

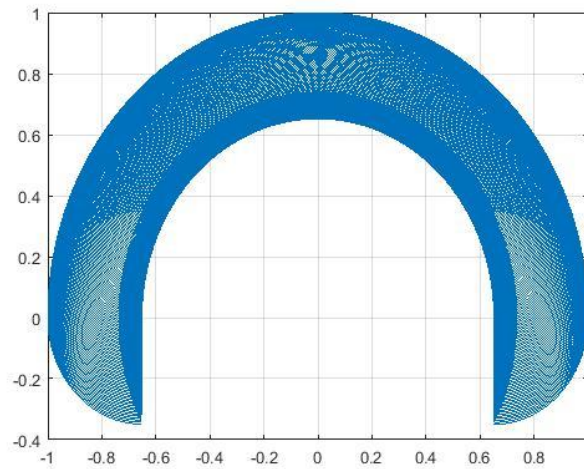
TP2 Robotique de manipulation

Etude du MGD

On a le MGD du bras manipulateur

$$X = \begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{pmatrix}$$

On peut alors observer la surface atteignable par le robot pour $l_1 = 0,65 \text{ m}$ et $l_2 = 0,35 \text{ m}$:



Surface atteignable par le bout du bras du robot

Etude du MGI

On étudie ensuite le modèle géométrique inverse (MGI) du bras manipulateur pour avoir une fonction g telle que :

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} = g(X)$$

On a alors :

$$\begin{cases} \theta_2 = \pm \arccos\left(\frac{x_r^2 + y_r^2 - (l_1^2 + l_2^2)}{2l_1 l_2}\right) \\ \theta_1 = \arccos\left(\frac{x_r(l_1 + l_2 \cos(\theta_2)) + y_r l_2 \sin(\theta_2)}{x_r^2 + y_r^2}\right) \\ \theta_1 = \arcsin\left(\frac{y_r(l_1 + l_2 \cos(\theta_2)) - x_r l_2 \sin(\theta_2)}{x_r^2 + y_r^2}\right) \end{cases}$$

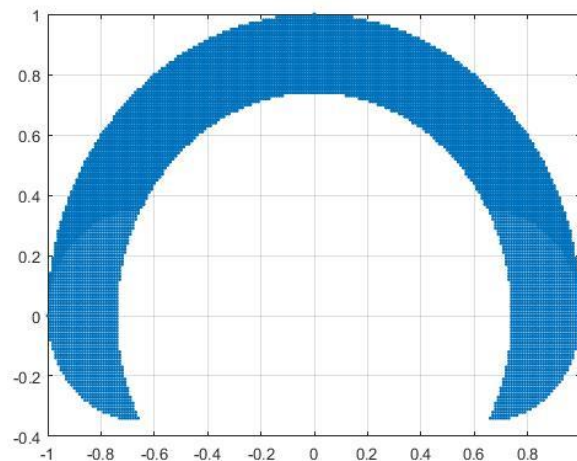
Pour vérifier la validité des résultats, on a 4 critères à respecter :

$$x_r^2 + y_r^2 \leq l_1 + l_2 \quad , \quad \left| \frac{x_r^2 + y_r^2 - (l_1^2 + l_2^2)}{2l_1 l_2} \right| \leq 1 \quad , \quad -\frac{\pi}{2} \leq \theta_1 \leq \frac{\pi}{2} \quad , \quad 0 \leq \theta_2 \leq \pi$$

Traduit dans Matlab, on a le code suivant :

```
a = xr*xr + yr*yr;
b = (a - (l1*l1+l2*l2))/(2*l1*l2);
if a<=l1+l2 && abs(b)<=1
    %coude bas
    theta2 = acos(b);
    if all(theta2<=pi/2 && theta2>=-pi/2)
        t1 =
        acos((xr*(l1+l2*cos(theta2))+yr*l2*sin(theta2))/(xr*xr+yr*yr));
        t2 = asin((yr*(l1+l2*cos(theta2))-
        xr*l2*sin(theta2))/(xr*xr+yr*yr));
        theta1 = atan2(sin(t2),cos(t1));
        if all(theta1<=pi && theta1>=0)
            k = k+1;
            x(k) = xr; y(k) = yr;
        end
    end
    %coude haut
    theta22 = -acos(b);
    t1 =
    acos((xr*(l1+l2*cos(theta22))+yr*l2*sin(theta22))/(xr*xr+yr*yr));
    t2 = asin((yr*(l1+l2*cos(theta22))-
    xr*l2*sin(theta22))/(xr*xr+yr*yr));
    theta12 = atan2(sin(t2),cos(t1));
    if all(theta12<=pi && theta12>=0)
        k = k+1;
        x(k) = xr; y(k) = yr;
    end
end
end
end
```

Si on trace le graphe correspondant aux solutions possibles, on obtient la figure suivante :



Ensemble des points respectant les critères de validité

On remarque ainsi que le graphe correspond à celui obtenu avec le MGD.

Etude du MCD

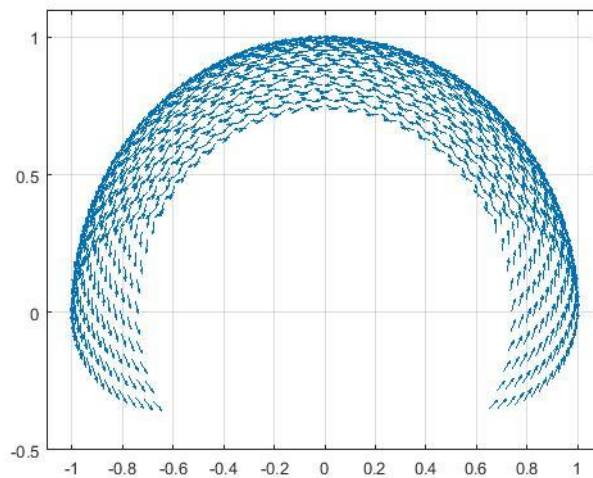
Maintenant, on s'intéresse au modèle cinématique direct (MCD) du bras. On cherche la matrice J telle que :

$$\dot{X} = J\dot{\theta}$$

En dérivant X pour obtenir \dot{X} , on a alors :

$$\dot{X} = \begin{pmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}$$

En considérant les vitesses maximales dans l'espace articulaire $|\dot{\theta}_{1max}| = 5rd/s$ et $|\dot{\theta}_{2max}| = 5rd/s$, on a le champ de vitesses suivant :



Champ de vecteurs des vitesses cartésiennes

Etude du MCI

On peut alors déduire du MCD le modèle cinématique inverse (MCI), qui est tel que :

$$\dot{\theta} = J^{-1}\dot{e}$$

avec $\dot{e} = -\lambda e$ et $e = P_{des} - P$, où P est le vecteur des positions désirées.

On obtient donc :

$$\dot{\theta} = -\lambda J(\theta)^{-1}e$$

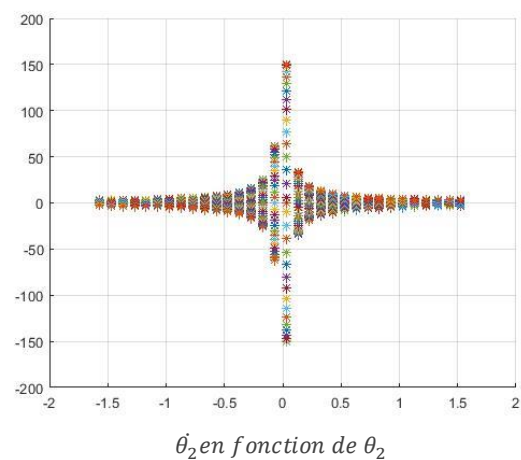
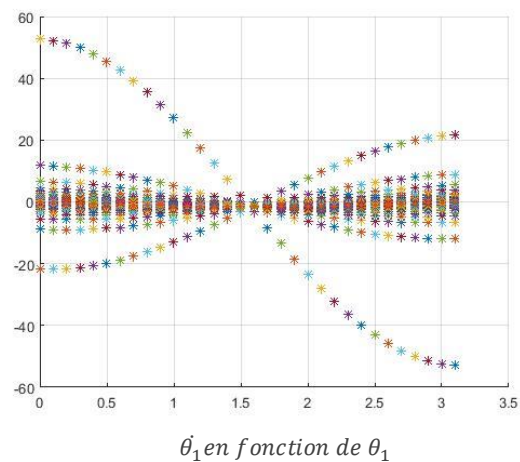
Cette équation est valide seulement si J est inversible, c'est-à-dire si son déterminant est non nul, soit :

$$(-l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2)) l_2 \cos(\theta_1 + \theta_2) + (l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)) l_2 \sin(\theta_1 + \theta_2) \neq 0$$

$$\Leftrightarrow \sin(\theta_2) \neq 0$$

$$\Leftrightarrow \theta_2 \neq k\pi \text{ avec } k \in \mathbb{N}$$

Ainsi, en prenant une vitesse constante $\dot{x}_r = 1m/s$ et $\dot{y}_r = 0m/s$, on obtient les graphes des vitesses angulaires en fonction des positions articulaires suivantes :

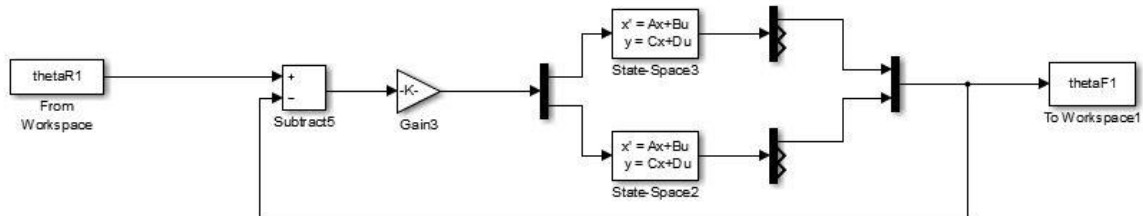


On peut remarquer que la vitesse angulaire en $\dot{\theta}_1$ est bien plus élevée quand θ_1 est proche de 0 ou π . De même pour $\dot{\theta}_2$, quand θ_2 vaut 0.

Commande dans l'espace de la tâche en utilisant le MGI

On décide maintenant de contrôler le bras manipulateur en utilisant le MGI décrit précédemment.

Sous Simulink, on modélise le robot par 2 moteurs, donnant ainsi le modèle schéma-blocs suivant :

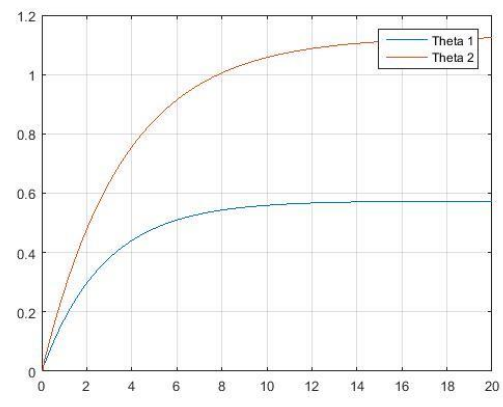
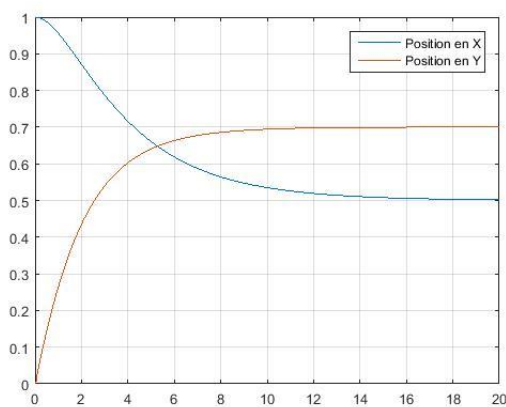


L'entrée *thetaR1* correspond aux valeurs d'angles calculées par le MGI, tandis que la sortie *thetaF1* correspond aux angles qui seront traités dans le MGD pour obtenir la position du bras en fonction de chaque pas de temps.

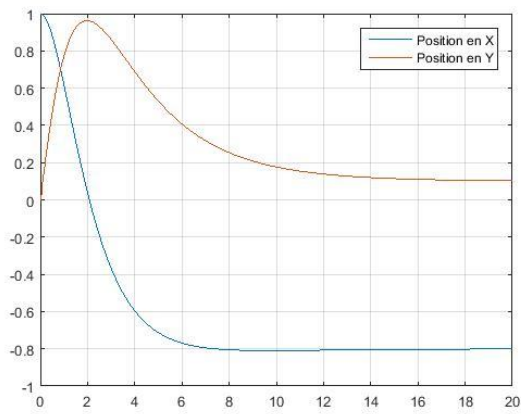
Le gain sert à pouvoir visualiser l'évolution de la position et des angles plus nettement (ici, on a $K=0.01$)

Les *State-Spaces* correspondent à chacun des moteurs (moteur 1 en haut, moteur 2 en bas)

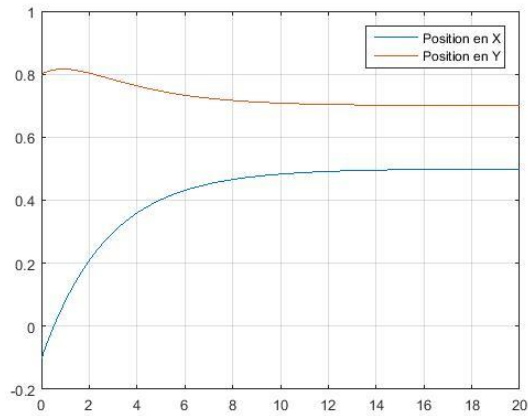
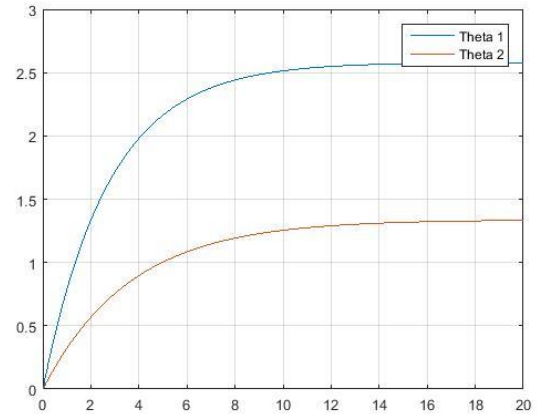
En prenant plusieurs positions initiales et finales pour le bout du bras de robot, et en remplaçant la position non-atteignable $P_{init_2} = [-0.1 \ -0.8]^T m$ par $P_{init_2} = [-0.1 \ 0.8]^T m$, et en se limitant coudes bas initial et final, on obtient les graphes suivants :



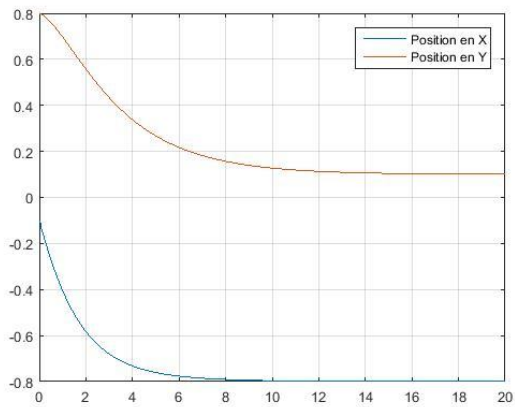
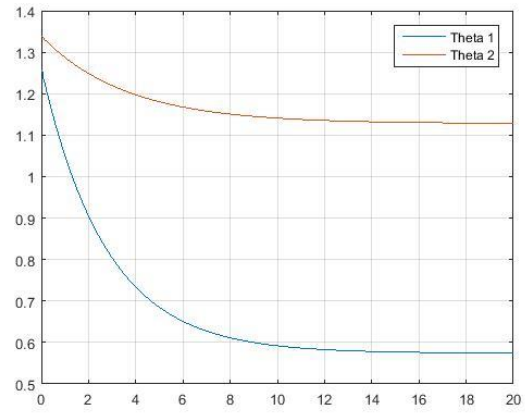
Positions et angles pour $P_i = [1 \ 0]$ et $P_f = [0.5 \ 0.7]$



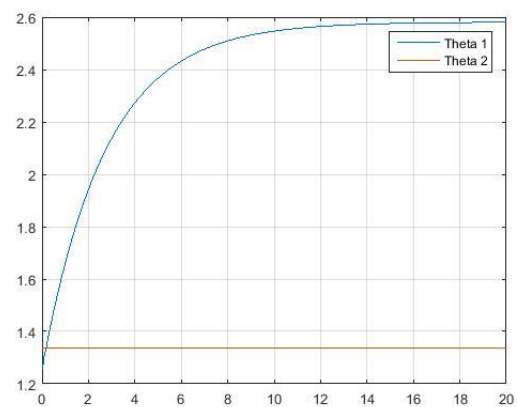
Positions et angles pour $P_i = [10]$ et $P_f = [-0.8 \ 0.1]$



Positions et angles pour $P_i = [-0.1 \ 0.8]$ et $P_f = [0.5 \ 0.7]$



Positions et angles pour $P_i = [-0.1 \ 0.8]$ et $P_f = [-0.8 \ 0.1]$

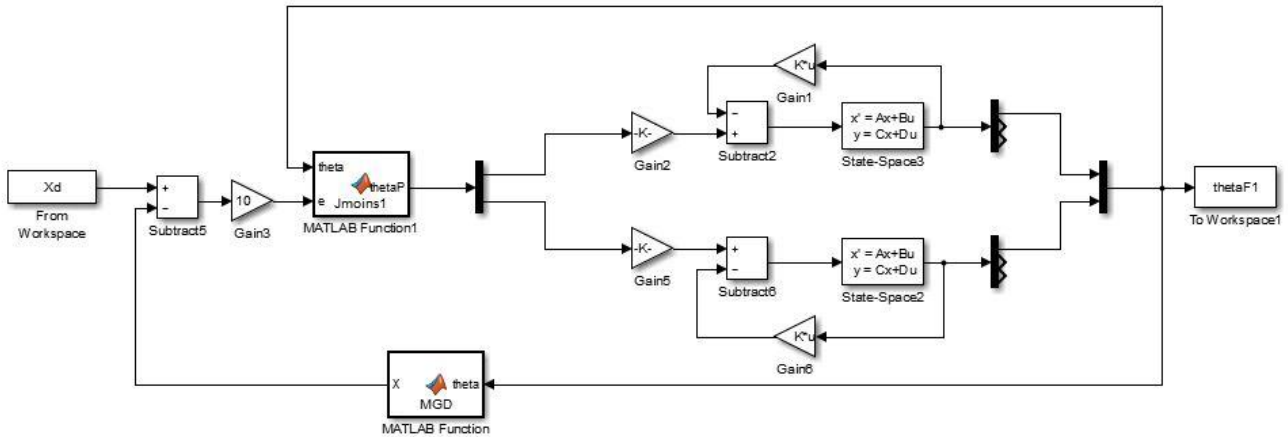


On remarque que la position finale est atteinte plus ou moins rapidement suivant son éloignement de la position initiale (ce qui est logique), mais en gardant le même sens d'évolution des θ . Par exemple, dans le cas $P_{init} = [-0.1 \ 0.8]^T m$ et $P_{fin} = [-0.8 \ 0.1]^T m$, on peut voir que θ_1 augmente continuellement, et ce dès le début, vers sa position finale et que θ_2 reste constant.

Commande dans l'espace de la tâche en utilisant le MCI

On décide maintenant de contrôler le bras manipulateur en utilisant le MCI décrit précédemment.

Sous Simulink, on modélise le robot par 2 moteurs, donnant ainsi le modèle schéma-blocs suivant :



L'entrée X_d correspond à la position finale à atteindre, tandis que la sortie θ_{F1} correspond aux angles qui seront traités dans le MGD par la suite pour obtenir la position du bras à chaque pas de temps.

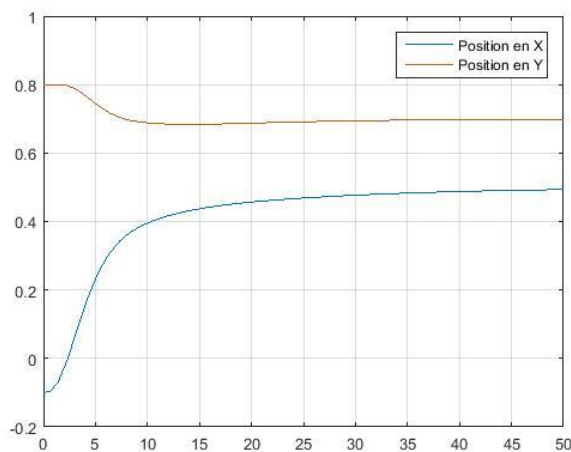
Le premier gain ($Gain3$) correspond au terme λ présent dans l'équation permettant de définir $\dot{\theta}$. Les deux gains $Gain2$ et $Gain5$ ne servent qu'à réduire le signal afin de pouvoir visualiser plus nettement les angles et les positions. Les gains $Gain1$ et $Gain6$ servent de retour correctif, vu que l'on utilise ici un contrôle en vitesse (cf. TP1).

Les $State-Space3$ et $State-Space2$ représentent les moteurs, respectivement le moteur 1 et le moteur 2.

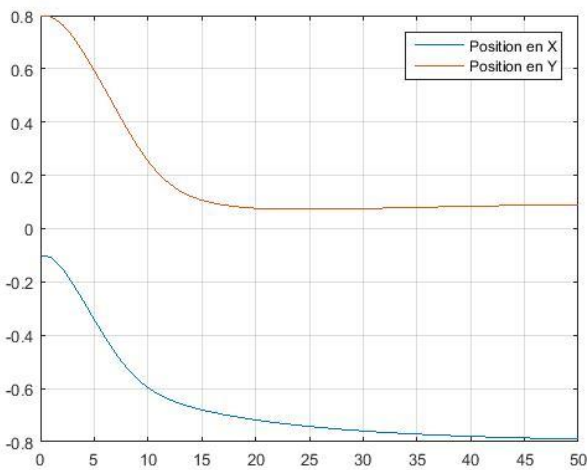
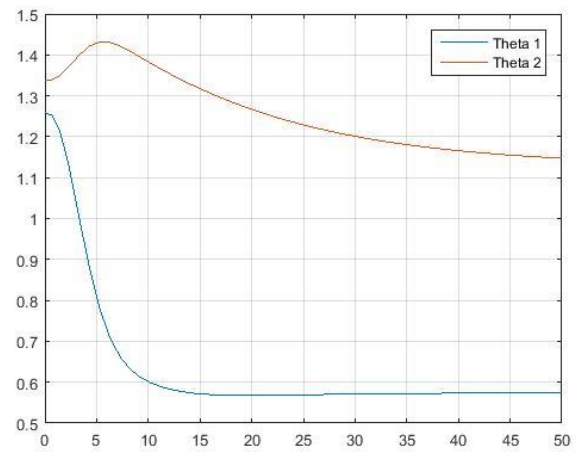
La fonction J_{moins1} permet de calculer la matrice J^{-1} , et sert à passer des \dot{X} aux $\dot{\theta}$. La fonction MGD permet quant à elle de changer les θ en X , en utilisant le MGD défini plus tôt.

En prenant différentes positions initiales et finales pour le bout du bras du robot, on peut alors effectuer le contrôle en vitesse grâce au programme Simulink ci-dessus. Il faut par contre vérifier le critère de validité sur J^{-1} , qui dit que $\theta_2 \neq k\pi$, avec $k \in \mathbb{N}$. Ainsi, la position $P_{init_1} = [1.0 \ 0.0]^T m$ n'est pas utilisable comme position initiale, car on a $\theta_2 = 0$.

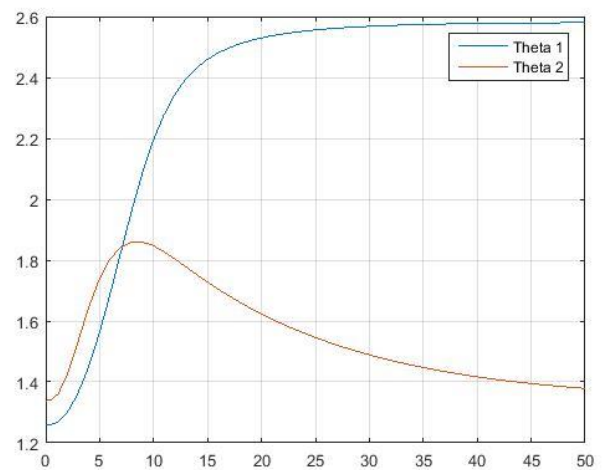
Par contre, en prenant $P_{init_2} = [-0.1 \ 0.8]^T m$, on a les courbes suivantes :



Positions et angles pour $P_i = [-0.1 \ 0.8]$ et $P_f = [0.5 \ 0.7]$



Positions et angles pour $P_i = [-0.1 \ 0.8]$ et $P_f = [-0.8 \ 0.1]$



On peut remarquer que le sens d'évolution des θ varie en fonction du temps : ainsi, on peut voir dans le second cas que θ_2 augmente puis diminue, tout cela pour partir de sa position initiale et y retourner à la fin.

Finalement, on peut comparer le contrôle avec le MCI et le contrôle avec le MGI : le MGI permet d'avoir toutes les positions initiales possibles, tant qu'elles restent dans les domaines de définition des angles, et permet de choisir entre coude haut et coude bas ; le MCI lui permet de contrôler en position, et donc offre de meilleures performances pour ce qui est de la vitesse de convergence de la solution.