

الجمهورية الجزائرية الديمقراطية الشعبية  
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA  
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : Sciences de L'ingéniorat  
Département : Electronique  
Domaine : Sciences et Techniques  
Filière : Automatique  
Spécialité : Automatique et Informatique Industrielle

## Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

Simulation et commande d'un robot manipulateur

Présenté par : Benlaribi Houssef Eddine

Encadrant : Mr Benmoussa Samir Grade : Professeur

Université : BADJI MOKHTAR Annaba

### Jury de Soutenance :

<b>Boulbtatech brahim</b>	<b>Professeur</b>	<b>UBM Annaba</b>	<b>Président</b>
<b>Laafifi mourad</b>	<b>Professeur</b>	<b>UBM Annaba</b>	<b>examineur</b>
<b>Benmoussa Samir</b>	<b>Professeur</b>	<b>UBM Annaba</b>	<b>Encadreur</b>

Année Universitaire : 2020/2021

:

## Résumé :

Dans ce projet de fin d'étude, nous nous sommes intéressés à la planification de trajectoire d'un robot industriel type SCARA à deux degrés de liberté. Pour cela nous avons utilisé un modèle CAO issu de la plateforme Solidworks et en association avec Simulink/Matlab, un modèle qui sera utilisé pour la suite dans la simulation est généré. La commande du robot est assurée par le modèle géométrique inverse.

## Abstract:

In this end of study project, we are interested in trajectory planning of a SCARA industrial robot with two degrees of freedom. For this, we have used a CAD model from the Solidworks platform and in combination with Simulink/Matlab, a model for simulation is obtained. The Inverse geometric model is used for the robot control.

## ملخص:

في نهاية مشروع الدراسة هذا ، نحن مهتمون بتخطيط مسار الروبوت الصناعي SCARA بدرجتين من الحرية. لهذا ، استخدمنا نموذج CAD من منصة Solidworks وبالاقتران مع Simulink / Matlab ، تم الحصول على نموذج للمحاكاة. يستخدم النموذج الهندسي العكسي للتحكم في الروبوت.

## **Remerciement :**

Je tiens à remercier énormément Monsieur  
Benmoussa pour sa générosité.

Et aux messieurs les membres de jury d'avoir  
accepté d'évaluer ce mémoire.

## **Introduction générale :**

L'homme a depuis toujours été fasciné par la possibilité de pouvoir recréer la vie. Désir profond de jouer les créateurs, ou bien tentation de la mystification, ou encore simple jeu de copie de l'existant, les divers hommes d'airain, androïdes ou robots, créés dans la littérature, au cinéma ou dans la réalité, ont depuis la haute antiquité répondu à nos fantasmes.

La robotique s'agit d'un terme facile à définir, mais dont la réelle portée est souvent difficile à appréhender. La robotique désigne l'ensemble des techniques mises au point et exécutées pour fabriquer des robots. Derrière cette signification se cachent des possibilités infinies et un grand, un très grand éventail de champs d'application.

Dans ce projet, nous nous sommes intéressés à la planification de trajectoire d'un robot type SCARA à deux degrés de liberté. La plateforme Solidworks et en association avec Simulink/Matlab est utilisée pour avoir le modèle CAO du robot, à partir de lequel, un modèle dynamique du robot est obtenu. Pour la commande, un modèle géométrique (direct et indirect) est implémenté.

# **Chapitre 1 : Robotique**

## **Qu'est-ce qu'un robot :**

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) conçu pour accomplir automatiquement des tâches imitant ou reproduisant, dans un domaine précis, des actions humaines. La conception de ces systèmes est l'objet d'une discipline scientifique, branche de l'automatique nommée robotique. Il est composé de 3 éléments : • Les capteurs : laser ou ultrason permet au robot de prendre conscience de son environnement, les caméras pour traiter l'environnement et avoir une vision à 3D. • Les circuits de commande : des circuits électroniques que ce soit un microcontrôleur (MC) ou microprocesseur (MP), il contient la programmation du robot • Les actionnaires : les moteurs électriques rotatif (CC/CA ou pas à pas) et les vérins

## **La robotique :**

La robotique est une activité multidisciplinaire visant l'étude, la conception et la construction de robots ou plus simplement de machines automatiques. Sa pratique réunit des savoir-faire techniques et des connaissances scientifiques des domaines de l'électronique, de l'informatique et de la mécanique. Des progrès dans la technologie se produisent et il devient évident que les ordinateurs, les machines, les robots et les algorithmes seront en mesure de faire la plupart des emplois de routine et répétitifs. Les ingénieurs en robotique conçoivent la prochaine génération de robots qui ressemblent, ressentent et agissent plus comme nous, humains. Les ingénieurs font des cheveux et une peau réaliste avec des capteurs qui permettront aux robots de réagir en tant qu'humain, les robots pourront se sentir touchés sur leur épaule et se tourner pour vous accueillir. Ils auront également des yeux artificiels qui vont se déplacer et cligner des yeux comme des objets naturels, leur coffre se déplacera légèrement qui stimulera la respiration et ils auront des muscles fabriqués par l'homme qui changeront 12 leurs expressions faciales. À l'avenir, nous pouvons également avoir des animaux de compagnie robotiques avec une intelligence artificielle [1].

## **Historique :**

De tous temps, l'homme a cherché à se faire remplacer pour des tâches spécifiques pouvant être dangereuses, rébarbatives, longues ou répétitives. Ainsi, il faut remonter jusqu'à la préhistoire pour trouver les premiers automatismes avec les premiers pièges. Ainsi, un collet à déclencheur est un bon exemple de mécanisme doté d'un détecteur et d'un actionneur. Unimate est le premier robot industriel créé. Il fut intégré aux lignes d'assemblage de General Motors en 1961. En 1970, le robot lunaire Lunokhod 1, envoyé par l'Union soviétique, a voyagé sur une distance de 10 km et a transmis plus de 20 000 images

## **Les grands constructeurs des robots industriels :**

- **Adept Technology Inc.** : une société américaine dont le siège est en Californie, industrielle. Fondée en 1983 leur robots sont utilisés dans la fabrication de précision et à haute vitesse, l'emballage et l'automatisation industrielle.

- **Kuka** : société dont le siège est à Augsburg, en Allemagne son premier robot industriel en 1973 utilisé beaucoup plus dans le domaine d'automobile et la métallurgie

- **ABB** : est une société multinationale dont le siège est à Zürich suisse à cause de l'expérience elle a prêté une grande place en automatisation et beaucoup plus dans le domaine industriel avec 4600 employés dans 53 pays et l'installation plus de 250 000 robots dans le monde entier.

- **Stäubli** : est une société suisse, elle a commencé le domaine robotique en 1982, SCARA, robot à 4 même à 6 axes (ddl) destiné vers des différentes matières plastiques, l'électronique, l'énergie photovoltaïque. [2]

### **Robot industriel :**

On appelle robot industriel un ensemble « autonome » constitué de trois éléments :

- **Le bras** : Est un élément mécanique et le responsable de mouvement du robot et en cite 4 grandes familles : -Robots SCARA 3 ou 4 axes 1 -Robots poly articulés (4 à 7 axes) 2 -hexapode 3 - double bras

- **L'armoire de commande** : C'est l'élément électronique qui gère le mouvement d'un robot, il peut être un automate ou une commande numérique il contient le programme de travail du robot pour commander son déplacement et aussi les relations avec les actionnaires

- **Le pupitre de programmation** : Est appelé aussi boîtier opérateur il s'agit du panneau de commande déporté du robot qui permet d'effectuer la programmation par apprentissage. Il comporte généralement un écran d'affichage, des boutons de commande et un dispositif de mise en mouvement du robot.



Figure 1.1 : quelques robots industriels (poly articulés, hexapode, double bras)

### **Domaine d'application :**

Ils effectuent sans relâche des tâches répétitives et avec rigueur. Dans les chaînes de montage de l'industrie automobile, ils y remplacent les ouvriers dans les tâches pénibles et dangereuses : peinture, soudage, emboutissage, etc.. Les robots industriels sont souvent munis de systèmes de vision qui leur procurent une souplesse d'exécution et des moyens de vérifier la qualité des produits fabriqués

## Les avantages de la robotique :

Les robots ont des avantages dans notre vie ; ils font leur travail avec une haute rapidité et précision, la rareté des erreurs, beaucoup de production dans un court temp et ils travaillent sans pause (sauf pour une petite maintenance ou entretien), il travaille aussi dans des conditions difficiles et dangereuses, et aussi sont conviviaux, intelligents et abordables [1]

## Les éléments d'un robot :

Dans un robot industriel en trouve : Base, axe (articulation), Actionneur, corps, organe terminal, effecteur.

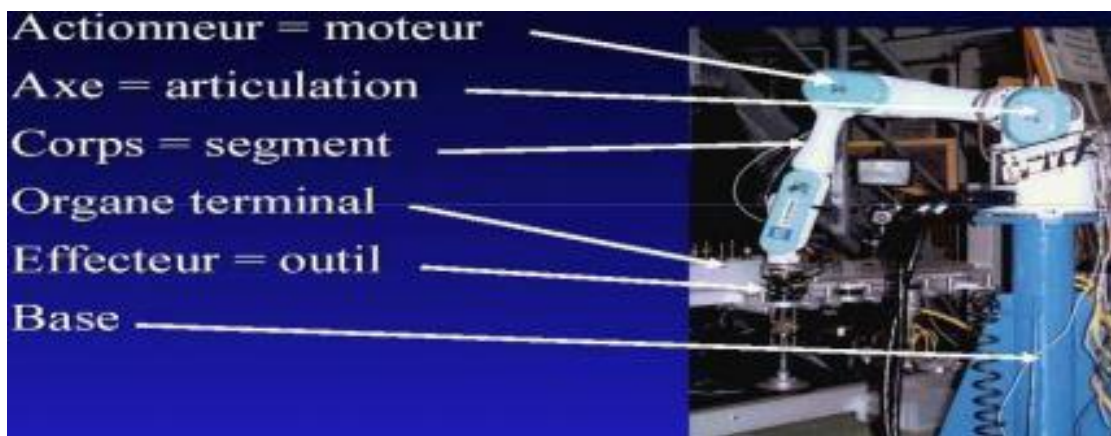


Fig1.2 : Les éléments d'un robot.

## Morphologie des robots :

Dans un robot en façon générale il y'as 2 types d'articulation, rotorique à l'aide d'un moteur et prismatique à l'aide d'un vérin ou un moteur avec une vis sans fin. Le porteur ne possède que 3 ddl. Les ddl résiduels forment l'organe terminal caractérisé par des dimensions beaucoup plus petites et une plus faible masse. Nous avons donc 5 classes de morphologies : Articulé (RRR), cartésien (PPP), Cylindrique (RPP), Sphérique (RRP), SCARA [5]



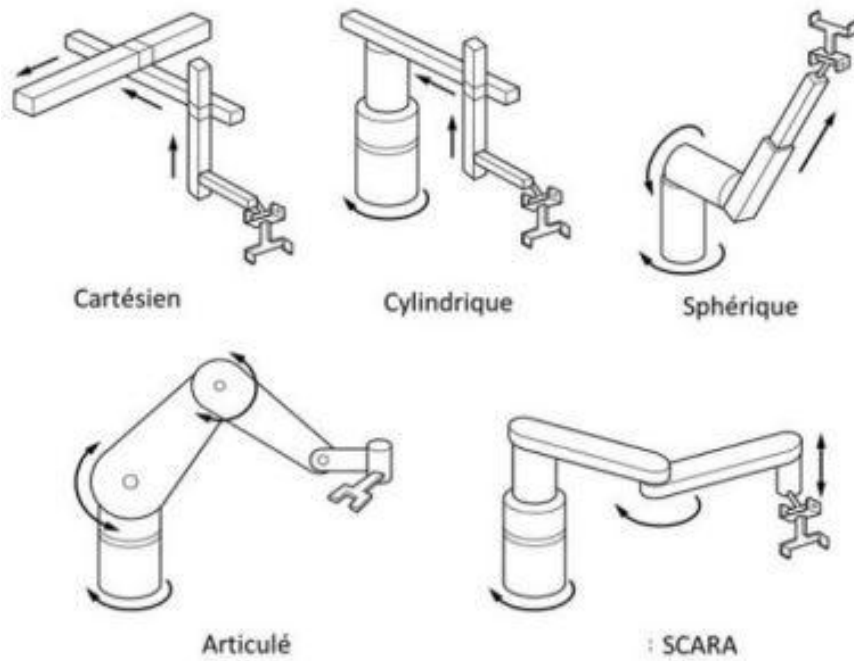


Fig1.3 classifications de morphologie des robots manipulateurs

## Classification des robots :

En robotique il y'a 3 types de robot :

### • Robot manipulateur :

Est un robot a plusieurs ddl (2,3,4et 6) reprogrammable, On les trouve dans les industries car il est destiné à faire des taches dangereuses. Il est rapide et précis .

### • Robot mobile :

Un robot mobile est un système mécanique, électronique et informatique agissant physiquement sur son environnement en vue d'atteindre un objectif qui lui a été assigné. Cette machine est polyvalente et capable de s'adapter à certaines variations de ses conditions de fonctionnement. Elle est dotée de fonctions de perception, de décision et d'action. Ainsi, le robot devrait être capable d'effectuer des tâches diverses, de plusieurs manières, et accomplir correctement sa tâche, même s'il rencontre de nouvelles situations inattendues [4].

### • Robot humanoïde :

On les appelle robot domestique à cause de son apparence a 60% humaine avec une tête deux bras manipulateurs et deux jambes.



Fig1.4 (a)robot humanoïde,(b)robot manipulateur(c)robot mobile

### **Modélisation des robot manipulateurs :**

L'étude de mouvement des robots manipulateur il est devenu une plateforme très importante dans le domaine de recherche donc certain professeur en informatique, mathématique et robotique en créant la modélisation des robots qui est consisté a représenté certain robot dans un modèle mathématique et informatique pour étudier ses mouvements

### **Modèle géométrique direct :**

Le modèle géométrique direct (MGD) d'un robot permet de calculer à chaque instant  $t$ , la position et l'orientation de l'organe terminal d'un robot fonction de ses variables d'articulation (les consignes des actionneurs) et ses paramètres géométriques. Aussi, le MGD est utilisé dans la définition de l'espace de travail du robot en utilisant les limites minimales et maximales des articulations motorisées. Il existe une méthode la plus utiliser en robotique c'est de Denavit-Hartenberg (DH 1955) est la suivante :

- Tout d'abord on numérise les articulations et définie les repères
- Le passage d'un repère  $i-1$  vers le repère  $i$  est exprimé à l'aide des paramètres suivant : –  $a_i$  : la distance entre  $Z_{i-1}$  et  $Z_i$  suivant l'axe  $X_i$ . –  $\alpha_i$  : l'angle formé par  $Z_{i-1}$  et  $Z_i$  suivant l'axe  $X_i$ . –  $d_i$  : la distance entre  $X_{i-1}$  et  $X_i$  suivant  $Z_{i-1}$ . –  $\theta_i$  : l'angle formé par  $X_{i-1}$  et  $X_i$  suivant  $Z_{i-1}$ . Le calcul de la matrice de transformation homogène de chaque repère de  $i-1$  vers  $i$  est la suivante :1.1

$${}^{i-1}T_i = \begin{matrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & \alpha_{i-1} \\ \sin(\theta_i)\cos(\alpha_i) & \cos(\theta_i)\cos(\alpha_i) & -\sin(\alpha_i) & -di\sin(\alpha_i) \\ \sin(\theta_i)\sin(\alpha_i) & \cos(\theta_i)\sin(\alpha_i) & \cos(\alpha_i) & di\cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{matrix}$$

Après on trouve La matrice de transformation homogène associée au robot à l'aide de cette équation :

$${}^0_nT = {}^0_1T \cdot {}^1_2T \dots {}^{n-2}_{n-1}T \cdot {}^{n-1}_nT$$

Cette dernière Peut s'écrire sous la forme suivante :

$${}^0_nT = \begin{matrix} {}^0_nR & {}^0_nP \\ 0 & 1 \end{matrix}$$

### Model géométrique inverse :

Le modèle géométrique inverse (MGI) est utilisé pour générer l'ensemble des consignes des actionneurs pour que l'effecteur atteigne une position et des orientations désirées. Il existe plusieurs méthodes et procédures numériques pour le calcul du MGI, dans ce cours, nous nous intéressons à la méthode de Paul.[5] On a un robot défini par sa matrice de transformation homogène est la suivante :

$${}^0_nT = {}^0_1T(q_1) + {}^1_2T(q_2) \dots {}^{n-2}_{n-1}T(q_{n-1}) + {}^{n-1}_nT(q_n)$$

U0 est la position désire exprime par :

$$U_0 = \begin{matrix} R_{ref} & P_{ref} \\ 0 & 1 \end{matrix}$$

Pour l'obtention de MGI on suit la méthode de Paul la suivante : • Multiplier les deux côtés de l'équation  ${}^0_nT = U_0$  par  ${}^0_1T$  on obtient :

$${}^1_0T = {}^1_2T \cdot {}^2_3T \dots {}^{n-1}_nT$$

On résoudre cette dernier pour on trouve q1

•q2 est obtenu en résolvant l'équation :

$${}^2_1T \quad {}^1_0T \quad {}^0U = {}^2_3T \dots \quad {}^{n-1}_nT$$

- Suivre cette procédure jusqu'à ce que toutes les variables soient déterminées.

### Modèle cinématique :

La modélisation cinématique consiste à exprimer la vitesse des coordonnées opérationnelles en fonction des vitesses des articulations :

$$X \cdot = J(q) \cdot q \cdot = \begin{matrix} V_n \\ \omega_n \end{matrix}$$

$J(q)$  : désigne la matrice jacobienne du mécanisme de dimension

$V_n$  : désigne la vitesse de translation de l'origine  $o_n$  du repère  $R_n$ , égale à est la dérivée par rapport au temps du vecteur

$\omega_n$  : désigne la vitesse de rotation du repère  $R_n$

L'intérêt de la matrice jacobienne est multiple :

– elle est à la base du modèle différentiel inverse, permettant de calculer une solution locale des variables articulaires  $q$  connaissant les coordonnées opérationnelles  $P$

– en statique, on utilise le jacobien pour établir la relation liant les efforts exercés par l'organe terminal sur l'environnement aux forces et couples des actionneurs ; Donc la relation entre les vitesses des articulations d'un robot  $q \cdot$ , et la vitesse de l'effecteur du robot  $P \cdot$  est calculé par l'équation suivante :

$$P \cdot = J \cdot q \cdot$$

Ainsi, le modèle cinématique du robot est obtenue [5]

### Modèle cinématique inverse :

La modélisation cinématique inverse est de calculer à partir d'une configuration  $q$  donnée, les vitesses articulaires  $q \cdot$  qui assurent à l'organe terminal une vitesse opérationnelle  $X \cdot$  imposée, donc un MCI est obtenu en inversant le MCD et à partir de la résolution d'un système d'équations linéaires, en cas normal (une matrice Jacobienne carrée de déterminant non nul) le MCI est obtenu comme suit :

$$q \cdot = J^{-1} X \cdot_{ref}$$

### 11.5 Modélisation dynamique :

Les deux modèles qui permettent de représenter la dynamique du robot sont les suivants :

– **Modèle Dynamique Direct (MDD) :**

Consiste à déterminer comment le robot va réagir sous l'application d'un ensemble de couples actionneurs. En d'autres termes, ayant le vecteur  $\tau$ , il s'agit de trouver  ${}''q$  f  $q, q, \dot{q}$  ( $\tau$  )

– **Modèle Dynamique Inverse (MDI) :**

Permet de trouver le vecteur  $\tau$  de couples aux actionneurs nécessaires pour répondre à une trajectoire dont  $q, \dot{q}$  et  ${}''q$  sont données. Ce modèle est le plus utilisé dans le monde de la robotique car on pilote le robot par ses variables articulaires  $q, \dot{q}$  et  ${}''q$  et on cherche à trouver les couples  $\Gamma$  au niveau des articulations.

**Conclusion :**

Dans ce premier chapitre, nous avons présenté d'une manière succincte la robotique avec ses différents éléments. Par la suite, nous aborderons la modélisation du robot SCARA à 2 ddl.

**Chapitre II :**  
**Modélisation, génération de trajectoire**  
**et commande**

## Le Robot SCARA :

Un robot SCARA est défini dans la norme ISO 8373 : 1994, No.3.15.6, en tant que « robot comporte deux liaisons pivots parallèles pour fournir conformément à un plan sélectionné » Il peut être considéré comme un cas particulier d'un robot cylindrique. «SCARA» signifie «Selective Compliance Arm for Robot Assembly». (Autre interprétation comprend « Selective Compliance Articulated for Robot Assembly ».). Le robot SCARA est un robot manipulateur à 2 axes rotative (RR). L'enveloppe de travail cylindrique permet une utilisation maximale de l'espace de travail

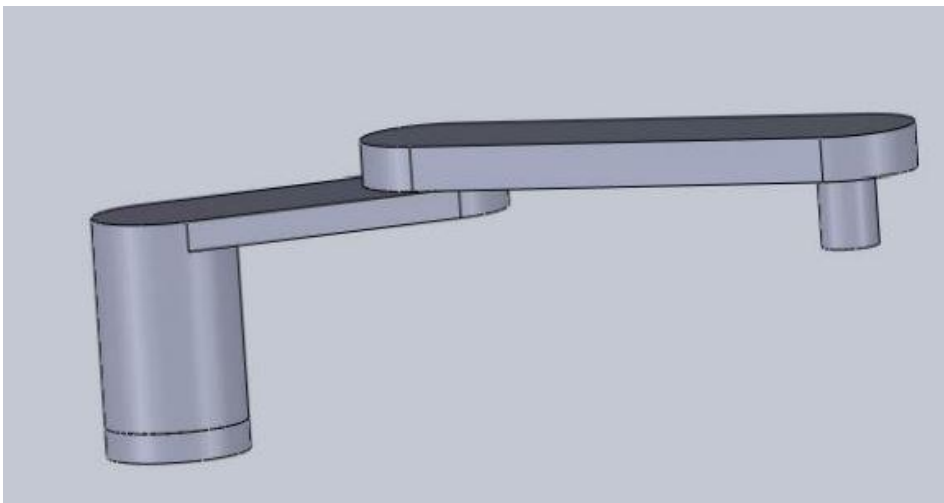


Fig.2.1 : Schéma représentatif d'un robot SCARA

SCARA a des avantages et des caractéristiques qui permettent d'être un robot industriel très utilisé comme :

- Grande précision et vitesse supérieur
- Maintenance minimale
- Plus facile à utiliser
- Rigidité supérieure
- Désigné très compact

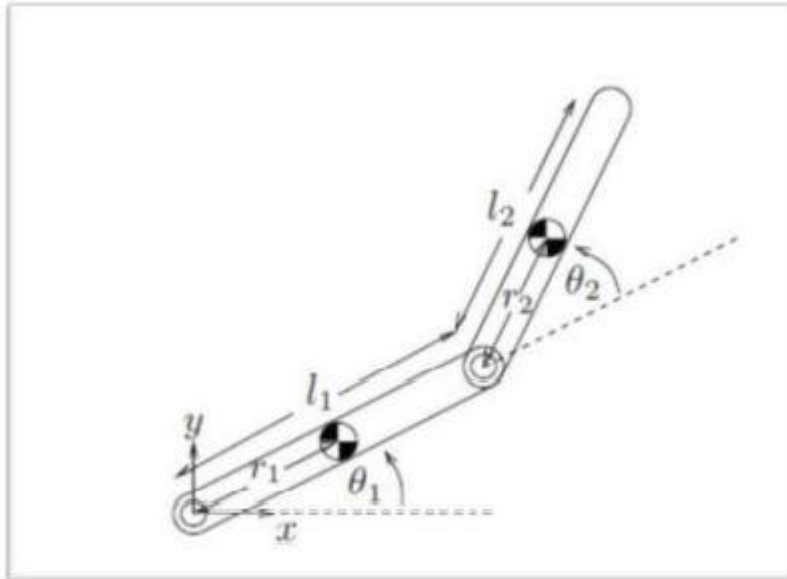


Fig2.2 : Bras a de degré de liberté (2DDL RR)

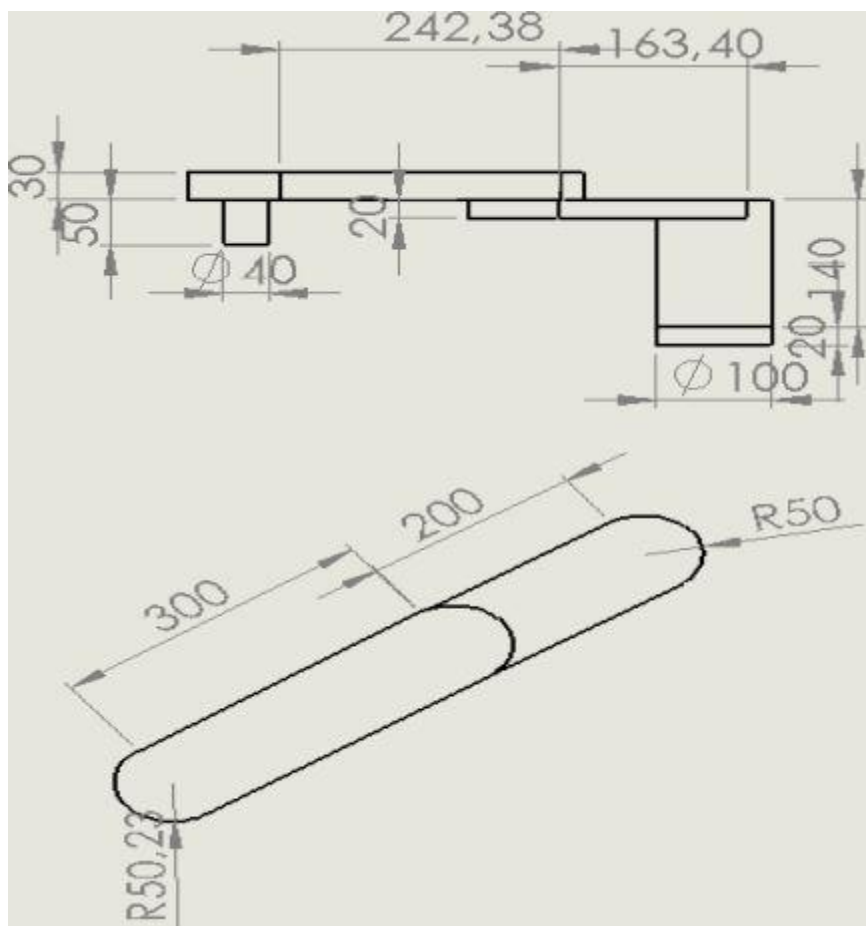


Fig2.3 : Dimension de robot SCARA 2ddl RR



## Modélisation géométrique du robot :

### Modèle géométrique direct :

Ce modèle permet de savoir la position exacte de l'organe terminal à partir de les angles  $\theta$  ; Dans notre cas notre robot RR en exprime la position P en fonction de  $\theta_1$   $\theta_2$ .

### Tableaux Paramètres de Denavit-Hartenberg :

Le scara a 2DDL rotative on peut la présenter dans le tableau de D-H comme suit : •  $a_i$  : la distance entre  $Z_{i-1}$  et  $Z_i$  suivant l'axe  $X_i$  ; •  $\alpha_i$  : l'angle formé par  $Z_{i-1}$  et  $Z_i$  suivant l'axe  $X_i$  ; •  $d_i$  : la distance entre  $X_{i-1}$  et  $X_i$  suivant  $Z_{i-1}$  ; •  $\theta_i$  : l'angle formé par  $X_{i-1}$  et  $X_i$  suivant  $Z_{i-1}$ .

Link	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	$l_1$	0	0	$\theta_1$
2	$l_2$	0	0	$\theta_2$

TAB1.2 : Tableau D-H

Avec :  $l_1=0.2m$  et  $l_2=0.3m$

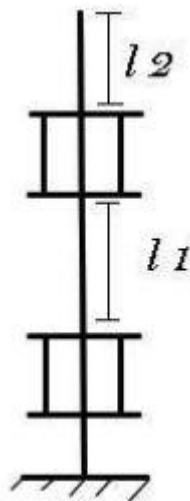


Fig2.4 : Configuration géométrique

Pour la calcul de MGD de SCARA on calcule la matrice de transformation entre 0 2T (R0 et R2). Pour cela, on réalise les calculs successifs suivants :

$${}^{i-1}T_i = \begin{matrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & \alpha_{i-1} \\ \sin(\theta_i)\cos(\alpha_i) & \cos(\theta_i)\cos(\alpha_i) & -\sin(\alpha_i) & -di\sin(\alpha_i) \\ \sin(\theta_i)\sin(\alpha_i) & \cos(\theta_i)\sin(\alpha_i) & \cos(\alpha_i) & di\cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{matrix}$$

$${}^0T_2 = {}^0T_1 \cdot {}^1T_2$$

Cos  $\theta_1 = C1$  et Sin  $\theta_1 = S1$  et Sin  $\theta_2 = S2$  et Cos  $\theta_2 = C2$  et Cos  $(\theta_1 + \theta_2) = C12$  et Sin  $(\theta_1 + \theta_2) = S12$

Les matrices de transformation sont les suivant :

$${}^0T_1 = \begin{matrix} \mathbf{Cos1} & \mathbf{-Sin1} & \mathbf{0} & \mathbf{0} \\ \mathbf{Sin1} & \mathbf{Cos1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{matrix} \quad {}^1T_2 = \begin{matrix} \mathbf{Cos2} & \mathbf{-Sin2} & \mathbf{0} & \mathbf{l1} \\ \mathbf{Sin2} & \mathbf{Cos2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{matrix}$$

$${}^0T_2 = {}^0T_1 \cdot {}^1T_2 = \begin{matrix} \mathbf{sx} & \mathbf{nx} & \mathbf{ax} & \mathbf{Px} \\ \mathbf{sy} & \mathbf{ny} & \mathbf{ay} & \mathbf{Py} \\ \mathbf{sz} & \mathbf{nz} & \mathbf{az} & \mathbf{Pz} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{matrix}$$

$$x = C1.C2 - S1.S2$$

$$nx = -C1.S2 - C2.S1$$

$$ax = ay = 0$$

$$Sy = C1.S2 + C2.S1$$

$$ny = C1.C2 - S1.S2$$

$$Sz = C1.C2 - S1.S2$$

$$nz = -C1.S2 - C2.S1$$

$$az = 1$$

Donc :  $2 P_x = l_1 + \cos(\theta) \cdot l_1$  et  $2 P_y \sin(\theta) \cdot l_1$

### Modèle géométrique inverse :

Il existe plusieurs méthodes et procédures numériques pour le calcul du MGI, nous intéressons à la méthode de Paul [5]. La méthode de Paul consiste à prémultiplier, successivement, les deux membres de l'équation par  $jT^{j-1}$  ( $j=1, \dots, n-1$ ) ce qui permet de calculer, successivement, les variables  $q_j$ . Nous résoudrons :

$$U_0 = {}^0_1T(q_1) \cdot {}^1_2T(q_2)$$

on trouve :

$$\theta_2 = \alpha \cos\left(\frac{(x^2 + y^2 - l_1^2 - l_2^2)}{(2 \cdot l_1 \cdot l_2)}\right)$$

$$beta = \alpha \cos\left(\frac{(x^2 + y^2 - l_1^2 - l_2^2)}{2 \cdot l_1 \cdot \sqrt{x^2 + y^2}}\right) \quad alpha = atan(y, x)$$

$$\theta_1 = alpha - beta$$

### Modèle cinématique direct :

Le modèle cinématique d'un robot se base sur la matrice jacobienne donc on suit les étapes suivantes : Le robot à 2 ddl de la Figure 2.5. La position de l'effecteur ( $a_x, b_y$ ) dans le plan ( $x, y$ ) est exprimée en fonction des variables articulaires ( $\theta_1, \theta_2$ ) par :

$$a_x = l_1 \cdot \cos(\theta_1) + l_2 \cdot \cos(\theta_1 + \theta_2)$$

$$b_y = l_1 \cdot \sin(\theta_1) + l_2 \cdot \sin(\theta_1 + \theta_2)$$

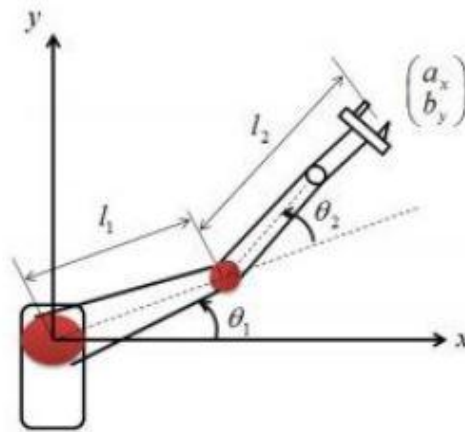


Fig2.8 : Robot SCARA 2ddl Rotorique

Sous l'hypothèse que l'effecteur du robot se déplace par de petits accroissements, les dérivées partielles des coordonnées opérationnelles par rapport aux variables des articulations sont exprimées par :

$$ax = \frac{\partial ax(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 + \frac{\partial ax(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2$$

$$by = \frac{\partial by(\theta_1, \theta_2)}{\partial \theta_1} d\theta_1 + \frac{\partial by(\theta_1, \theta_2)}{\partial \theta_2} d\theta_2$$

Ou sous la forme matricielle suivante :

$$\begin{bmatrix} ax \\ by \end{bmatrix} = \begin{bmatrix} \frac{\partial ax(\theta_1, \theta_2)}{\partial \theta_1} & \frac{\partial ax(\theta_1, \theta_2)}{\partial \theta_2} \\ \frac{\partial by(\theta_1, \theta_2)}{\partial \theta_1} & \frac{\partial by(\theta_1, \theta_2)}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix}$$

On peut l'écrire sous la forme suivante :  $dP = J q \cdot dq$

Avec :

$$dP = [dax, dby]^T, dq = [d\theta_1, d\theta_2]^T$$

$J(q)$  est une matrice appelée jacobienne.

Donc la matrice jacobienne de notre robot est :

$$J = \begin{pmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ \cos(\theta_1) - l_2 \cos(\theta_1 + \theta_2) & -l_2 \cos(\theta_1 + \theta_2) \end{pmatrix}$$

## Génération de trajectoire :

Pour le déplacement d'un robot, on doit d'abord déterminer la trajectoire à suivre d'un point A vers un point B, pour cela, on doit prendre en considération les 2 contraintes suivantes : - Eventuellement, le passage par une position intermédiaire (C) - Certaines contraintes : temporelles, cinématique, dynamique, ...

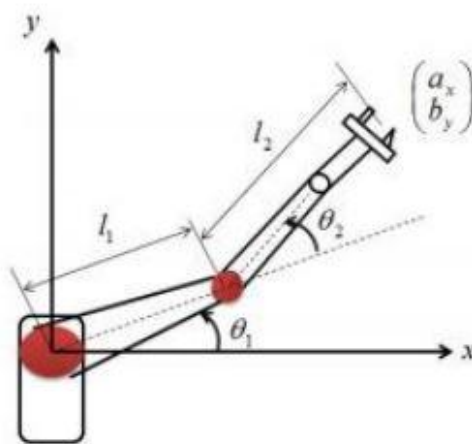


Fig 2.6 : trajectoire Robot à 2ddl

On peut distinguer 4 grandes classes de mouvement :

1. Le mouvement entre deux points avec trajectoire libre entre les points ;
2. Le mouvement entre deux points via des points intermédiaires, spécifiés notamment pour éviter les obstacles, avec trajectoire libre entre les points intermédiaires ;
3. Le mouvement entre deux points avec trajectoire contrainte entre les points (trajectoire rectiligne par exemple) ;
4. Le mouvement entre deux points via des points intermédiaires avec trajectoire contrainte entre les points intermédiaire

En génération de trajectoire on a 2 espaces :

- **L'espace articulaire :**

Est l'espace qui a pour référence le repère lié à chaque articulation motorisée du robot. Les coordonnées associées sont appelées coordonnées articulaires.

- **L'espace opérationnel :**

Est l'espace qui a pour référence le repère lié à l'organe terminal du robot. Les coordonnées associées sont appelées coordonnées opérationnelles.

La génération de mouvement peut se faire soit dans l'espace articulaire (mouvement 1 et 2) ou dans l'espace opérationnel (mouvement 3 et 4). Le fondement mathématique de la génération de mouvement est basé sur le calcul d'une ou plusieurs fonctions d'interpolation qui construisent l'équation du mouvement à partir de contraintes spatiales et temporelles.

Génération de mouvement dans l'espace Articulaire	Génération de mouvement dans l'espace Opérationnel
<ul style="list-style-type: none"> <li>- le mouvement est minimal sur chaque articulation.</li> <li>- elle nécessite moins de calcul en ligne</li> <li>- le mouvement n'est pas affecté par le passage sur les configurations singulières.</li> <li>- les contraintes de vitesse et de couples Maximaux sont connues avec précision Puisqu'elles correspondent aux limites physiques des actionneurs.</li> <li>-la géométrie de la trajectoire dans l'espace Opérationnel ne peut être imposée</li> </ul>	<ul style="list-style-type: none"> <li>-contrôler la géométrie de la trajectoire</li> <li>- elle implique la transformation en coordonnées articulaires de chaque point de la trajectoire,</li> <li>- elle peut être mise en échec lorsque la Trajectoire calculée passe par une position singulière,</li> <li>- elle peut être mise en échec chaque fois que les points de la trajectoire engendrée ne sont pas dans le volume accessible du robot</li> </ul>

### Génération de trajectoire dans l'espace articulaire

Le principe général de génération du mouvement dans l'espace articulaire est présenté par la boucle d'asservissement de la Figure 2.7. Soit  $T_d$  et  $T_f$  deux matrices de transformations homogènes associées respectivement à la configuration initiale (ou de départ) et finale de 31 l'organe terminal. En utilisant le MGD du robot, les vecteurs des coordonnées articulaires correspondant aux configurations initiale  $q_i$  et finale sont calculées.

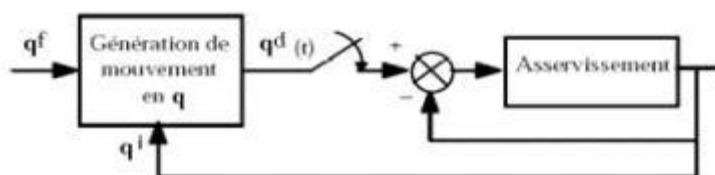


Fig2.7 : génération de mouvement dans l'espace articulaire

Pour satisfaire le passage par  $q_i$  à  $t = 0$  et  $q_f$  à  $t = t_f$  on utilise la méthode d'interpolation polynomial [5].

### Génération de trajectoire dans l'espace opérationnelle :

La génération d trajectoire dans l'espace opérationnelle c'est donnée la position désirée et à l'aide du MGI qui calcule les Thêta désirer de chaque articulation du robot qui donne le Thêta mesurer et en utilisant le MGD qui calcule la position mesurer qui donne la position finale qui doit le robot attendre .

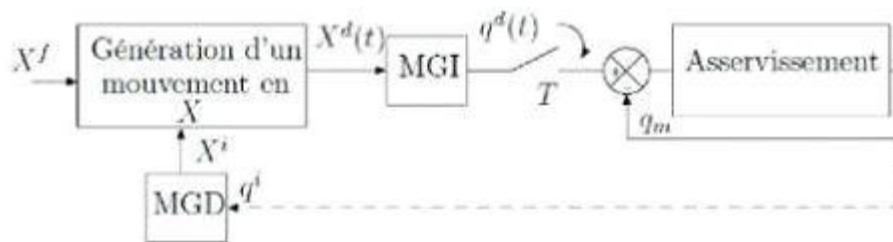


Fig 2.8 générations mouvement dans l'espace opérationnelle

### Méthode d'interpolation polynomiale :

#### Polynôme 3ème degré :

Le but de cette méthode est de trouvé les coefficients du polynôme en fonction de t (il y'as de 3ème degré et de 5ème degré) Supposons que le mouvement de la trajectoire de la consigne q en fonction du temps T est décrit par suivant le polynôme de 3ème degré suivant :

$$q(t) = C_0 + C_1t + C_2t^2 + C_3t^3$$

On la dérive en 1er ordre on trouve :

$$\dot{q}(t) = C_1 + 2C_2t + 3C_3t^2$$

On prend par considération des contrainte suivante :

$$q(0) = q_i$$

$$q(t_f) = q_f$$

$$\dot{q}(0) = 0$$

$$\dot{q}(t_f) = 0$$

2.13 et 2.15 contraintes représentent des contraintes associées à la position initiale et finale de l'organe terminal et 2.15 2.14 des contrainte sont des contraintes associées à la vitesse. Dans ce cas, les vitesses initiales et finales sont considérées nulles. Substituons les 3.3 et 3.6 dans le polynôme de 3.1 et sa dérivée 3.2, nous obtenons :

$$q(0) = C_0 = q_i$$

$$q(t_f) = C_0 + C_1 t_f + C_2 t_f^2 + C_3 t_f^3$$

$$\dot{q}(0) = C_1 = 0$$

$$\dot{q}(t_f) = C_1 + 2C_2 t_f + 3C_3 t_f^2 + C_3 t_f^3 = 0$$

La résolution de ces équations pour  $i$  tel que  $i = 1 \dots 4$  est :

$$C_0 = q_i$$

$$C_1 = 0$$

$$C_2 = \frac{3}{td^2}$$

$$C_3 = \frac{2}{td^3} (q_f - q_i)$$

En utilisant l'ensemble des équations ci-dessus, on peut calculer le polynôme qui permet de relier la position initiale à la position désirée pour des vitesses initiales et finales nulles.

### Polynôme 5ème degré :

Une interpolation polynomiale de degré cinq (2.25) est utilisée dans le cas où nous voulons spécifier la position, la vitesse et l'accélération au début et à la fin du mouvement. Ceci permettra d'assurer la continuité de l'accélération et aussi pour éviter le phénomène de résonance dans la SMA.

$$q(t) = C_0 + C_1 t + C_2 t^2 + C_3 t^3 + C_4 t^4 + C_5 t^5 +$$

Pour cela, un ensemble de six contraintes exprimées par les équations ci-dessous est à spécifier :

$$q(0) = q_0$$

$$q(t_f) = q_f$$



$$q \cdot (0) = q \cdot_0$$

$$q(t_f) = q \cdot_f$$

$$q \cdot \cdot (0) = q \cdot \cdot_0$$

$$q \cdot \cdot (t_f) = q \cdot \cdot_f$$

C'est à dire, un ensemble de six équations à six inconnues à résoudre :

$$q(0) = C0 = qi$$

$$q(t_f) = C0 = C_1 t_f + C_2 t_f^2 + C_3 t_f^3 + C_4 t_f^4 + C_5 t_f^5 = qf$$

$$q \cdot (0) = C1 = q \cdot_i$$

$$q \cdot (t_f) = C1 + 2C_2 t_f + 3C_3 t_f^2 + 4C_4 t_f^3 + 5C_5 t_f^4 = q \cdot f$$

$$q \cdot \cdot (0) = 2C2 = q \cdot \cdot_0$$

$$q \cdot \cdot (t_f) = 2C2 + 6C_3 t_f + 12C_4 t_f^2 + 20C_5 t_f^3 = q \cdot \cdot_f$$

## La régulation de trajectoire :

En robotique il y'as plusieurs méthodes de régulation de trajectoire (adaptative, PID, découplage linéaire...etc)

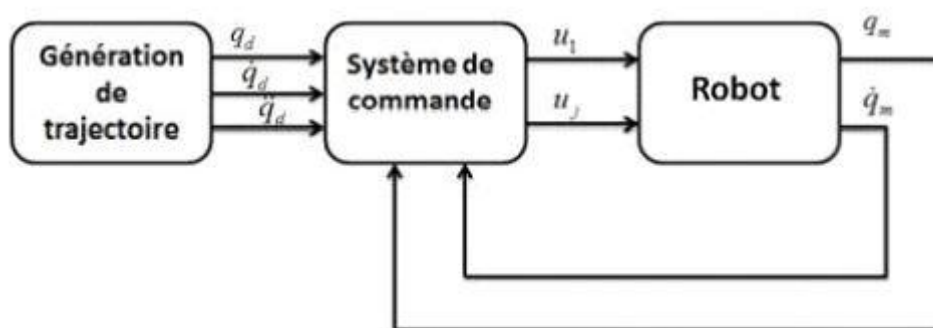


Fig2.9 : Schéma bloc de la régulation

## Régulation PID

La régulation utilisée dans ce projet est une régulation (Proportionnelle, Intégrale et Dérivée).

## L'espace opérationnelle :

L'équation 2.28 c'est la commande sur l'espace opérationnelle

$$T = J^T [K_p (x^d - x) + K_i \int_{t=0}^t (x^d - x) dt + K_d (x^d - x)]$$

$K_p$ ,  $K_d$  et  $K_i$  sont des matrices diagonales définies positives, de dimension  $[n, n]$ , dont les éléments génériques sont respectivement les gains des actions proportionnelles  $K_{pj}$ , dérivées  $K_{dj}$  et intégrales  $K_{ij}$ .  $X$  est le vecteur des coordonnées opérationnelles et  $J^T$  est le Jacobien du robot .

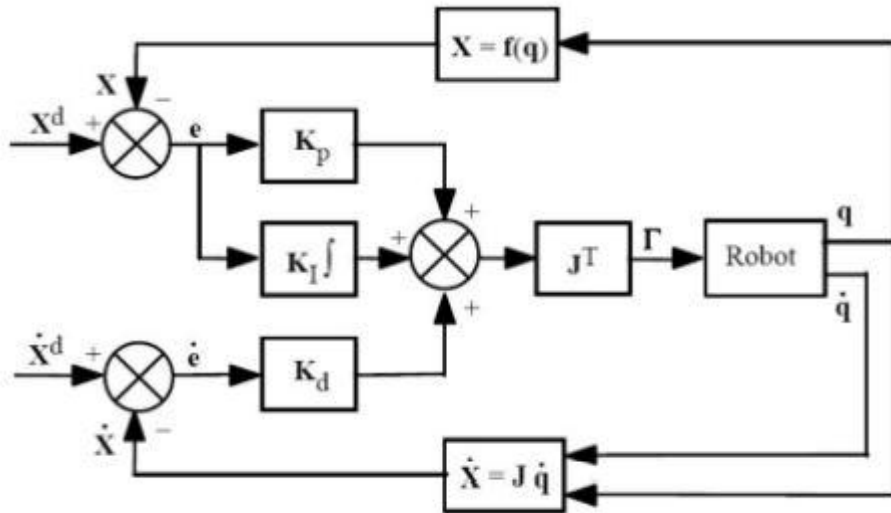


Fig2.10 : schéma bloc de la régulation PID dans l'espace opérationnelle

### Espace articulaire :

L'équation 2.29 c'est la commande dans l'espace articulaire :

$$T = K_p (q^d - q) + K_i \int_{t=0}^t (q^d - q) dt + K_d (\dot{q}^d - \dot{q})$$

Où  $q^d(t)$  et  $\dot{q}^d(t)$  désignent les position et vitesse désirées dans l'espace articulaire. Et  $K_p$ ,  $K_i$ , et  $K_d$  sont respectivement les gains des actions proportionnelle, intégrale et dérivé [5]

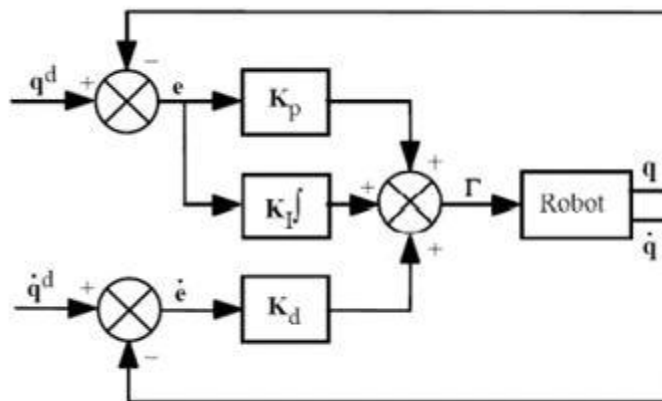


Fig2.11 schémas bloc de la régulation dans l'espace articulaire

### **Conclusion :**

Dans ce chapitre, nous avons présenté l'essentielle de la modélisation utilisée en robotique, la génération de trajectoire, ainsi la régulation PID dans les deux espaces (articulaire et opérationnel). Dans le prochain chapitre, les résultats de simulation sont présentés.

# **Chapitre III :**

## **Résultats de simulation**

## Introduction :

Après la modélisation géométrique et cinématique du robot, notre robot est simulé à l'aide des logiciels informatiques. La génération de trajectoire est basée sur la méthode de l'interpolation polynomial.

## La co-simulation par Simscape / Solidworks :

Solidworks est le logiciel le plus utilisé dans le Dessin Assisté par Ordinateur (DAO). Il a la capacité de présenter une structure mécanique articulée en 3D segment par segment et les rassembler un par un à l'aide des contraintes que ce soit coaxiale ou coïncidente ou autres. A la fin, nous obtenons une structure articulée, dans notre cas un robot. Pour modéliser la dynamique du robot et la simuler, l'assemblage réalisé sur la plateforme Solidworks est exporté vers Matlab dans l'environnement Simlink, dans laquelle, tous les modèles à savoir géométrique, cinématique et dynamique sont implémentés. L'outil choisi pour simuler le SCARA à 2ddl est Simmecahnics, qui est un sous-outil de Simulink®. Par conséquent, ses modèles peuvent être interfacés avec des schémas fonctionnels ordinaires de Simulink qui accélèrent la simulation et tout intégrer dans le même environnement. De plus, il est simple à utiliser et son ensemble de blocs se compose de sept sousbibliothèques qui représentent le corps, articulations, capteurs (capteurs articulaires, capteurs corporels), actionneurs (actionneurs articulaires, actionneurs), boîtes de vitesses, contraintes et entraîneurs, et éléments de force. L'outil Simmecahnics permet de :

- Modéliser tous les éléments d'un système multi-corps (corps, articulations, connexions, forces) dans Simulink.
- Importer des modèles complets à partir de systèmes de CAO (c'est-à-dire Stäubli SolidWorks CAD), avec la propriété d'inertie, longueurs, angles.
- Générez une animation 3D pour visualiser la dynamique du système [8].

## La bibliothèque de Simmecahnics :

Dans la bibliothèque de Simmecahnics on trouve 7 familles de fonctions que nous utilisons pour modéliser n'importe quelle structure mécanique articulée (dans notre cas c'est un robot) :



Fig3.1 Les fonctions de simmechanics First génération

Dans cette bibliothèque on s'intéresse à 2 sous-fonctions essentielles.

**Bodies (Corp) :**

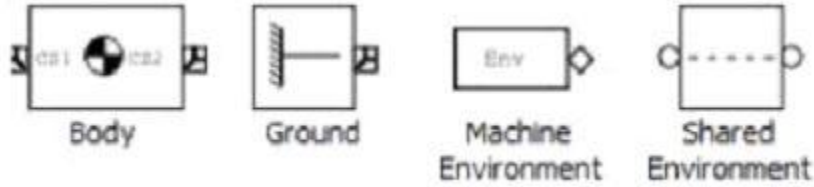


Fig3.2 sous bibliothèque corps

Cette bibliothèque nous aide à représenter le corps d'un robot on définit les propriétés suivantes : la masse, la matrice d'inertie, les coordonnées du centre de gravité

**Les articulations :**

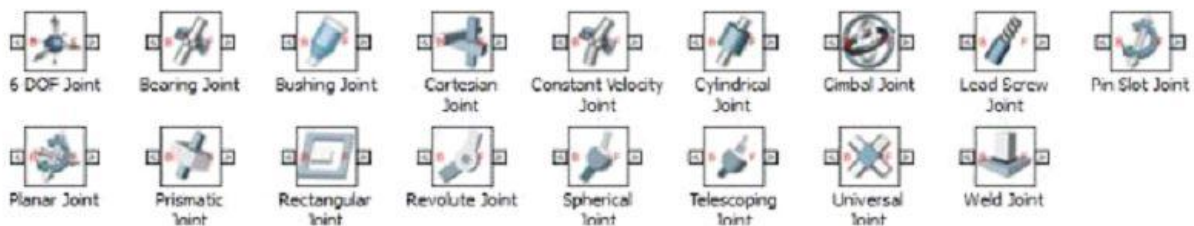


Fig3.3 : sous bibliothèque articulations

Les blocs de données de cette bibliothèque servent à donner des degrés de liberté aux articulations. Les articulations du SCARA sont toutes rotatives, le bloc correspondant c'est « REVOLTE JOINT »

**Assemblage de SCARA :**

Sur le site web j'ai téléchargé le fichier qui contient les pièces de notre robot (en totale on a 4 pièces)

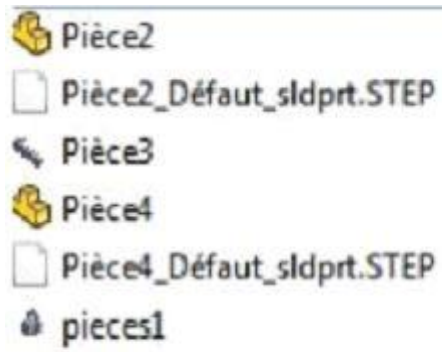


Fig3.4 les pièces de SCARA non assembler

Dans logiciel SolidWorks on fait le l'assemblage des pièces un par un et entre une pièce et l'autre on donne la contrainte :

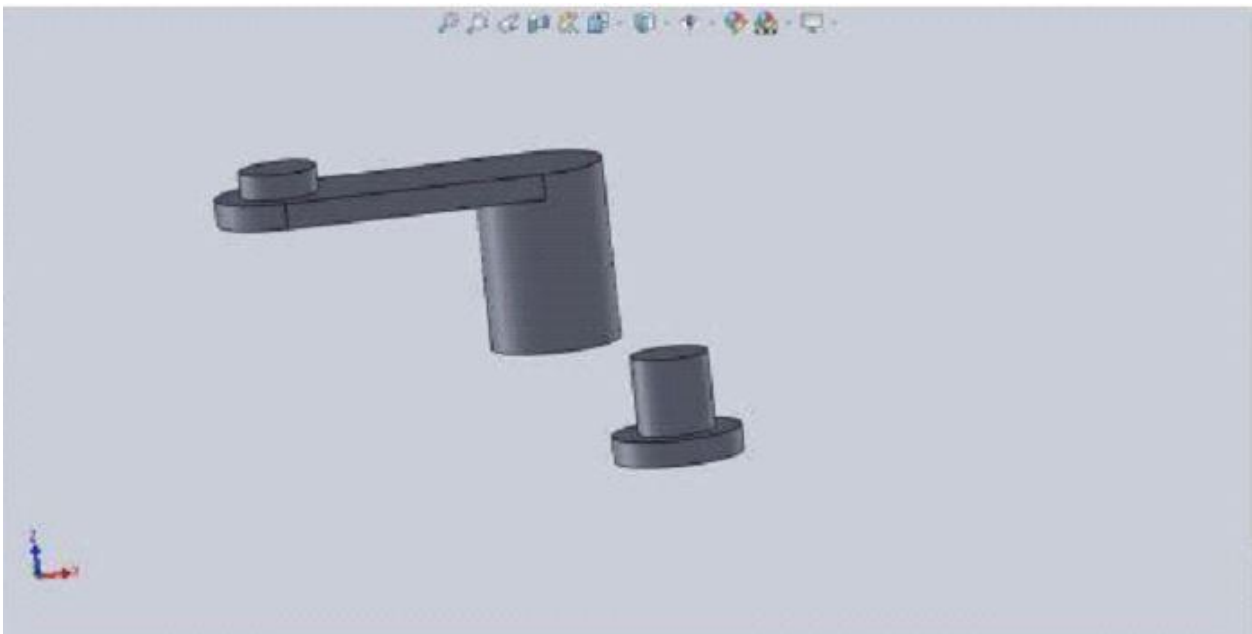


Fig3.5 Les 2 premiers pièces de SCARA

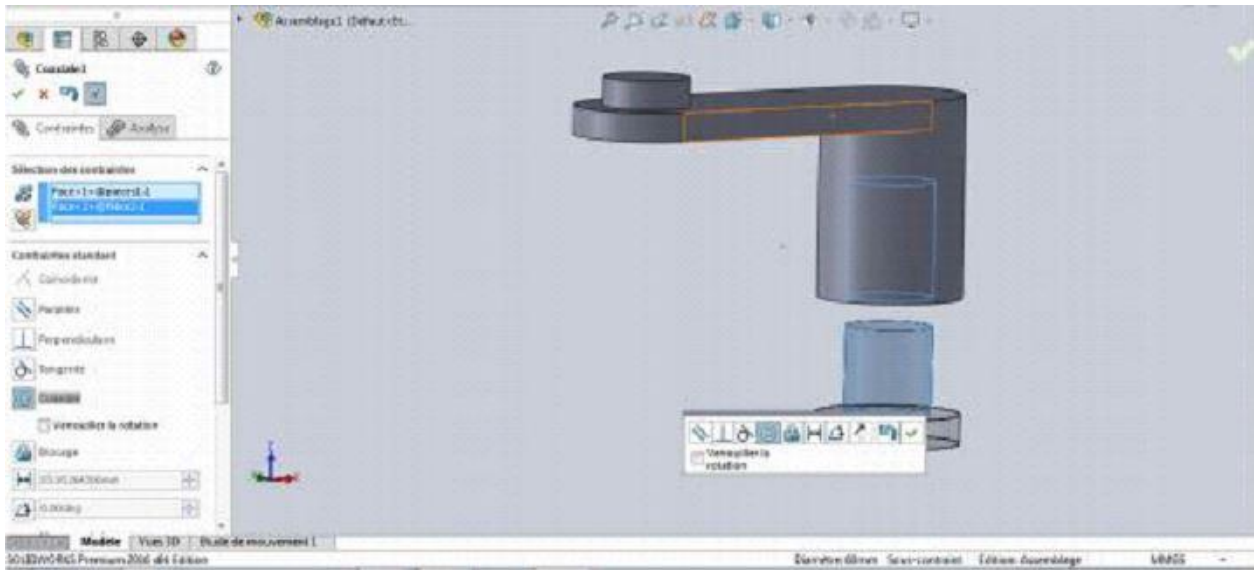


Fig3.6 : l'assemblage et la création des contraintes

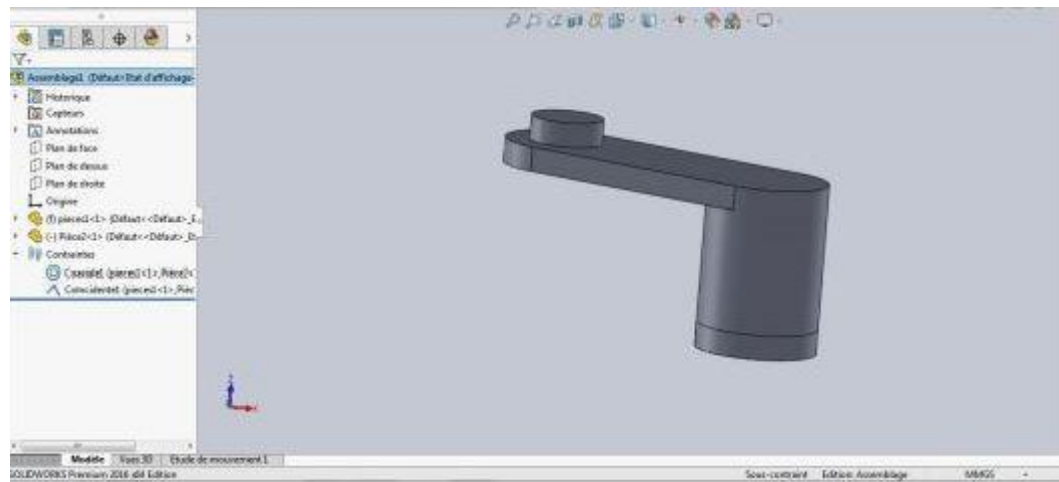


Fig3.7 : Les contraintes des pièces

Les mêmes étapes pour les autres pièces (1avec2 et 2avec 3...) jusqu'on obtient le modèle du robot complet sous forme 3D.



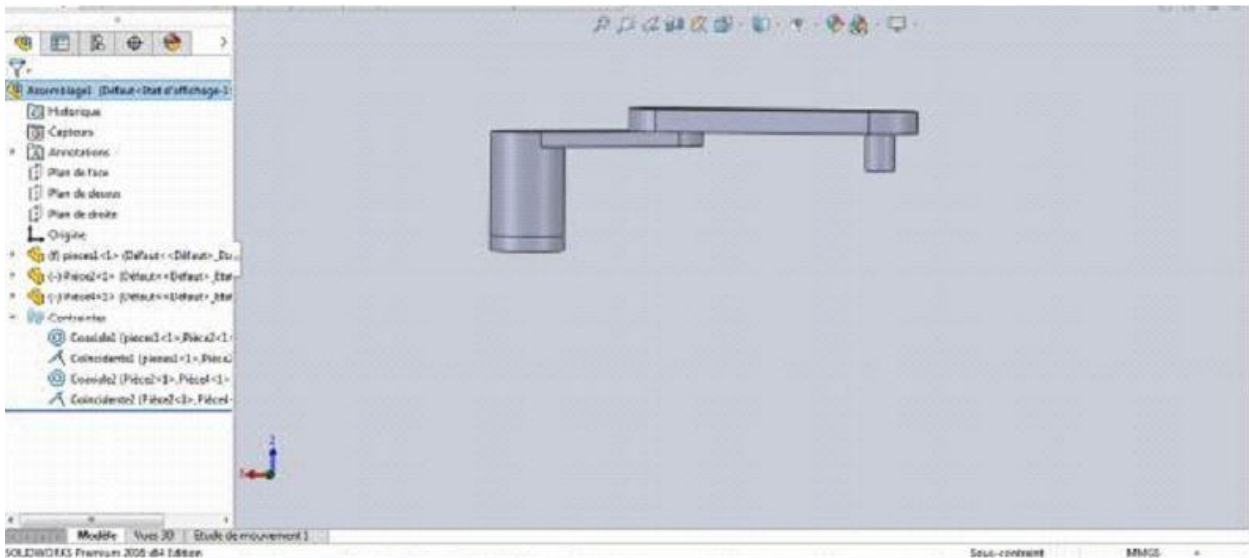


Fig3.8 : le modèle complet de SCARA sous SolidWorks

### Procédure d'importation d'un modèle CAO dans l'environnement Simulink :

-Tous d'abord il faut lier SolidWorks avec Matlab en suivant les instructions suivantes (il faut que les fichiers d'installation soient dans le répertoire du Matlab) :

```
>>install_addon('smlink.r2018a.win64.zip')
```

```
>>regmatlabserver
```

```
>>smlink_linksw
```

-dans le logiciel SolidWorks et après avoir assemblé le robot, on l'exporte en suivant les étapes suivantes :

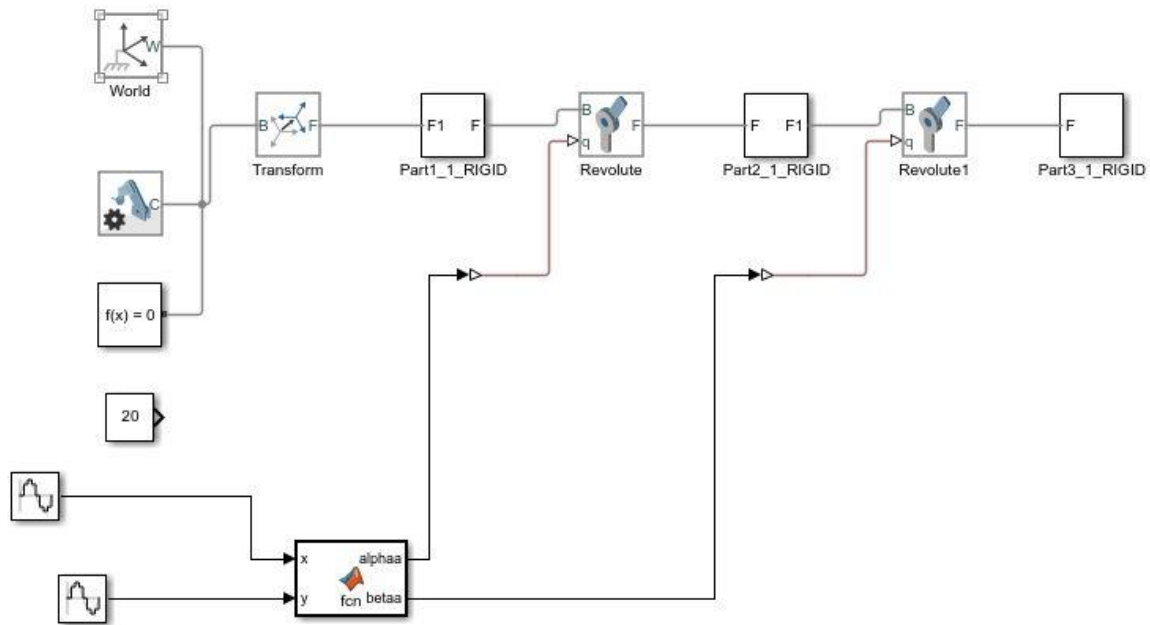
Outil>simscapemultibodylink>Exporte>simscapemultibody First generation (1G) ou

Outil>simscapemultibodylink>Exporte>simscapemultibody

-en Matlab en met en répertoire le dossier du robot en suivant les instructions suivantes :

```
>>mech_import('robot_scara.xml') (si c'est un robot de 1er génération)
```

```
>>smimport('robot_scara.xml') ; (si c'est un robot de 2ème génération)
```



- Pièce 1\_1\_RIGID, Pièce 2\_1\_RIGID, Pièce 4\_1\_RIGID : les pièces du robot
- Revolte, Revolte1 : les articulations Rotoriques du robot
- PS-Simulink Converter : convertisseur d'un signal numérique vers une valeur physique
- T1,T2 : Thêta désirées, et sont les entrées du robot
- theta1, theta2 : Thêta mesurées et sont les sorties du robot

## Simulation et résultat

Après avoir le modèle Simulink du robot (fig3.10) on intègre le MGD, MGI, pour générer les mouvements du robot.

-MGD : c'est la modélisation géométrique direct de notre robot représenté sous forme une fonction Matlab, avec (Theta1, Tehta2 Entrer et x2, y2 sortie)

```

x2 = x1 + l2*cos(theta1+theta2);
y2 = y1 + l2*sin(theta1+theta2);

PositionEffecteur = [x2 y2] ;

function PositionEffecteur = ModeleGeometriqueDirect (Theta)
global x1 y1 x2 y2

global theta1 theta2
global theta1_old theta2_old

l1 = 0.2;
l2 = 0.3;

theta1 = Theta (1);
theta2 = Theta (2);

theta1_old = theta1;
theta2_old = theta2;

x1 = l1*cos(theta1);
y1 = l1*sin(theta1);

```

-**MGI** : c'est le modèle géométrique inverse de notre robot robot représenté sous forme une fonction Matlab, avec (x, y entrer /theta1 theta2 sortie)

```

function Theta = ModeleGeometriqueInverse(Position)
%global l1 l2
global thetal theta2

x = Position(1);
y = Position(2);

l1= 0.2;
l2= 0.3;

%Les consignes x et y de l'effecteur doivent impérativement être
%contenues dans "l'espace de travail" du robot
if (((sqrt(x^2+y^2))>(l1+l2)) | (sqrt(x^2+y^2)<(l1-l2)))
    warning (['Attention la consigne (x,y) est en d''hors de l''espace
    attiegnable par le robot']);
end

%Approche géométrique (ad hoc) pour le calcul du MGI
theta2 = acos((x^2+y^2-l1^2-l2^2)/(2*l1*l2));

alpha = atan2(y,x);
beta = acos((x^2+y^2+l1^2-l2^2)/(2*l1*sqrt(x^2+y^2)));

thetal = alpha - beta;

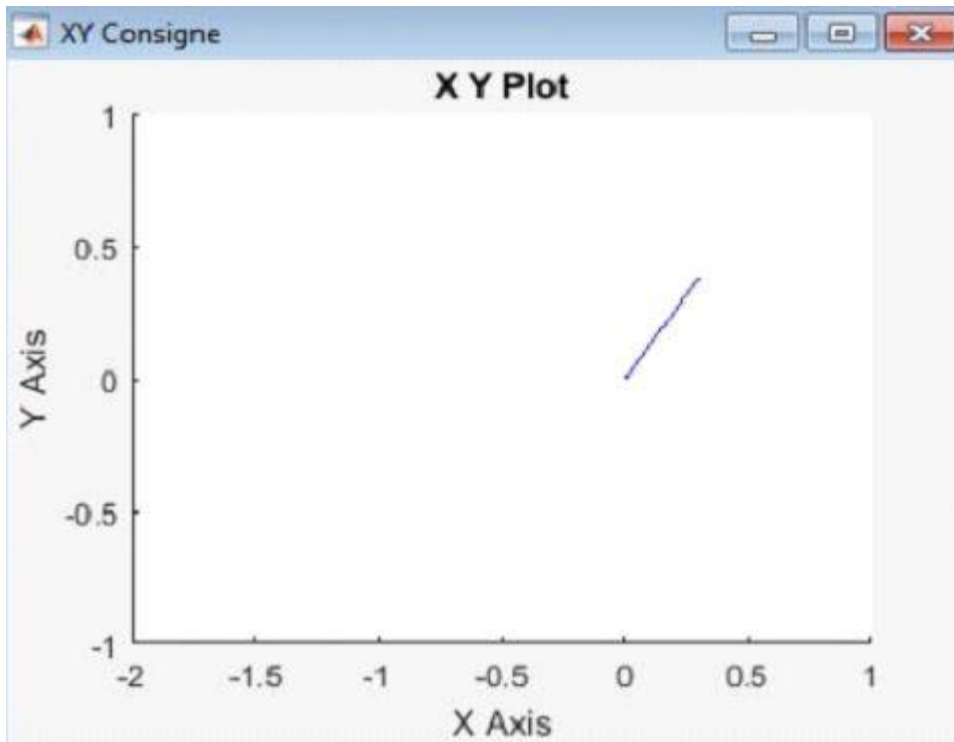
Theta=[real(thetal) real(theta2)];

```

- Trajectory C** : c'est la trajectoire que le ROBOT à suivre.
- XY consigne** : trajectoire en projection sur l'axe X, Y.
- **XY résultant** : sont les résultats de la trajectoire réalisée par le robot
- Err X et Err Y** : c'est pour vérifie les erreurs dans le plot X et Y

Dans cette simulation on a donné a notre robot en espace opérationnelle les coordonnées de  $x_i$   $y_i=0$  vers le point  $x_f=0.38$  et  $y_f=0.3$  (coordonnées initial),  $T=7$  est le temps de simulation

5.1 Trajectoire type rectiligne Pour générer une trajectoire rectiligne la fonction de trajectoire doit être :  $Consigne(1)=x_i+((x_f)/T)*t$ ;  $Consigne(2)=y_i+((y_f-y_i)/T)*t$



3.11 Consigne de trajectoire

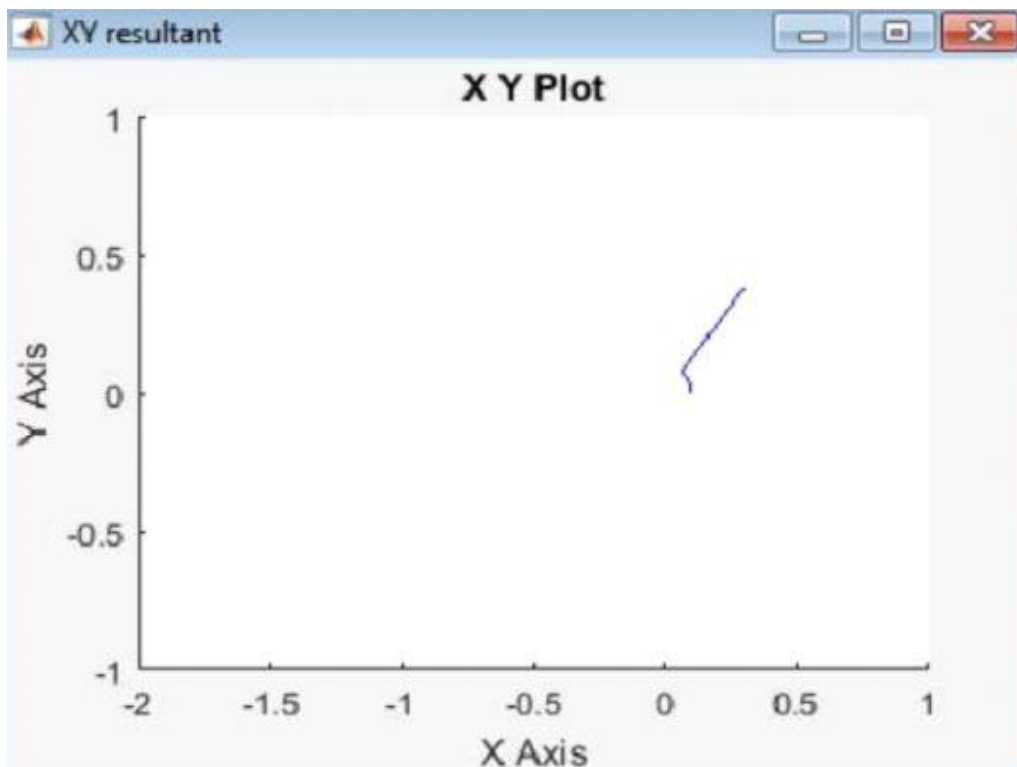


Fig3.12 Résultat de la trajectoire

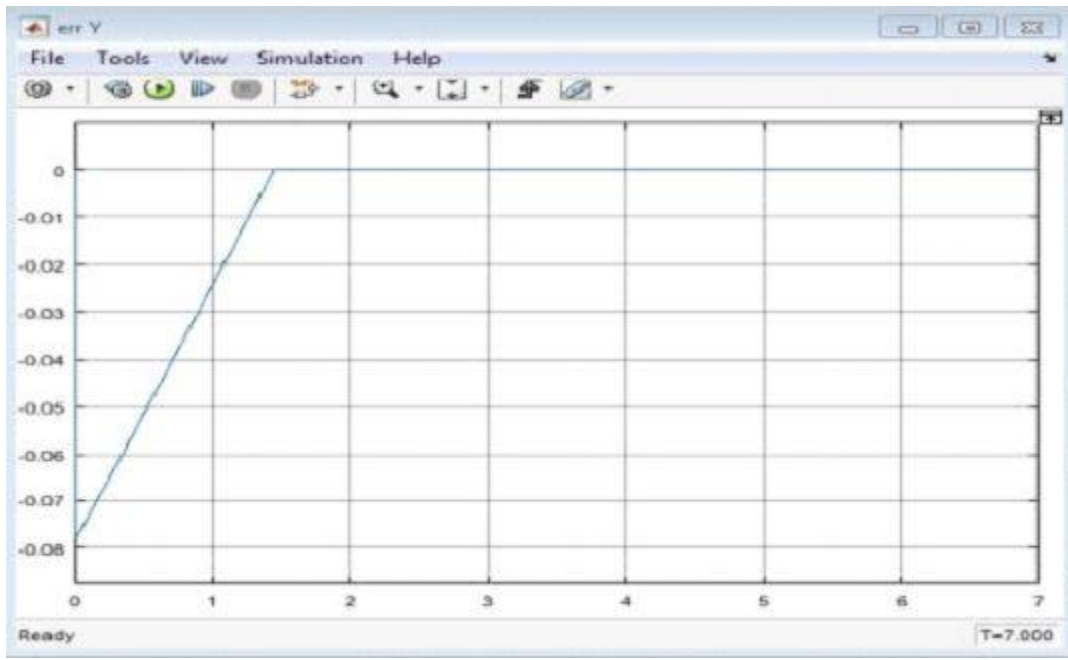


Fig3.13 : Erreur sur Y

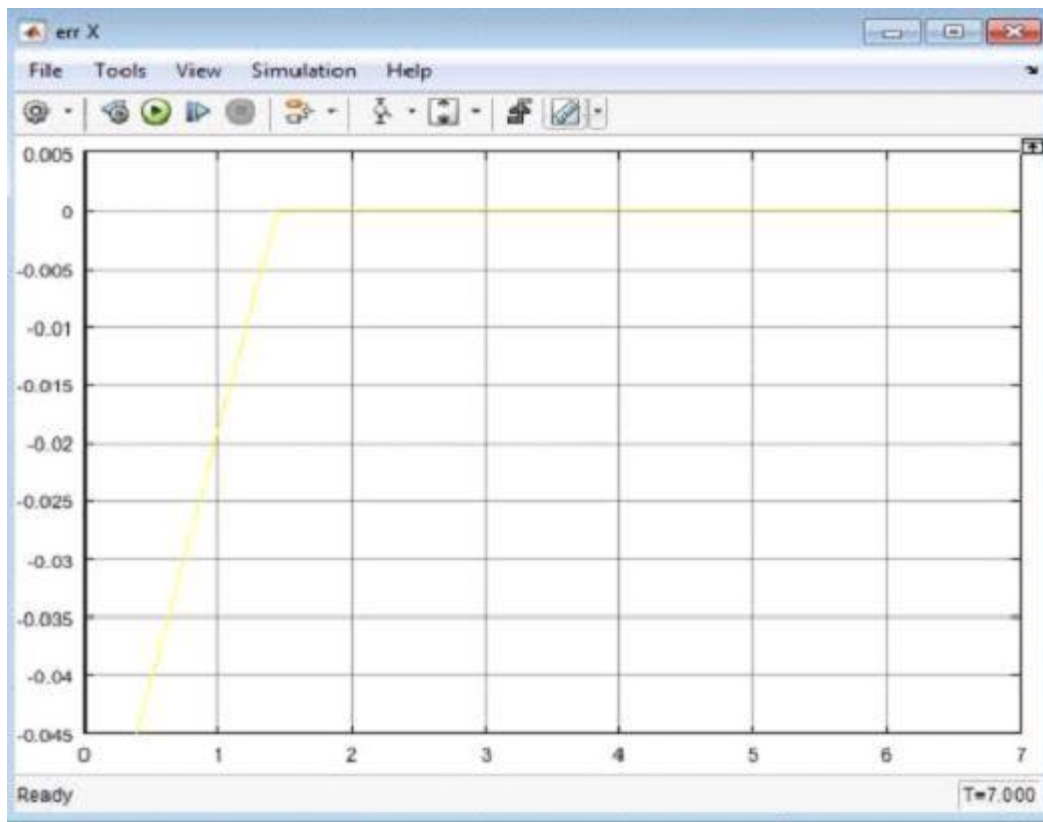


Fig3.14 : Erreur sur X

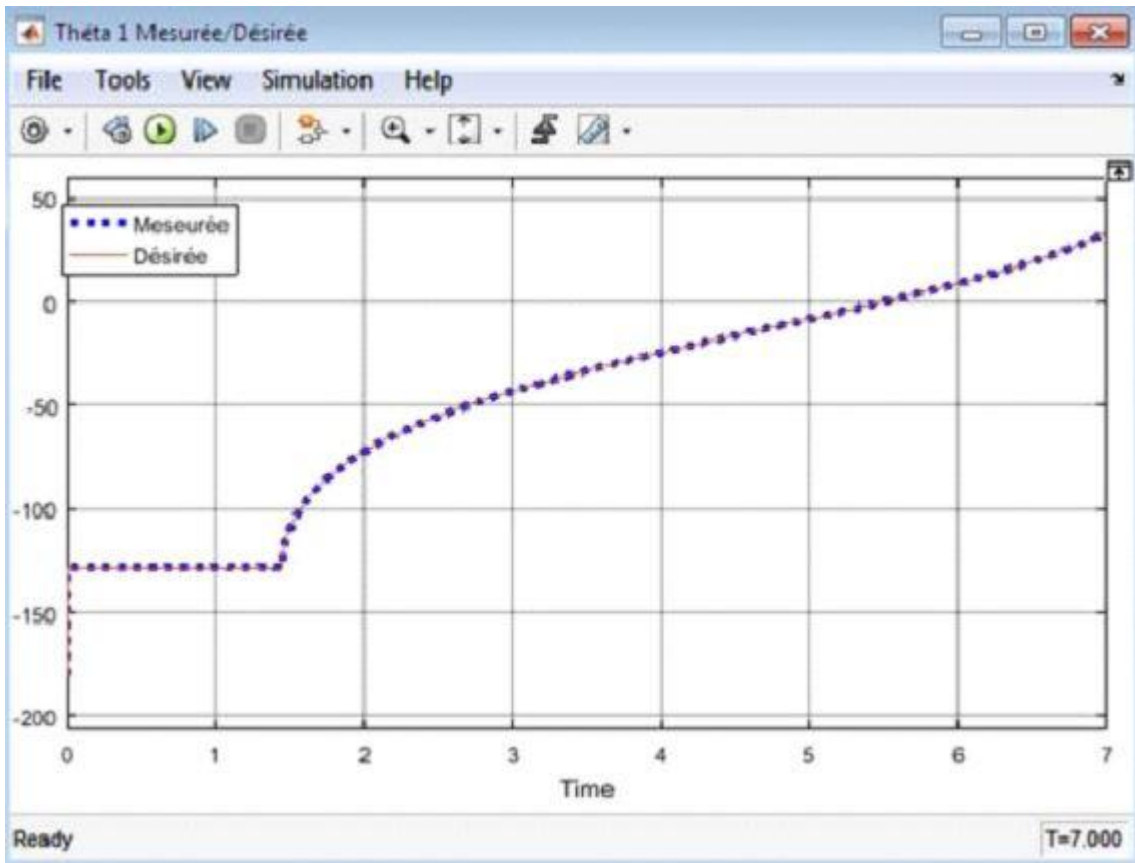


Fig3.15 Thêta 1 désirée et mesurée

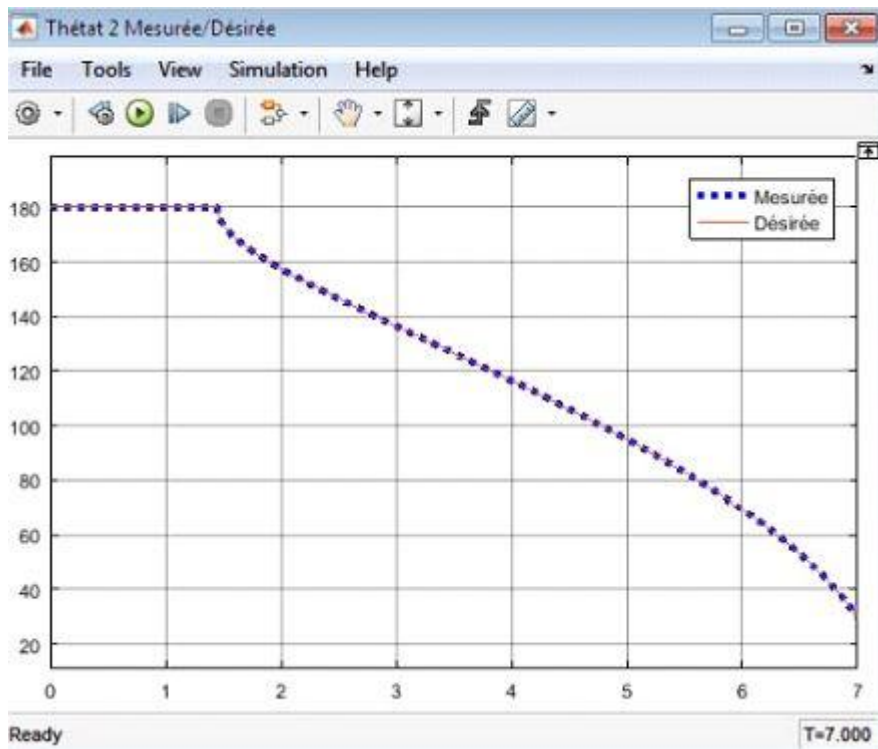


Fig3.16 Thêta 2 désirée et mesurée

### Trajectoire de type polynomiale :

Le polynôme de trajectoire X et Y trouvé a partir de la méthode de l'interpolation polynomial et on as trouvé : $C_0=0$ ,  $C_1=0$ ,  $C_2=0,001$  et  $C_3=-0,007$  pour X et  $C_0=0$ , $C_1=0$ , $C_2=0.01$ et  $C_3=-0.015$  pour Y .

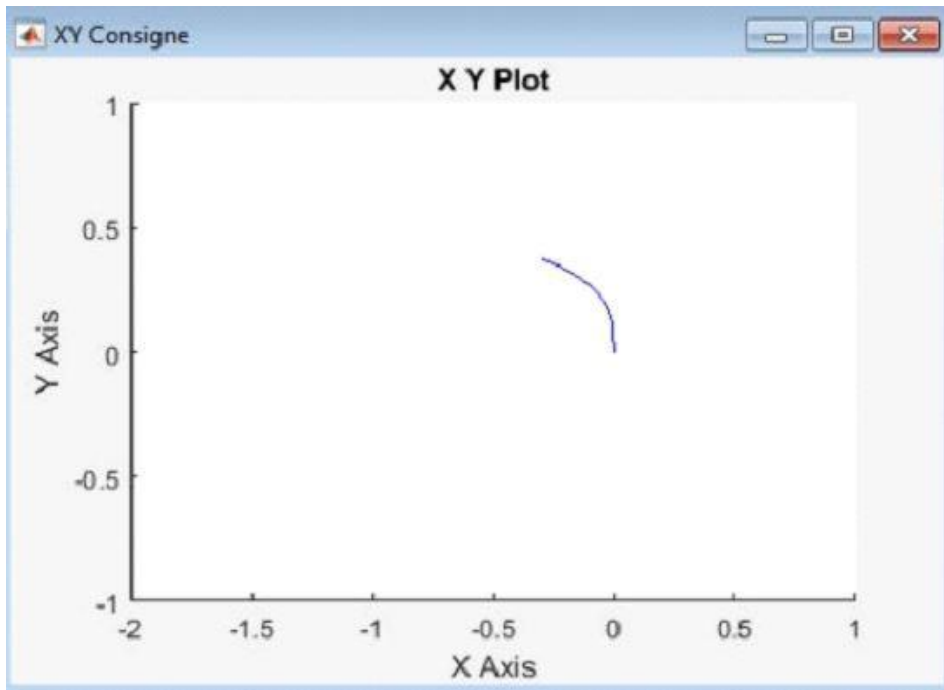


Fig3.17 : Trajectoire consigne

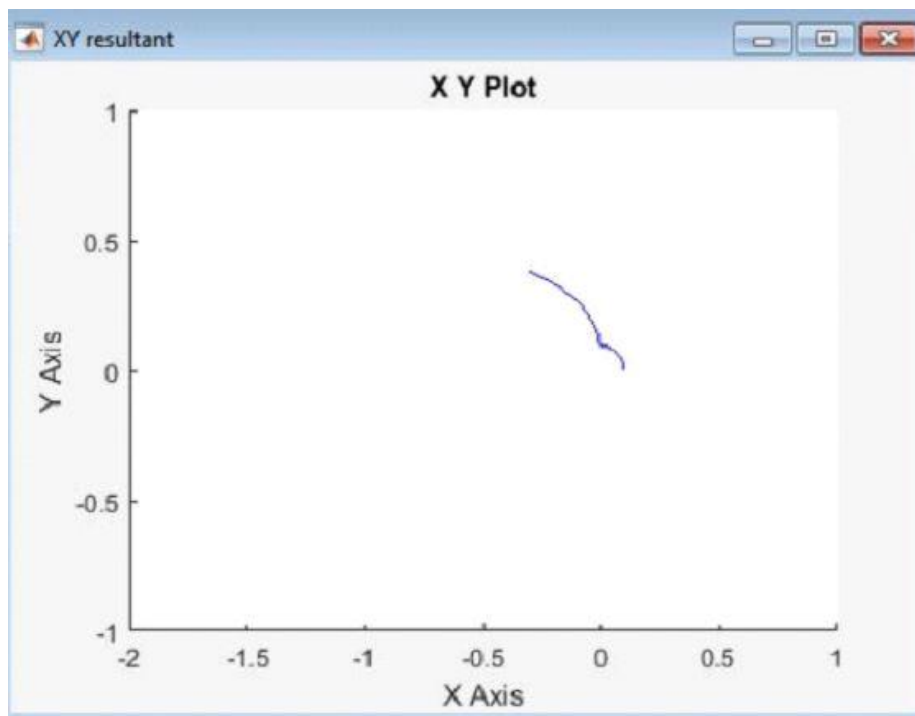


Fig3.18 Trajectoire resultat



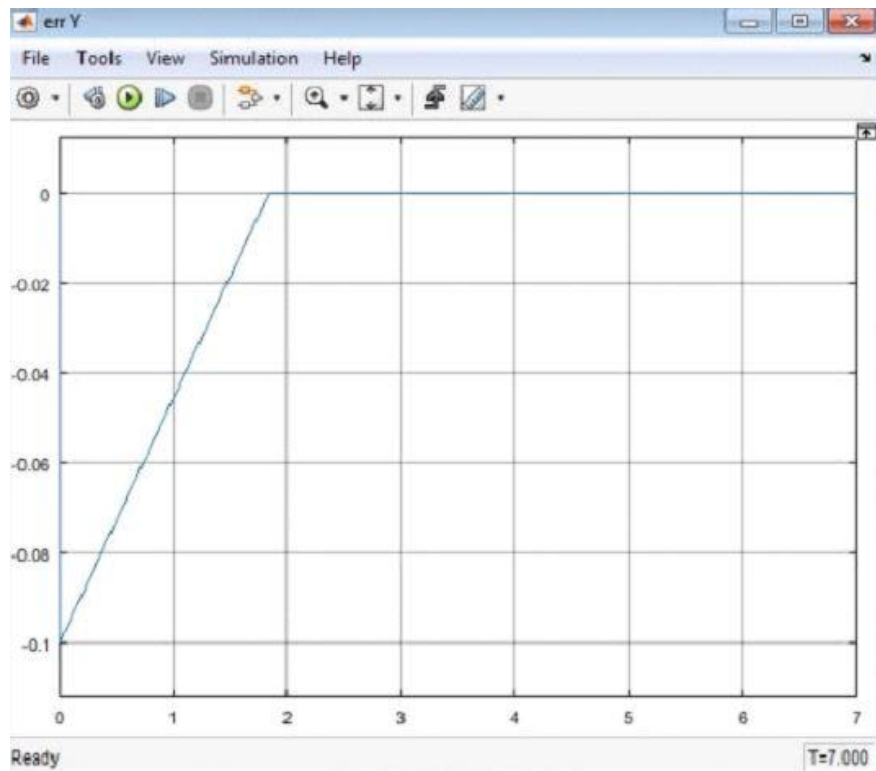


Fig3.19 Erreur sur Y

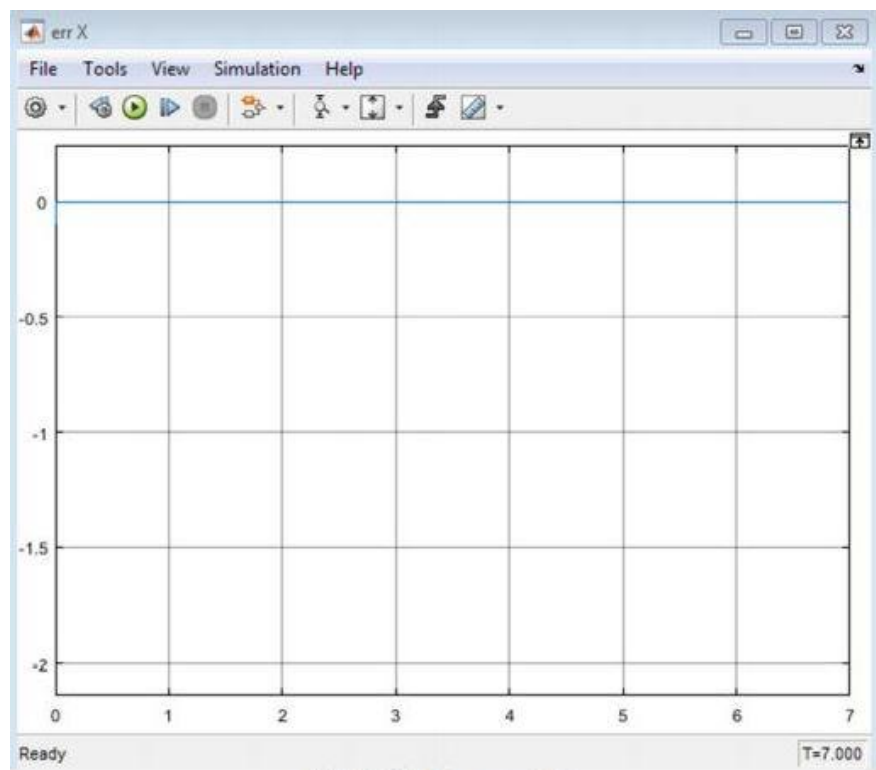


Fig3.20 : Erreur sur X

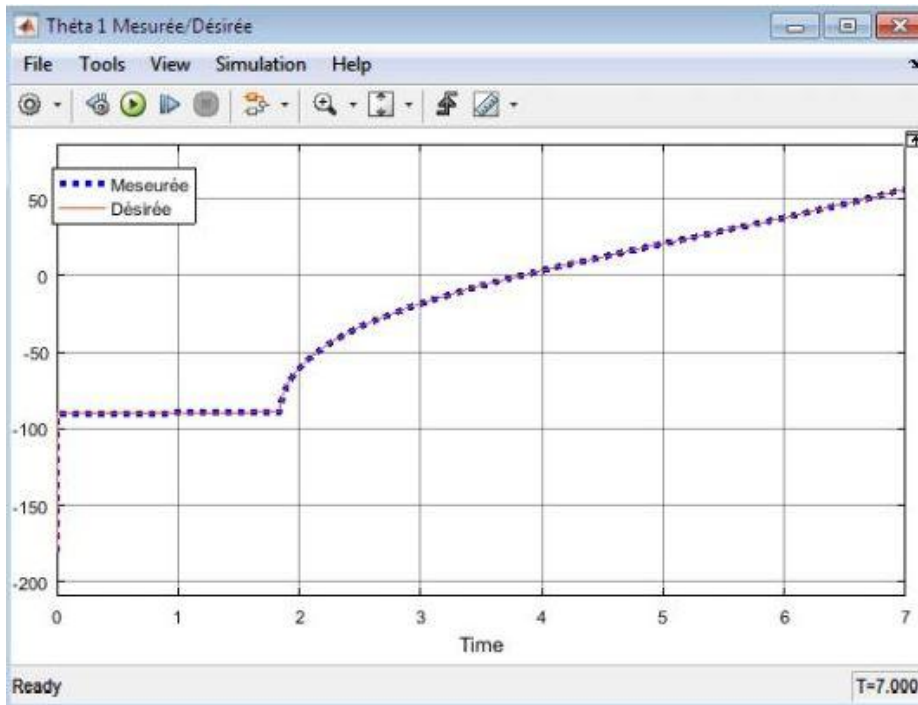


Fig3.21 : Thêta 1 mesurée et désirée

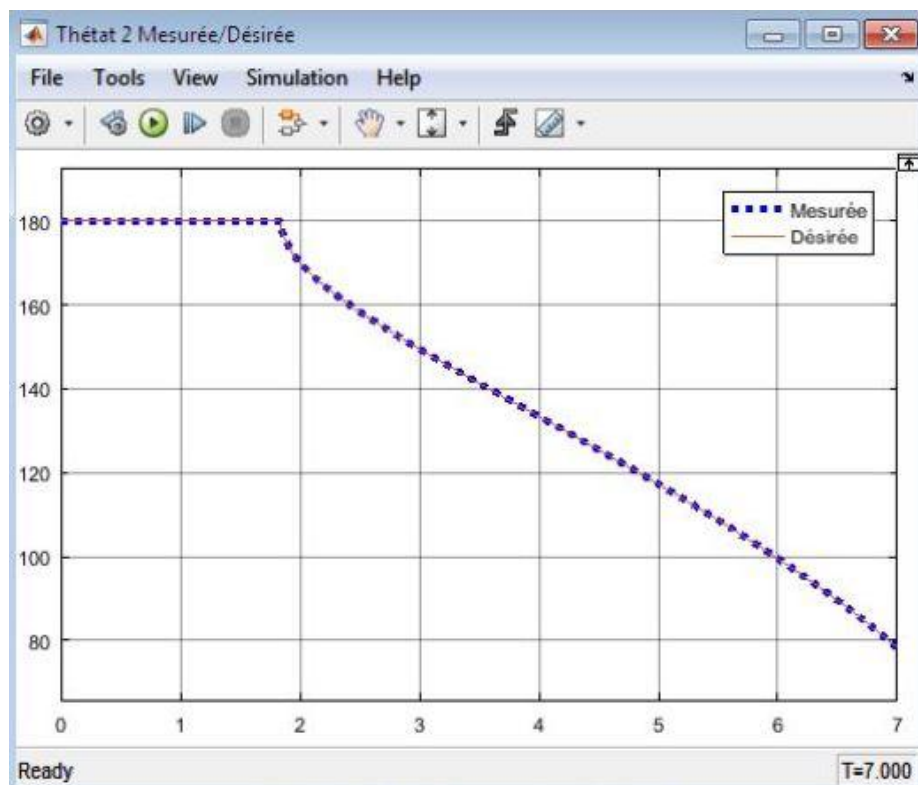


Fig3.22 : Thêta 2 mesurée et désirée

**Conclusion :**

Dans ce chapitre, nous avons présenté les résultats de simulation de notre réalisation. Les résultats obtenus sont satisfaisants.

## **Conclusion générale :**

Dans ce travail, nous nous sommes concentrés sur la modélisation d'un robots industriel type SCARA. Le modèle géométrique direct est obtenu à l'aide de la convention Denavit-Hartenberg, tandis que son modèle est obtenu par la méthode de Paul. L'intégrations de model CAO du solideworks dans la plateforme Simulink/simmechahnics a permet de simuler le robot et intégrer dans une seule plateforme les différents modèles générés.

## **Bibliographie :**

- 1] Alexandre PRÉFUMO Les robots Publié le 26/09/2017
- 2) <http://french.china.org.cn>
- 3) site officiel de staubli (staubli.com)
- 4) brocheur de TP de Promeo - Université de Picardie Jules Verne- 2015/2016
- 5) HibaHage Identification et simulation physique d'un robot Stäubli TX90 pour le fraisage à grande vitesse [6 Jun 2013]
- 6) Hocine TAKHI Université Amar Telidji - Instrumentation 2014
- 7) Philippe Bidaud simulation of a Stäubli TX90 Robot during Milling Using Simmecahnics (Pierre and Marie Curie University - Paris 6)