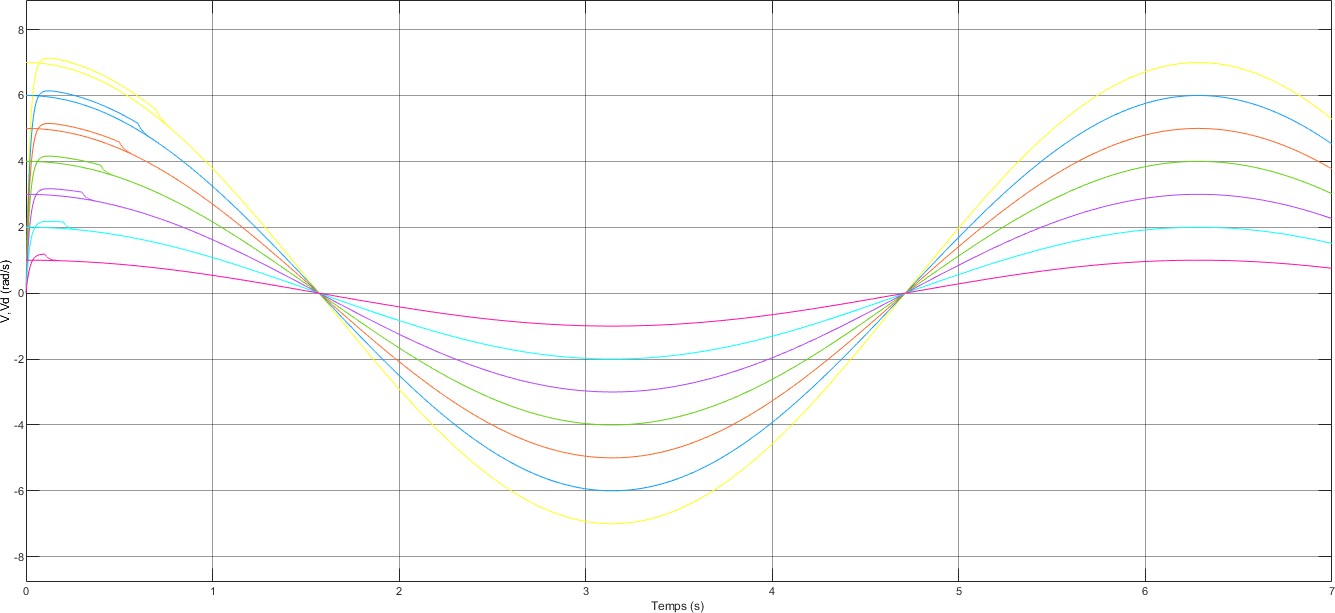
Application de la commande sur IIWA 14 :



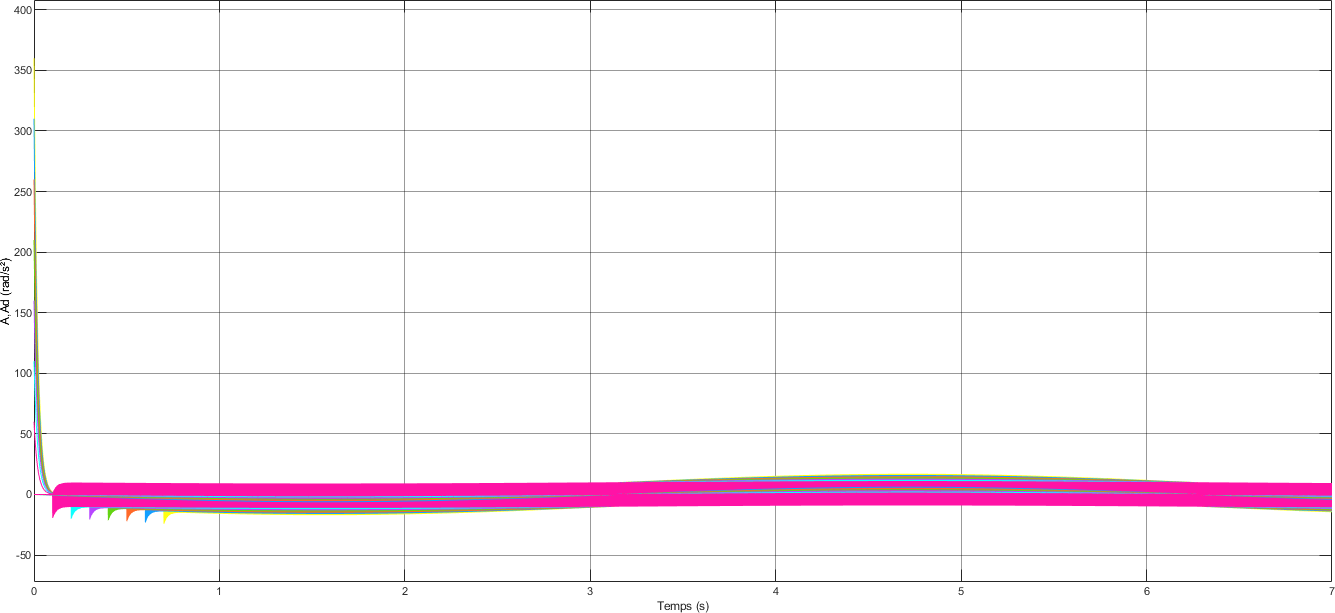
**Figure 3-12 :** Schéma synoptique de de la commande par mode glissant (SMC)

Pour un choix de 𝐾 = 𝑑𝑖𝑎𝑔{50} et 𝜆 = 𝑑𝑖𝑎𝑔{10} , on obtient les résultats suivants :



**Figure 3-13 :** Signaux des positions articulaires mesurées et désirées après l’application de la commande SMC.

**Figure 3-14 :** Signaux des vitesses articulaires mesurées et désirées après l’application de la commande SMC.



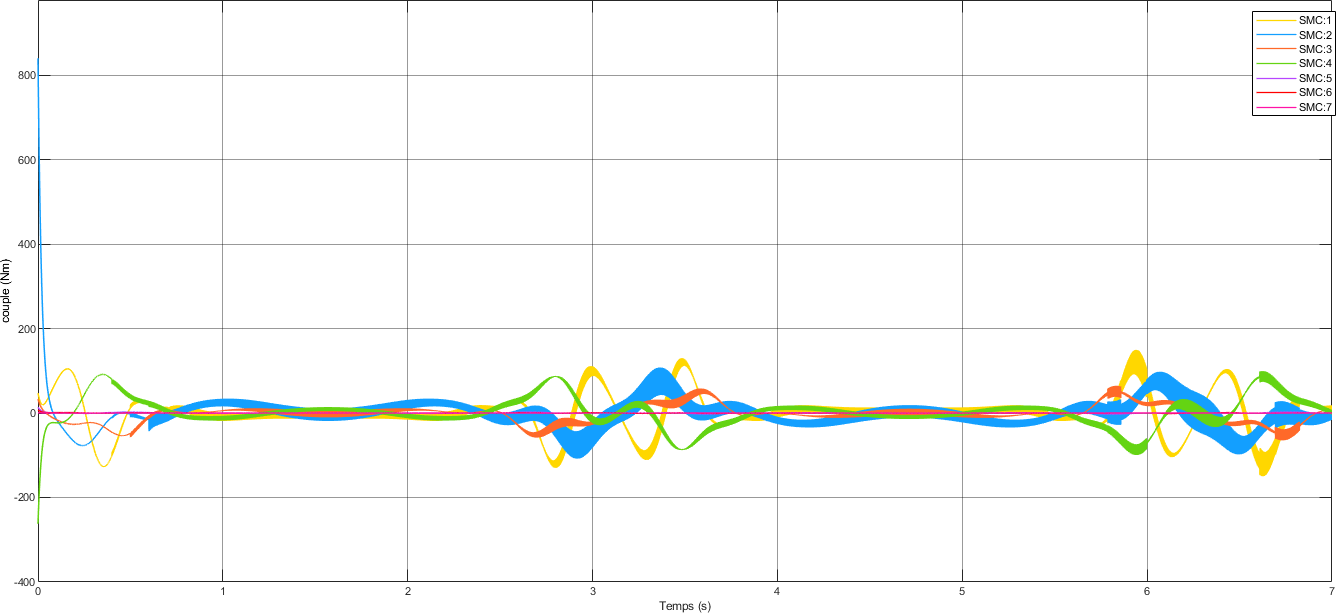
**Figure 3-15 :** Signaux des accélérations articulaires mesurées et désirées après l’application de la commande SMC.

### Interprétation :

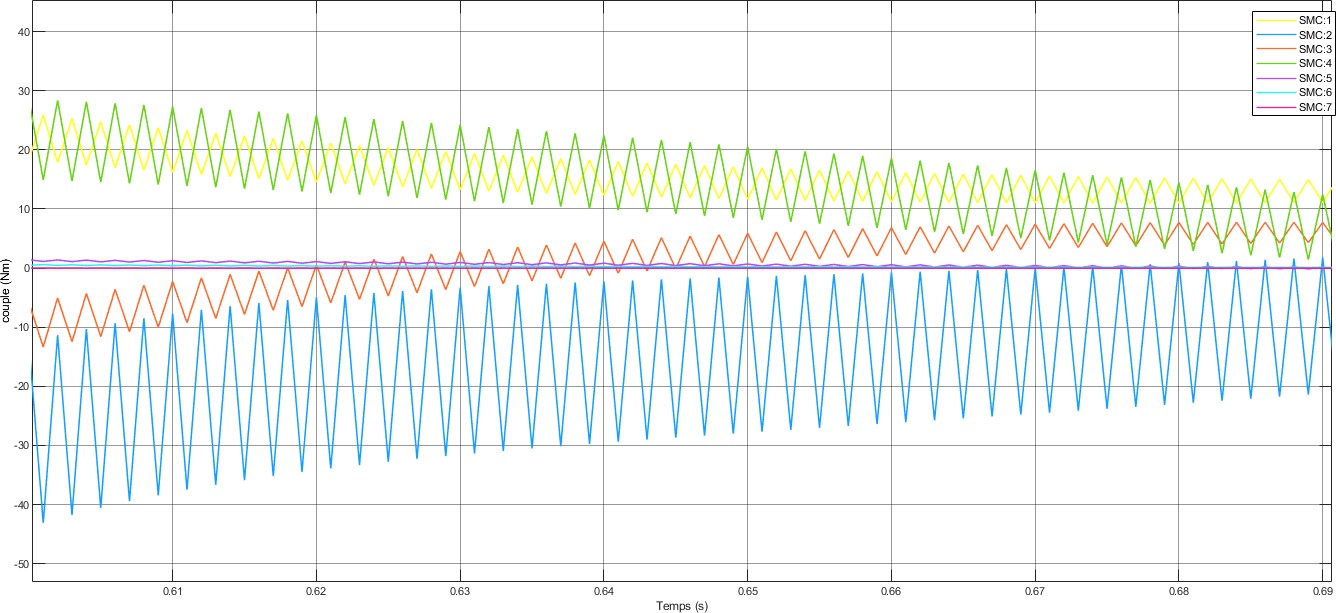
D’après les figures 3-14, 3-15 et 3-16, on constate que pour le choix de 𝐾 et 𝜆, les positions et vitesses articulaires convergent rapidement vers leurs références respectives avant que la première seconde ne s’écoule mais avec l’apparition d’oscillations à haute fréquence d’amplitude 10=𝜆 observées clairement au niveau des signaux d’accélérations ainsi que les signaux de commande qui sont dues à la fonction signe. Les erreurs tendent vers zéro.

Le phénomène de réticence (Chattering) :

La commande par mode glissant utilise essentiellement des lois de contrôle discontinues pour piloter la trajectoire d'état du système sur une surface de glissement ou de commutation et la confiner au voisinage de celle-ci. Cependant, elle présente des oscillations à haute fréquence appelées « chattering » lorsque l'état du système atteint la surface de glissement. Ce qui provoque des effets négatifs sur les actionneurs contrôlés et excite la dynamique non modélisée indésirable.



**Figure 3-16 :** Signaux de commande SMC.



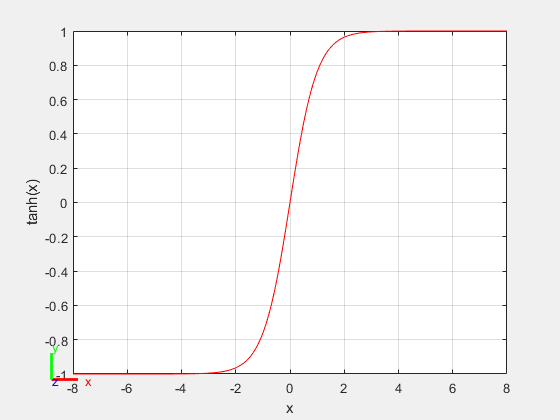
**Figure 3-17 :** phénomène de réticence (chattering).

Afin d’y remédier, plusieurs solutions peuvent être envisagées. Il est possible de :

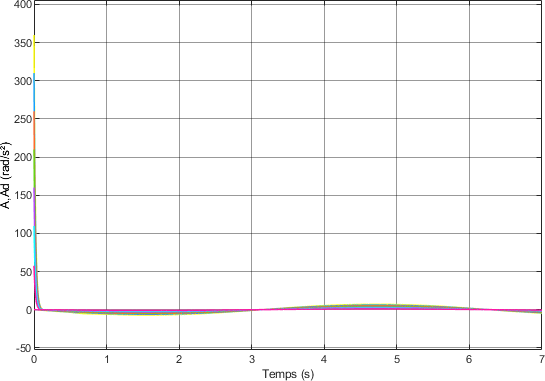
* Remplacer la fonction discontinue *Sign* par une une fonction continue et plus lisse telle que la fonction *Sat, Sigmoïde* ou *Tanh.*
* Algorithme du Super Twisting ou Mode Glissant d’ordre supérieur.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

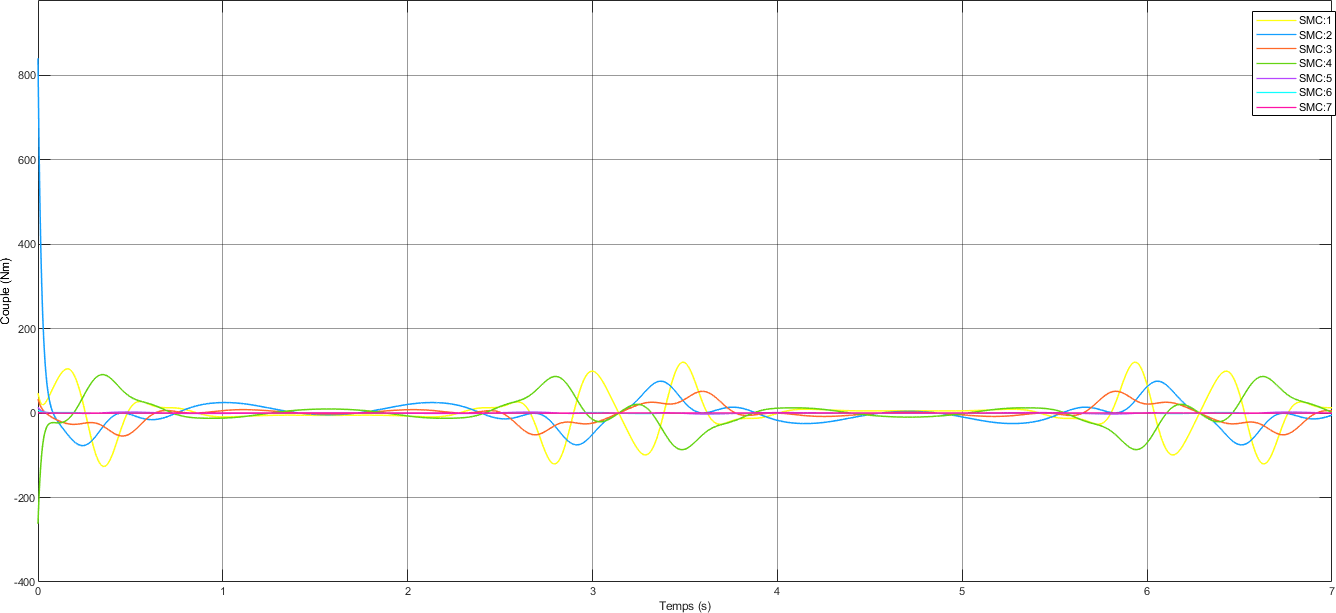
- **Figure 3-18 et 3-19 :** fonctions tanh(x) et sig(x).



-



**Figure 3-20 :** Signaux des accélérations articulaires mesurées et désirées après l’application de la commande SMC ( en remplaçant la fonction signe par tanh).



**Figure 3-21 :** Signaux de commande SMC ( en remplaçant la fonction signe par tanh).

## Génération de la trajectoire avec KUKA LBR IIWA 14

Après avoir résolu le modèle géométrique inverse en quatre points dans l’espace opérationnel vers

l’espace articulaires, une trajectoire a été générée entre ces points en partant de l’état initial de chaque articulation à l’aide d’un polynôme de degré cinq. La résolution de ce polynôme est conditionnée par des vitesses et des accélérations nuls dans chaque point désiré.

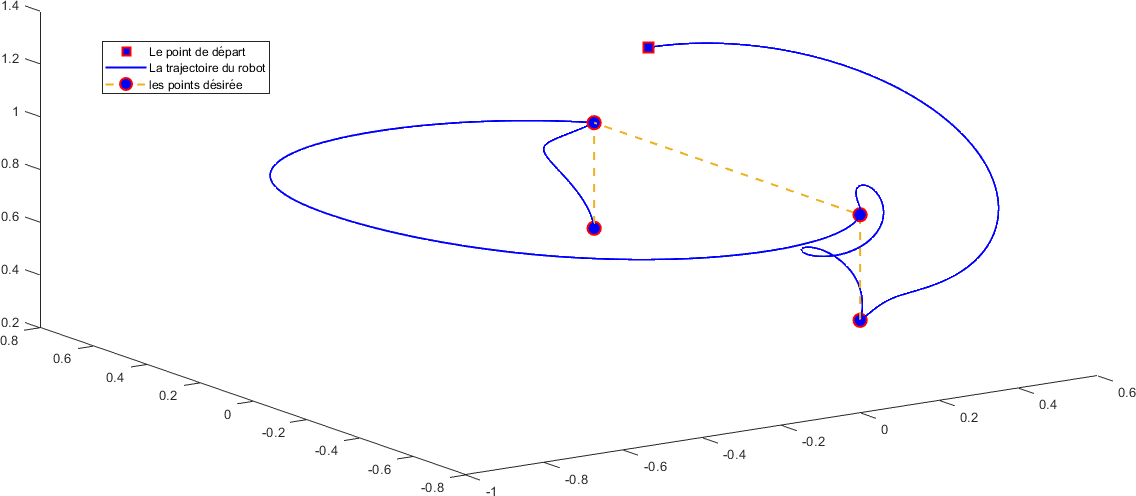
La trajectoire générée constituera l’entrée de référence du système à commander. Les quatre points désirés :

|  |  |  |  |
| --- | --- | --- | --- |
| **Points** | **X** | **Y** | **Z** |
| **1** | 0.2 | -0.5 | 0.4 |
| **2** | 0.2 | -0.5 | 0.8 |
| **3** | 0.2 | 0.5 | 0.8 |
| **4** | 0.2 | 0.5 | 0.4 |

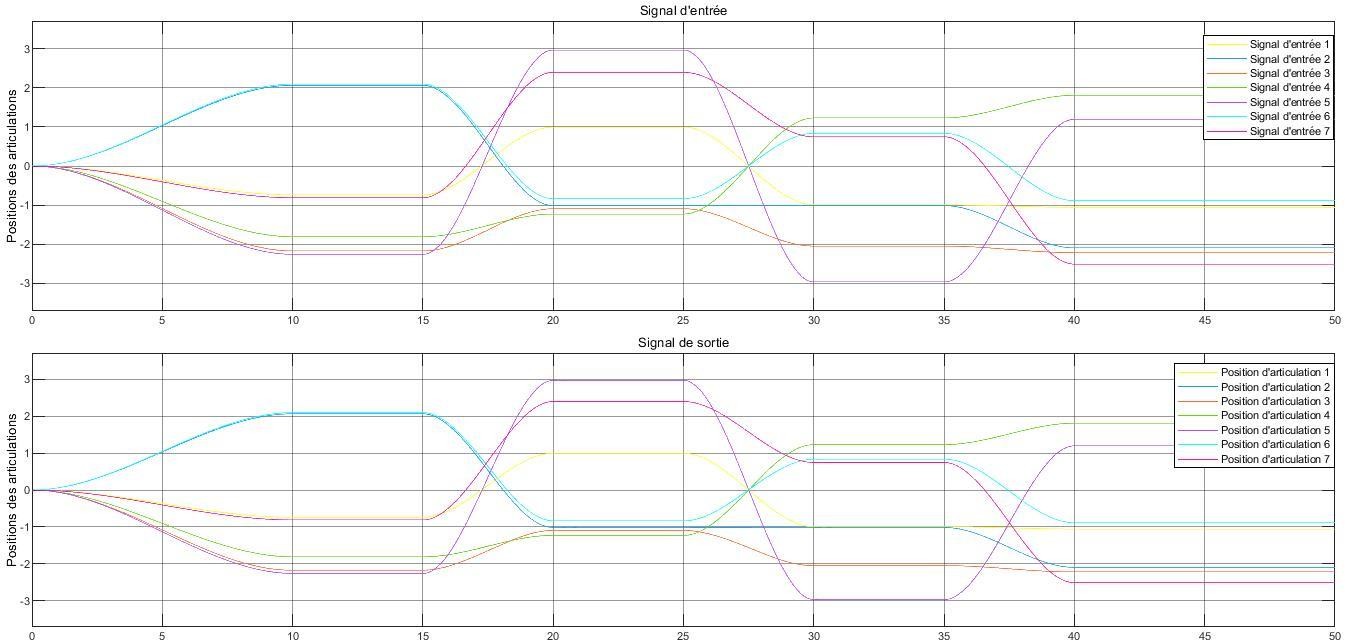
Les solutions des articulations :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1** | **Q2** | **Q3** | **Q4** | **Q5** |
| 0 | -0.7372 | 0.9939 | -0.9939 | -1.0569 |
| 0 | 2.0632 | -1.0135 | -1.0135 | -2.0944 |
| 0 | -2.1726 | -1.0935 | -2.0480 | -2.2173 |
| 0 | -1.8091 | -1.2288 | 1.2288 | 1.8091 |
| 0 | -2.2587 | 2.9671 | -2.9671 | 1.1948 |
| 0 | 2.0944 | -0.8352 | 0.8352 | -0.8866 |
| 0 | -0.8112 | 2.3958 | 0.7458 | -2.5090 |

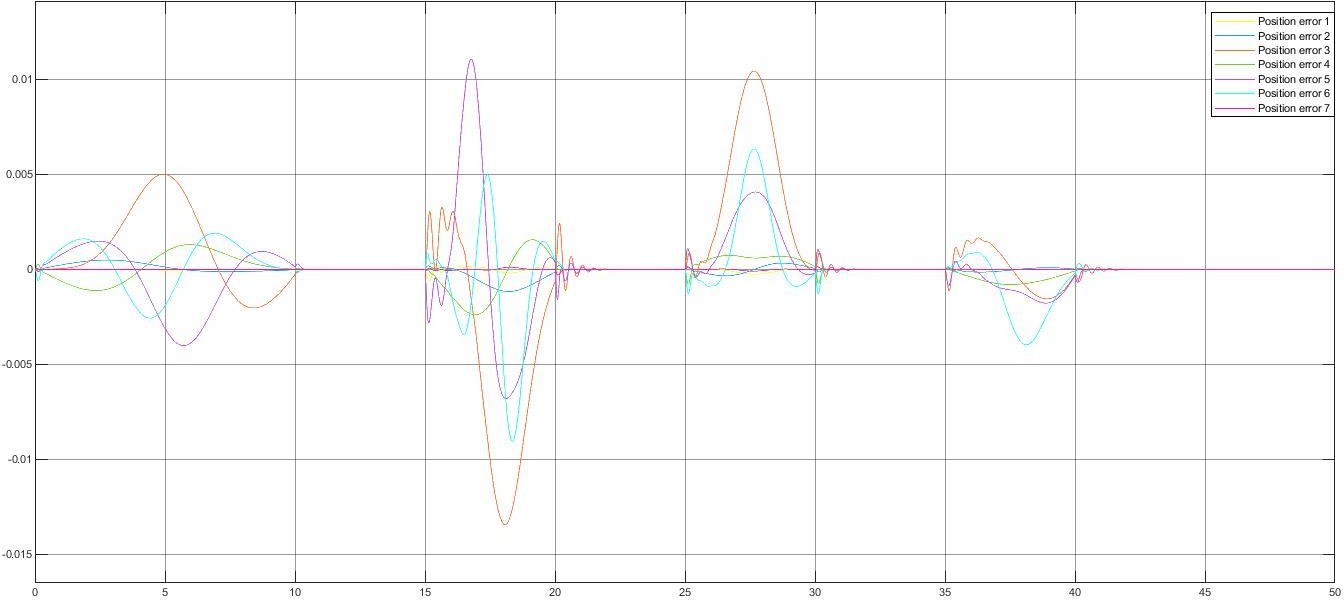
### Trajectoire commandée avec la commande PID :



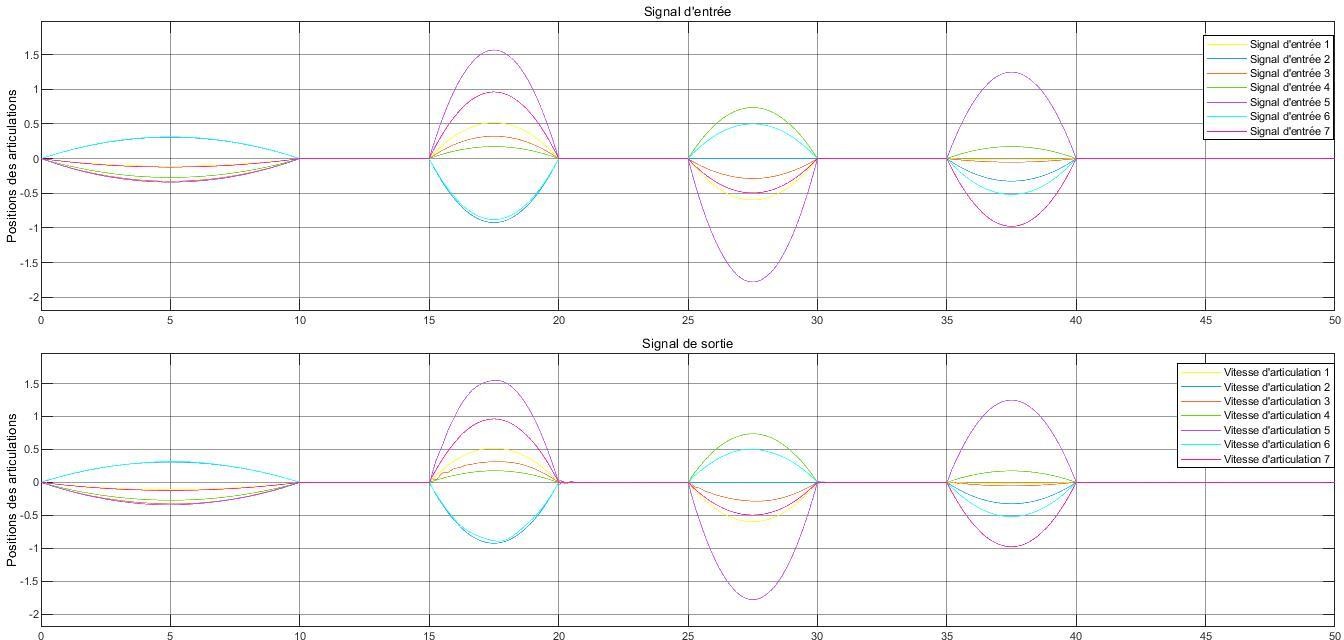
**Figure 4-1 :** trajectoire de l’effecteur du robot



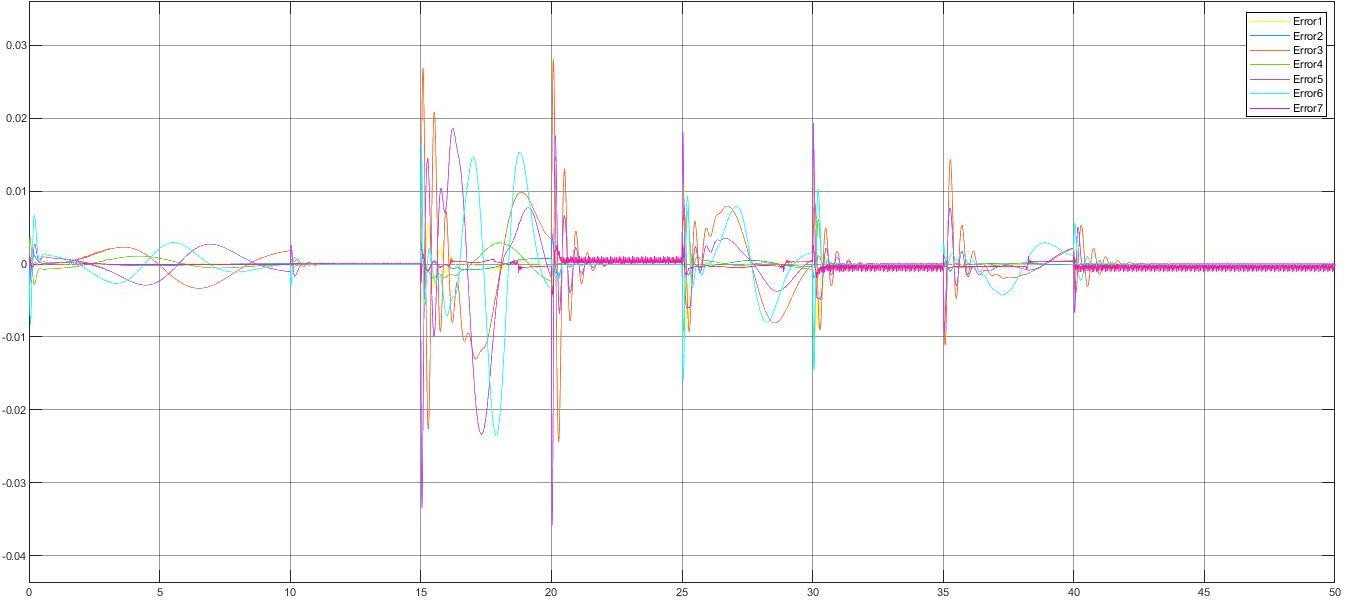
**Figure 4-2 :** Signal généré des trajectoires des positions et le signal de sortie des positions des articulations.



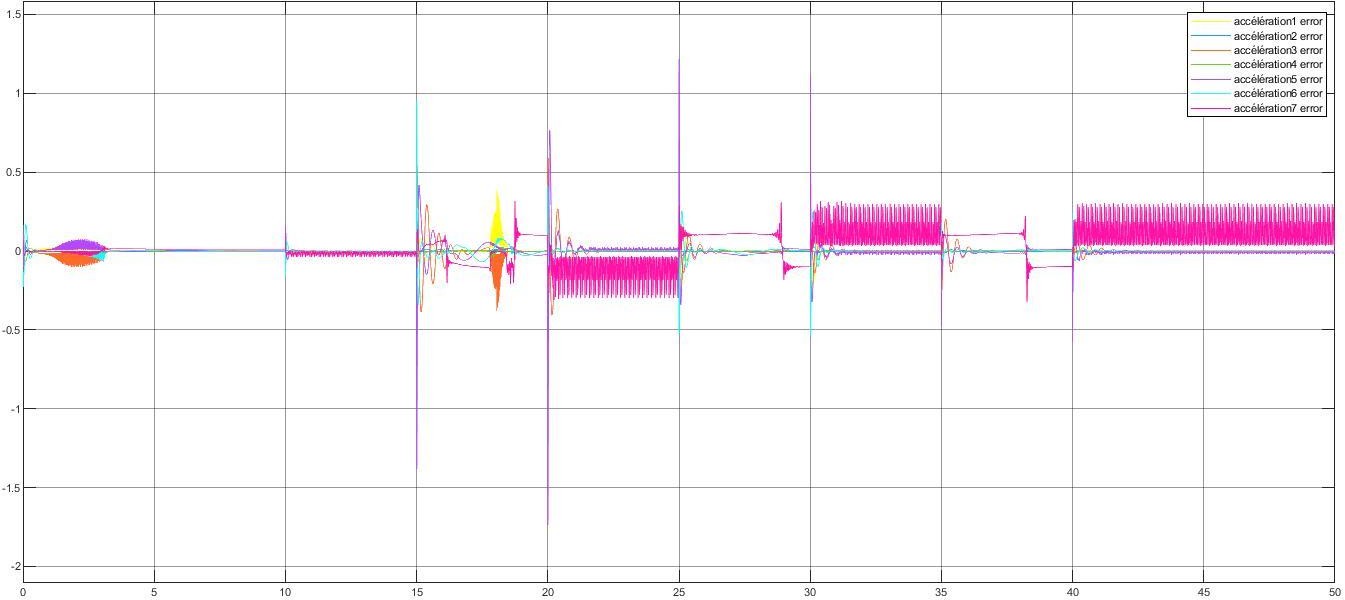
**Figure 4-3 :** Signal de l’erreur des positions.



**Figure 4-4 :** Signal généré des trajectoires des vitesses et le signal de sortie des vitesses des articulations.



**Figure 4-5 :** Signal de l’erreur des vitesses.

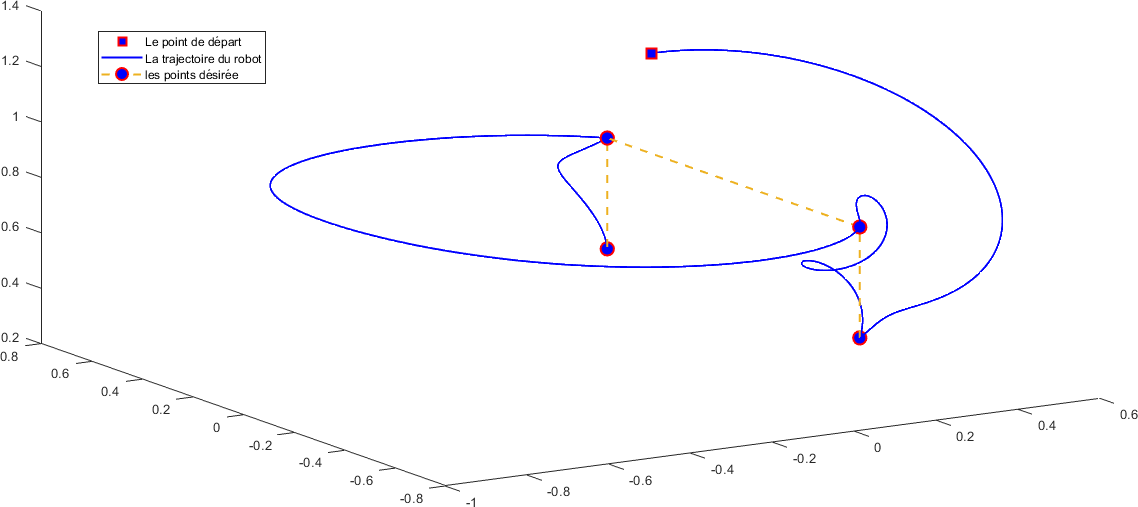


**Figure 4-6 :** Signal de l’erreur des accélérations.

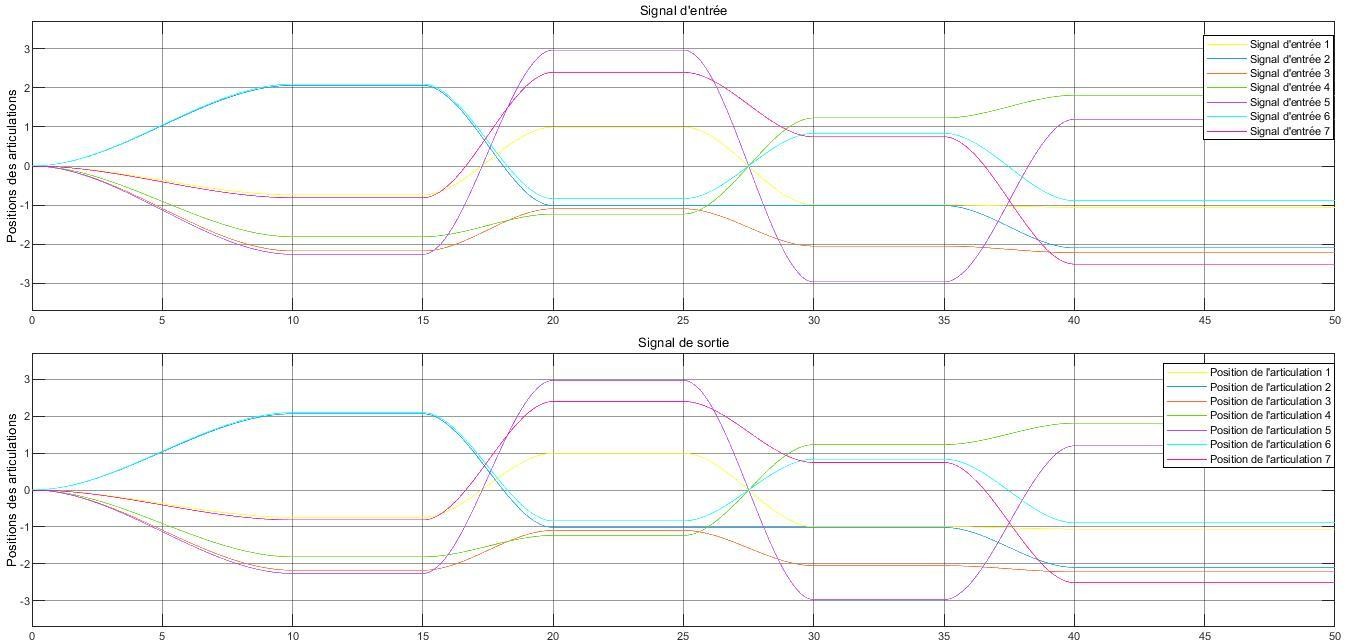
### Commentaire :

On remarque, que la commande PID décentralisée nous a données un bon suivi de trajectoire mais des erreurs importantes en positions des articulations et en vitesse et accélérations.

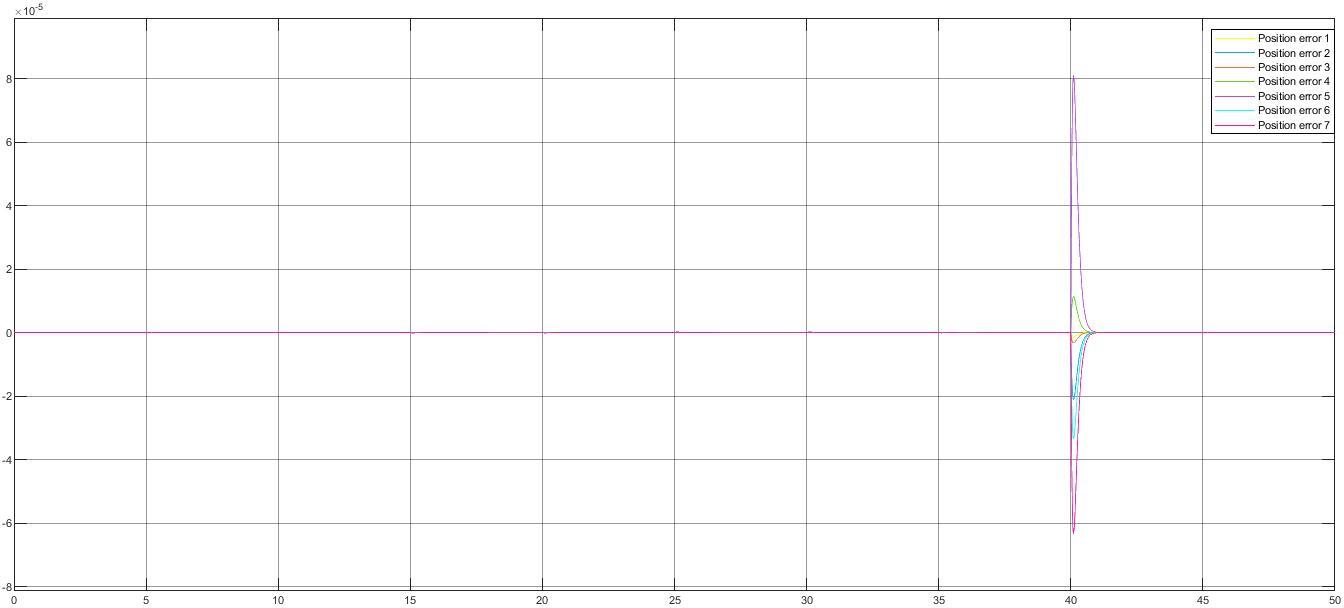
### Trajectoire commandée avec la commande Par Couple Calculé :



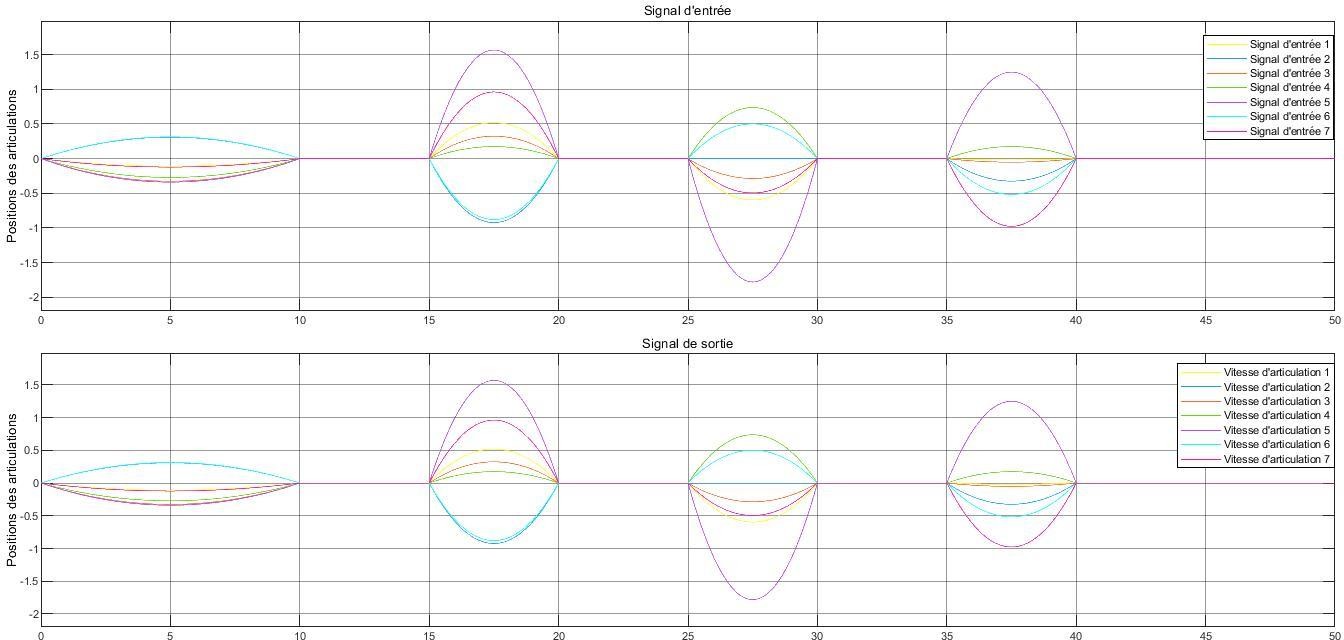
**Figure 4-7 :** trajectoire de l’effecteur du robot



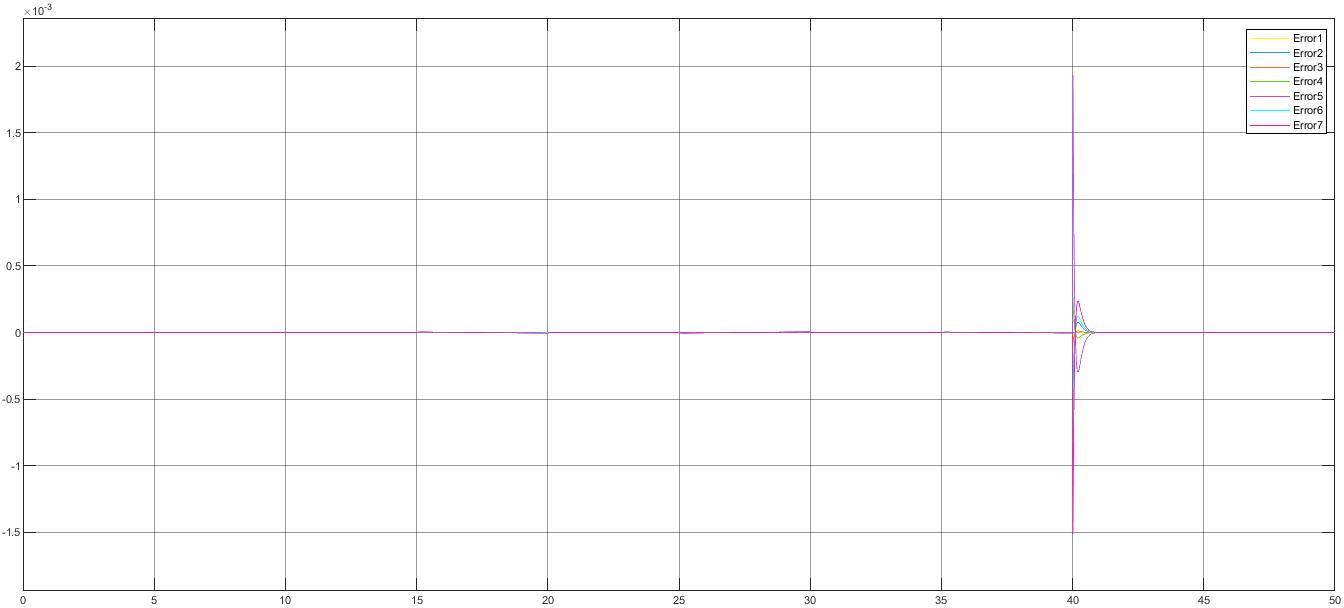
**Figure 4-8 :** Signal généré des trajectoires des positions et le signal de sortie des postions des articulations.



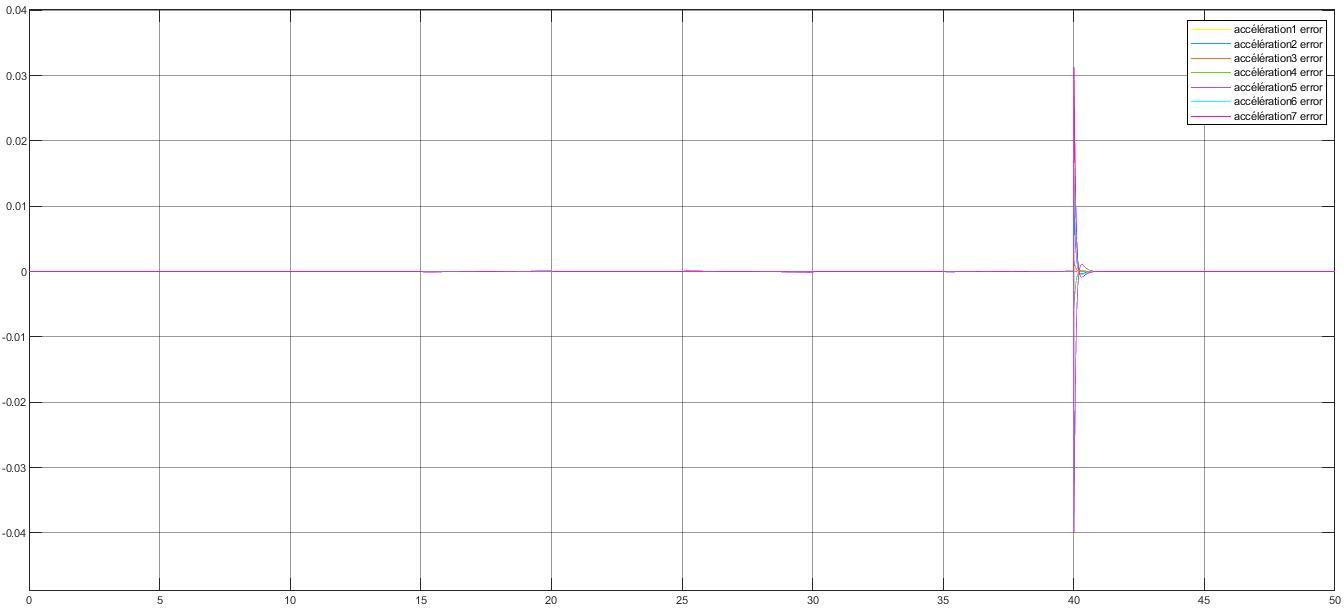
**Figure 4-9 :** Signal de l’erreur des positions.



**Figure 4-10 :** Signal généré de trajectoire des vitesses et le signal de sortie des vitesses des articulations.



**Figure 4-11 :** Signal de l’erreur des vitesses.

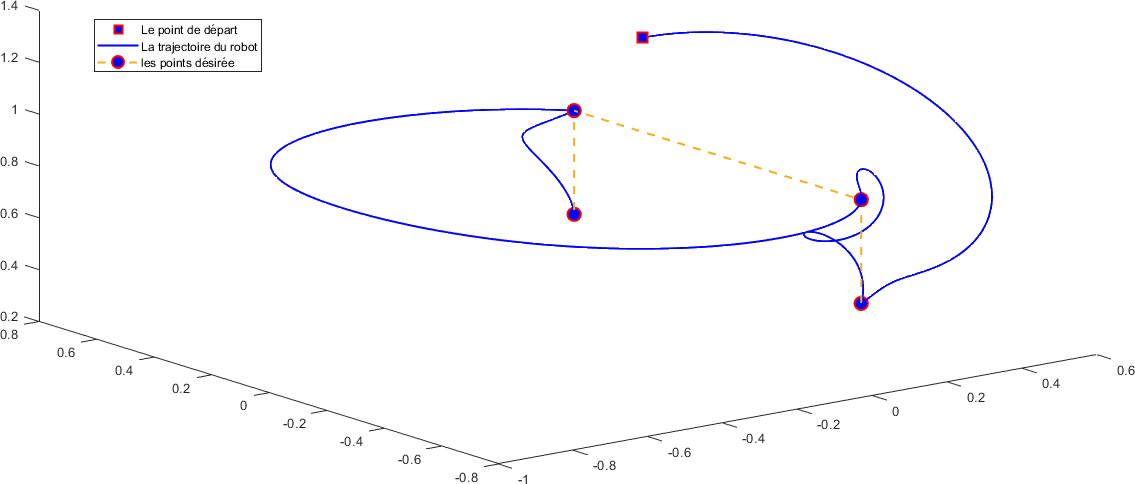


**Figure 4-12 :** Signal de l’erreur des accélérations.

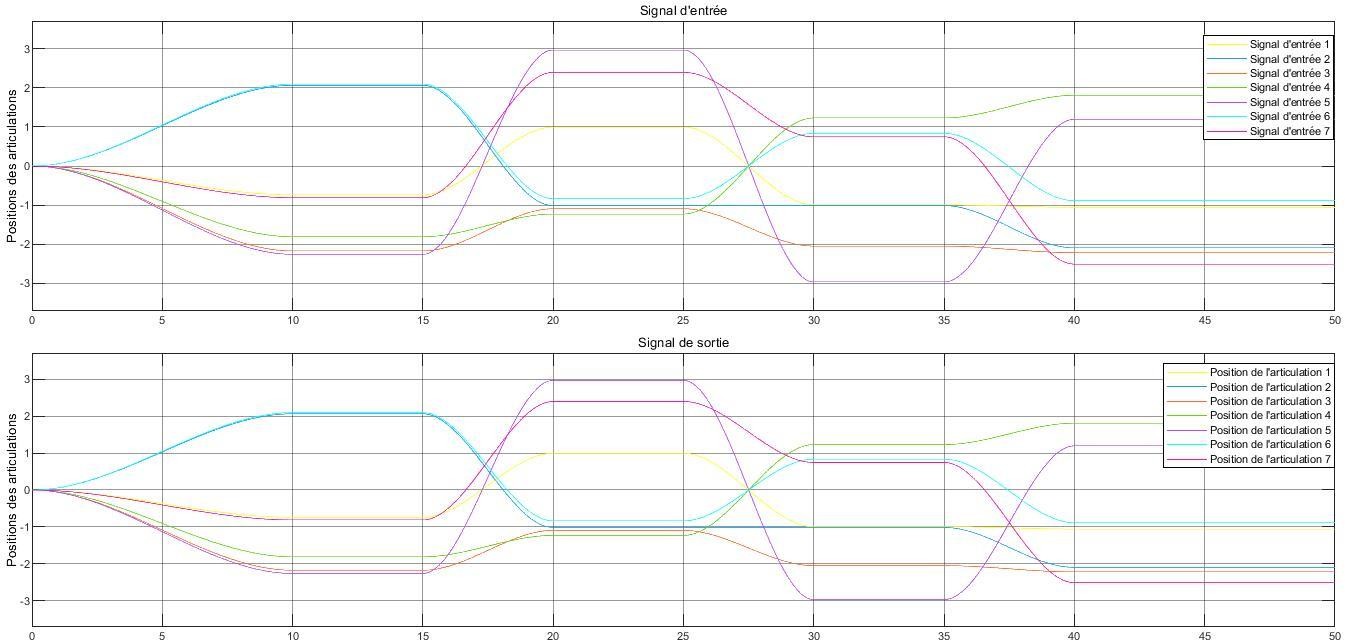
### Commentaire :

On remarque, que la commande par couple calculé a donné un bon suivi de trajectoire et une erreur presque nulle en position, vitesse et accélération.

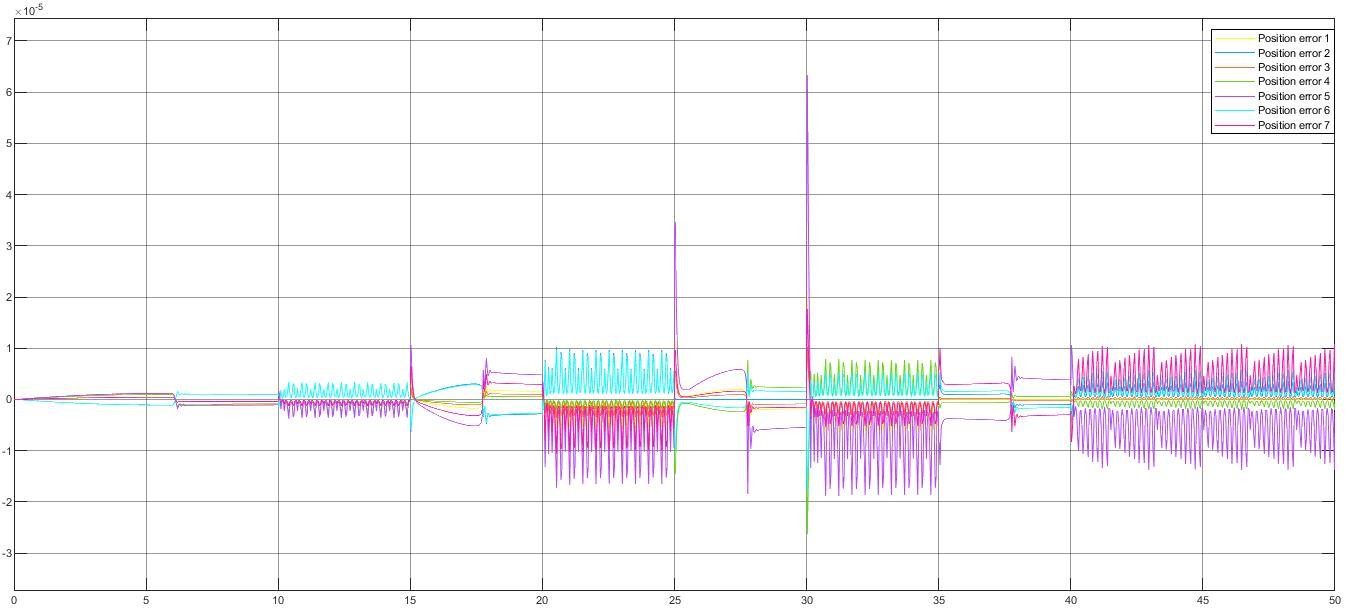
### Trajectoire commandée avec la commande par mode glissant :



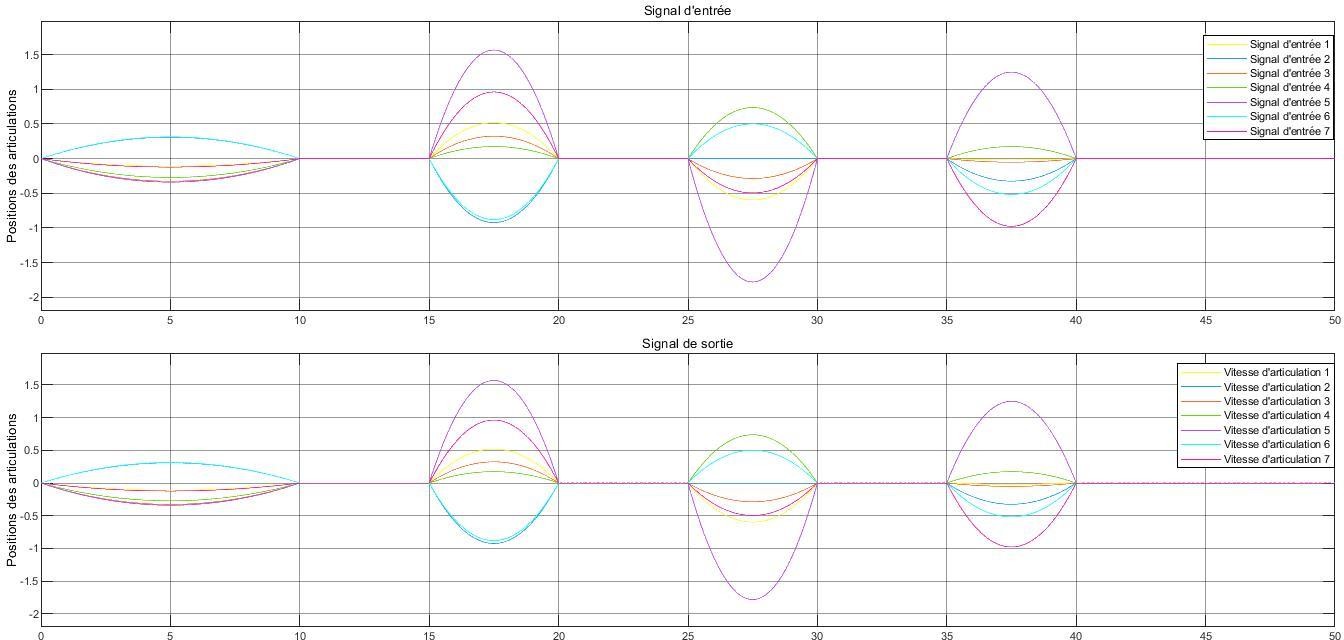
**Figure 4-13 :** trajectoire de l’effecteur du robot



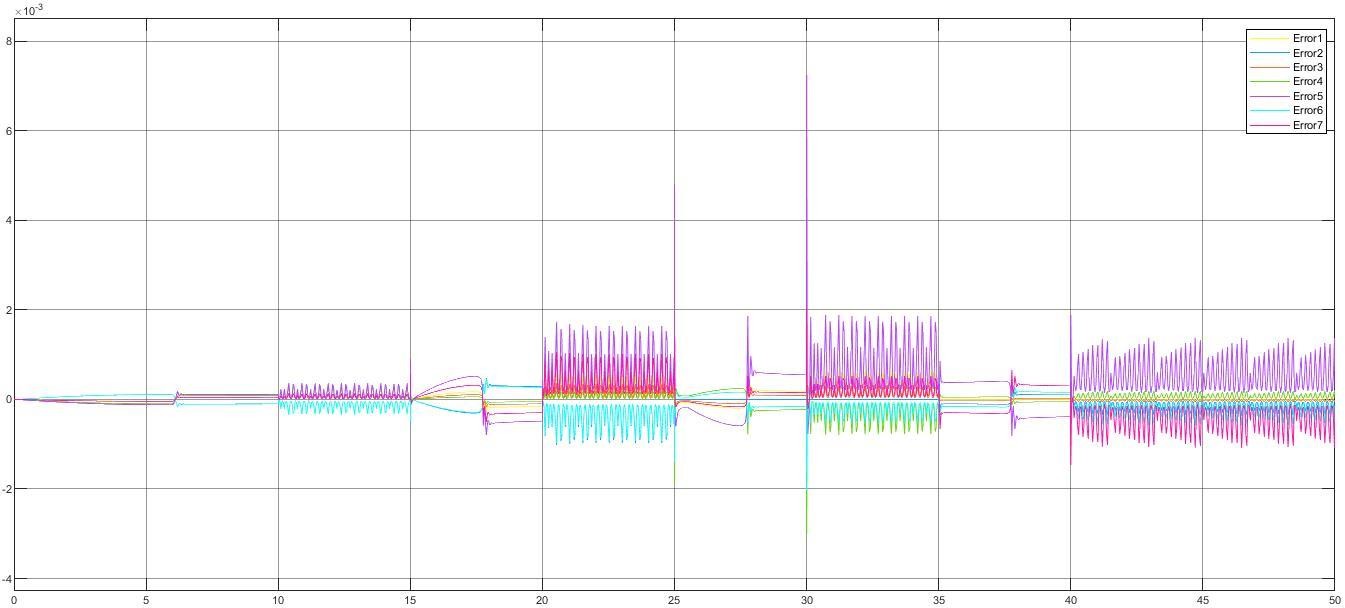
**Figure 4-14 :** Signal généré des trajectoires des positions et le signal de sortie des postions des articulations.



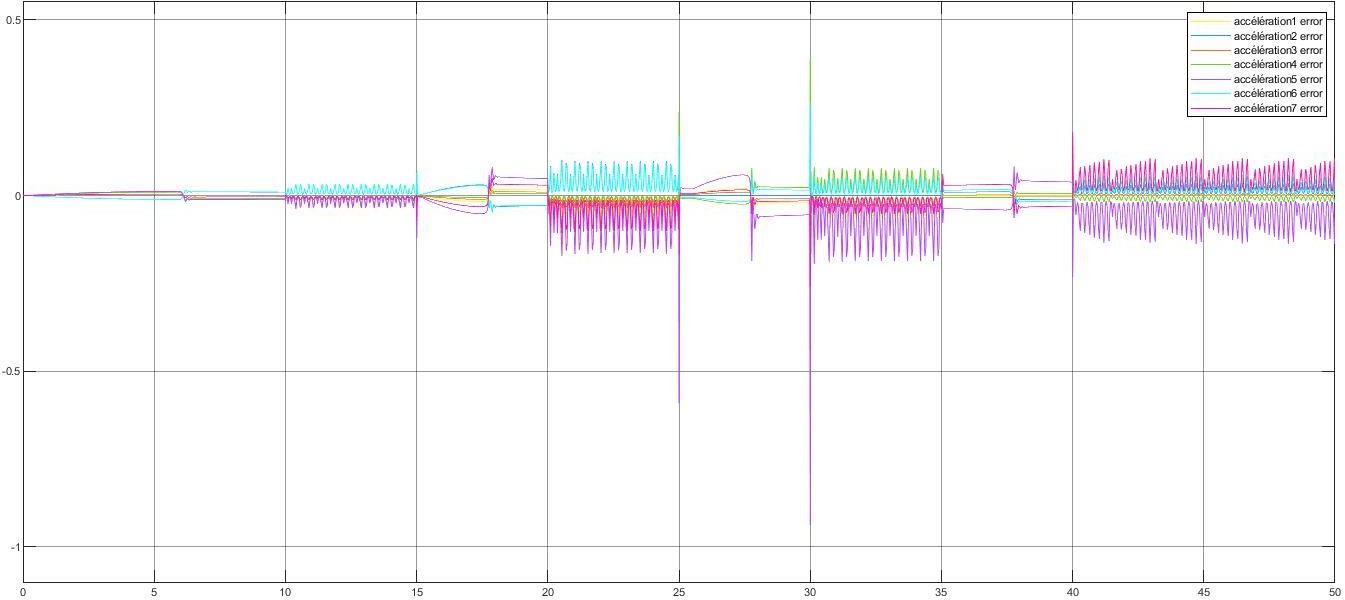
**Figure 4-15 :** Signal de l’erreur des positions.



**Figure 4-16 :** Signal généré de trajectoire des vitesses et le signal de sortie des vitesses des articulations.



**Figure 4-17 :** Signal de l’erreur des vitesses.



**Figure 4-18 :** Signal de l’erreur des accélérations.

### Commentaire :

On remarque, que la commande par mode glissant a donné un bon suivi de trajectoire et une erreur presque nulle en position, vitesse et accélération.

## Conclusion

### Etude comparative des différentes commandes appliquées sur le robot :

Ce projet s’articule principalement autour de trois parties : Dans la première, les différentes transformations entre les deux espaces, cartésien et articulaire ainsi que la modélisation du robot LBR IIWA 14 ont été présentées et expliquées. Dans la deuxième, différents types de commande des robots manipulateurs ont été présentés. Le premier type concerne les approches qui ne sont pas basées sur le modèle du robot, à savoir, la commande PID. Le deuxième type présente les approches qui sont à base du modèle, à savoir, la commande par couple calculé et la commande par mode glissant. Celles-ci ont été appliquées sur le robot LBR IIWA 14 afin de le commander et d’étudier leurs performances. La stabilité du système en boucle fermée est démontrée, pour les deux dernières commandes, en utilisant la théorie de Lyapunov. La dernière partie concerne la génération de trajectoire que les articulations de ce robot, pilotées par les lois de commandes traitées précédemment, doivent suivre.

On en conclut que le recours à la commande PID peut assurer des résultats acceptables mais de moindre qualité par rapport la commande par couple calculé et par mode glissant en termes de

performance, d’asservissement et de suivi de trajectoire, de précision, vitesse et accélération. Bien qu’elles soient très performantes et efficaces dans le cas où le modèle exact du système est connu, ces deux dernières commandes présentent un certain degré de complexité, rendant difficile leur mise en œuvre. Le calcul du modèle dynamique en ligne nécessaire au bon fonctionnement de celles-ci

représente un inconvénient majeur de leur application dans l’industrie. C’est pourquoi, la commande PID étant plus beaucoup simple à implémenter et moins gourmande en calcul, est la plus répandue et utilisée.