
Rapport TP Commande

Romain BANNERY
Max SONZOGNI
MI5

Table des matières

I.	Modélisation du robot	2
II.	Simulation du contrôle du robot.....	3
III.	Hiérarchisation des tâches.....	4
a.	Position A : $PR = -0.7$, $PL = 0.5$, $CdM = 0$	4
1.	Bras gauche-Centre de masse-Bras droit.....	4
2.	Centre de masse-Bras droit-Bras gauche.....	5
3.	Bras droit-Bras gauche-Centre de masse.....	6
b.	Position B : $PR = -0.9$, $PL = 1$, $CdM = 0$	7
1.	Bras gauche-Centre de masse-Bras droit.....	7
2.	Centre de masse-Bras droit-Bras gauche.....	8
3.	Bras droit-Bras gauche-Centre de masse.....	9
IV.	Conclusion.....	10
V.	Annexes.....	11
a.	Détail du bloc <i>MATLAB Function</i> du schéma Simulink.....	11

I. Modélisation du robot

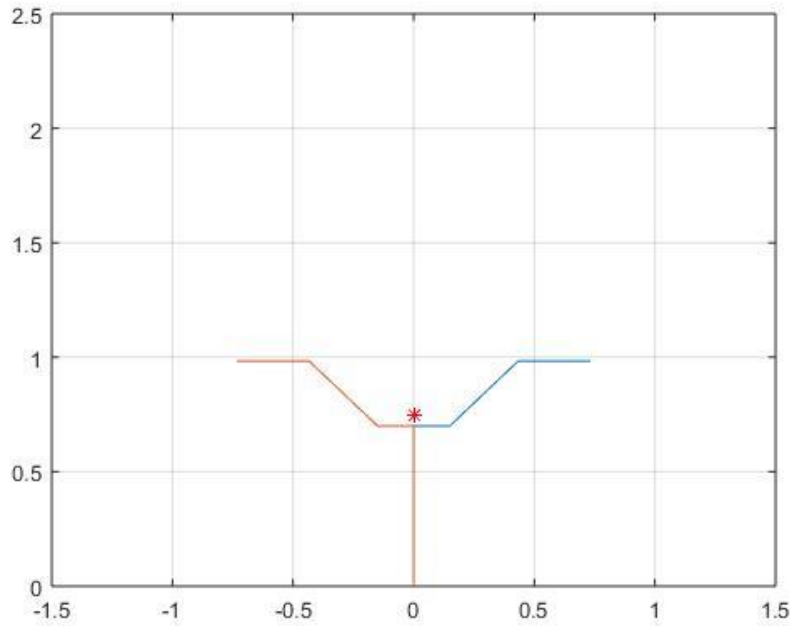


Figure 1: Représentation de la position initiale du robot sous Matlab

Nous avons dans un premier temps donné des positions à atteindre au robot. Nous avons donc défini trois positions distinctes appartenant au robot via le modèle géométrique direct :

- La position horizontale de la main gauche P_L
- La position horizontale de la main droite P_R
- La position horizontale du centre de masse CdM (ici représenté par une étoile rouge)

Ainsi, lorsque nous voulons que le robot effectue un mouvement nous avons juste à lui fournir les coordonnées des positions P_L et P_R que nous désirons atteindre. Dans le cas de la figure ci-dessus, les mains gauches et droites ont les mêmes coordonnées en ordonnées mais des coordonnées de signes opposés en abscisse. On décide par la suite de toujours lui indiquer d'avoir son centre de masse en position $CdM = 0$ pour qu'il soit stable.

Les positions P_L , P_R et CdM sont données par les formules suivantes :

$$\begin{aligned}
 P_L &= -l_1 \sin(\theta_1) + \frac{le}{2} \cos(\theta_1) + l_2 \cos(-\theta_1 + \theta_2) + l_3 \cos(-\theta_1 + \theta_2 - \theta_3) \\
 P_R &= -l_1 \sin(\theta_1) - \frac{le}{2} \cos(\theta_1) - l_4 \cos(-\theta_1 + \theta_4) - l_5 \cos(-\theta_1 + \theta_4 - \theta_5) \\
 CdM &= \frac{m_1 OC_1 + m_2 OC_2 + m_3 OC_3 + m_4 OC_4 + m_5 OC_5}{m_1 + m_2 + m_3 + m_4 + m_5} \\
 OC_1 &= -c_1 \sin(\theta_1) \\
 OC_2 &= -l_1 \sin(\theta_1) + \frac{le}{2} \cos(\theta_1) + c_2 \cos(-\theta_1 + \theta_2) \\
 OC_3 &= -l_1 \sin(\theta_1) + \frac{le}{2} \cos(\theta_1) + l_2 \cos(-\theta_1 + \theta_2) + c_3 \cos(-\theta_1 + \theta_2 - \theta_3) \\
 OC_4 &= -l_1 \sin(\theta_1) - \frac{le}{2} \cos(\theta_1) + c_4 \cos(-\theta_1 + \theta_4) \\
 OC_5 &= -l_1 \sin(\theta_1) - \frac{le}{2} \cos(\theta_1) - l_4 \cos(-\theta_1 + \theta_4) - c_5 \cos(-\theta_1 + \theta_4 - \theta_5)
 \end{aligned}$$

II. Simulation du contrôle du robot

Pour contrôler le robot, il a été nécessaire de créer un schéma Simulink qui simule son comportement afin de pouvoir le faire fonctionner sous Matlab. Il est représenté dans la Figure 2 ci-dessous, et le détail du bloc *MATLAB Function* est contenu dans l'Annexe A.

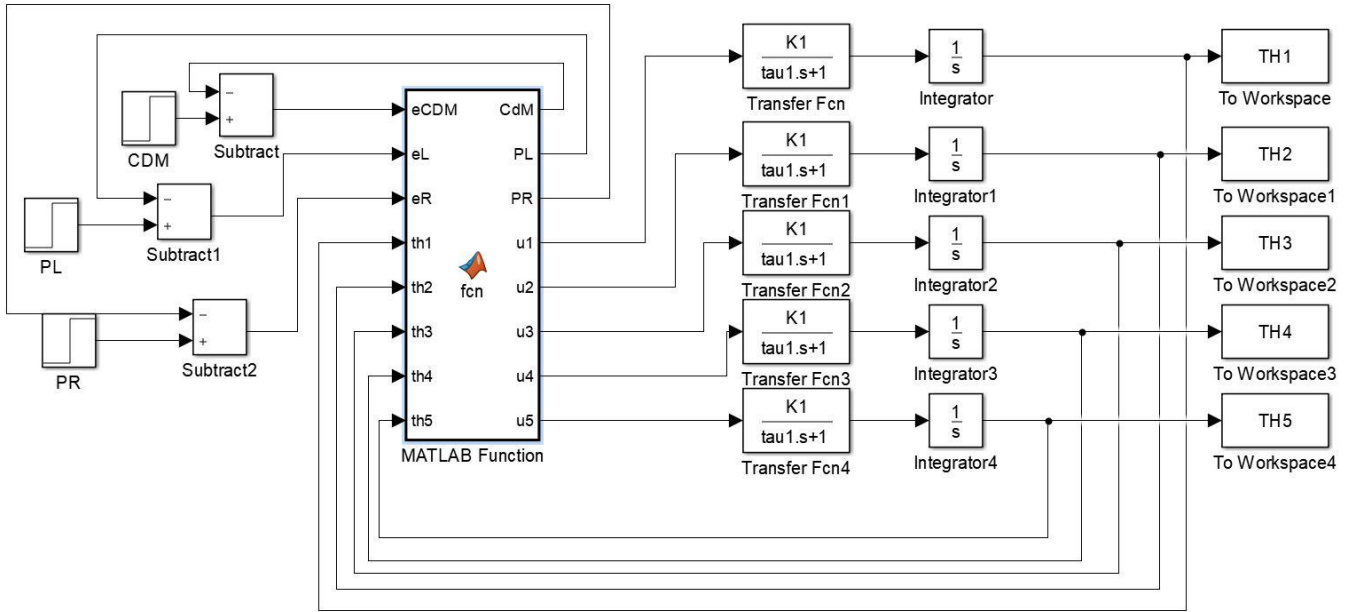


Figure 2 : Représentation sous Simulink du schéma de commande du robot

Pour pouvoir déterminer les vitesses angulaires $\dot{\theta}$ qui permettront aux moteurs de tourner, il faut passer par les matrices Jacobiennes des positions. Celles-ci sont calculées grâce à la formule $J = \frac{dP}{d\theta}$ avec $\theta = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]^T$. On obtient alors les matrices suivantes :

$$J_L = \begin{bmatrix} -l_1 \cos(\theta_1) - \frac{le}{2} \sin(\theta_1) + l_2 \sin(-\theta_1 + \theta_2) + l_3 \sin(-\theta_1 + \theta_2 - \theta_3) \\ l_2 \sin(-\theta_1 + \theta_2) - l_3 \sin(-\theta_1 + \theta_2 - \theta_3) \\ l_3 \sin(-\theta_1 + \theta_2 - \theta_3) \\ 0 \\ 0 \end{bmatrix}^T$$

$$J_R = \begin{bmatrix} -l_1 \cos(\theta_1) + \frac{le}{2} \sin(\theta_1) - l_4 \sin(-\theta_1 + \theta_4) - l_5 \sin(-\theta_1 + \theta_4 - \theta_5) \\ 0 \\ 0 \\ l_4 \sin(-\theta_1 + \theta_4) + l_5 \sin(-\theta_1 + \theta_4 - \theta_5) \\ -l_5 \sin(-\theta_1 + \theta_4 - \theta_5) \end{bmatrix}^T$$

Il en est de même pour la matrice J_{CdM} qui est la somme pondérée des matrices jacobiennes des positions J_{OC_i} (détail disponible dans l'Annexe A).

Les conditions initiales qui ont été appliquées aux différents angles sont $\theta = [0 \ -\frac{\pi}{4} \ -\frac{\pi}{4} \ \frac{\pi}{4} \ \frac{\pi}{4}]$. Elles ont été spécifiées dans les blocs *Integrator*.

III. Hiérarchisation des tâches

a. Position A : $PR = -0.7$, $PL = 0.5$, $CdM = 0$

1. Bras gauche-Centre de masse-Bras droit

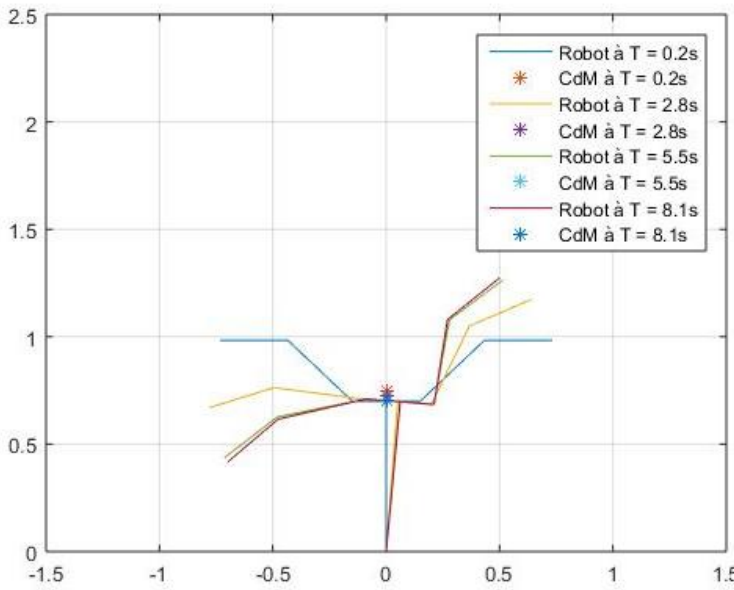


Figure 3 : Représentation du robot à 4 instants différents sous la hiérarchie Bras gauche – Centre de masse – Bras droit

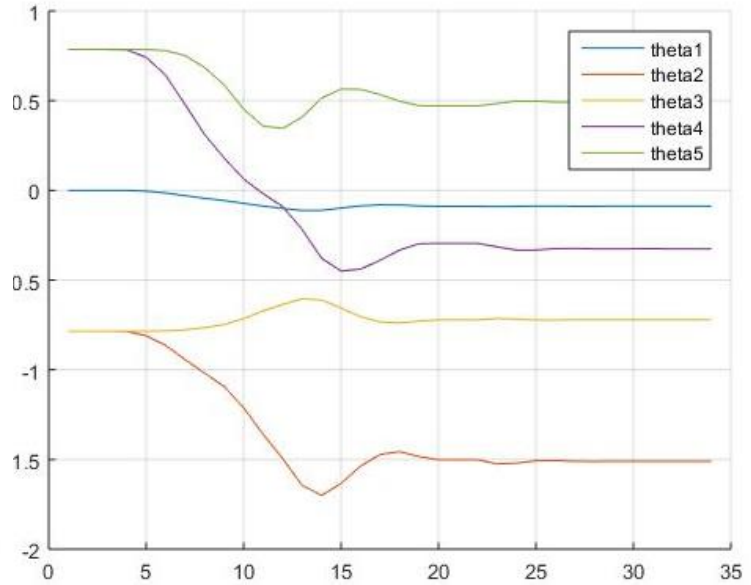


Figure 4: Evolution des 5 angles des articulations du robot

Positions finales : $PR = \begin{pmatrix} -0.6998 \\ 0.4162 \end{pmatrix}$, $PL = \begin{pmatrix} 0.4995 \\ 1.2733 \end{pmatrix}$, $CdM = \begin{pmatrix} 0 \\ 0.6999 \end{pmatrix}$

Ce choix dans les positions des bras gauche, droit et du centre de masse ne permet techniquement d'en réaliser précisément que deux à la fois, il est donc tout à fait normal d'avoir des écarts sur au moins une des valeurs finales.

On constate qu'avec cette hiérarchie, le centre de masse est atteint précisément, mais pas les positions P_L et P_R . Celles-ci sont décalées respectivement de 0.1% et 0.02% par rapport aux valeurs désirées, ce qui est tout à fait acceptable. Cette hiérarchie peut donc être considérée comme valable dans ce cas-ci.

2. Centre de masse-Bras droit-Bras gauche

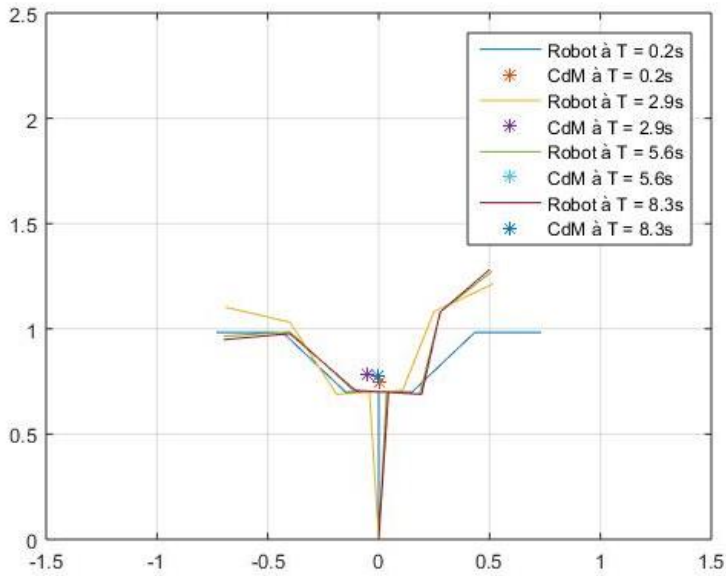


Figure 6 : Représentation du robot à 4 instants différents sous la hiérarchie Centre de masse - Bras droit - Bras gauche

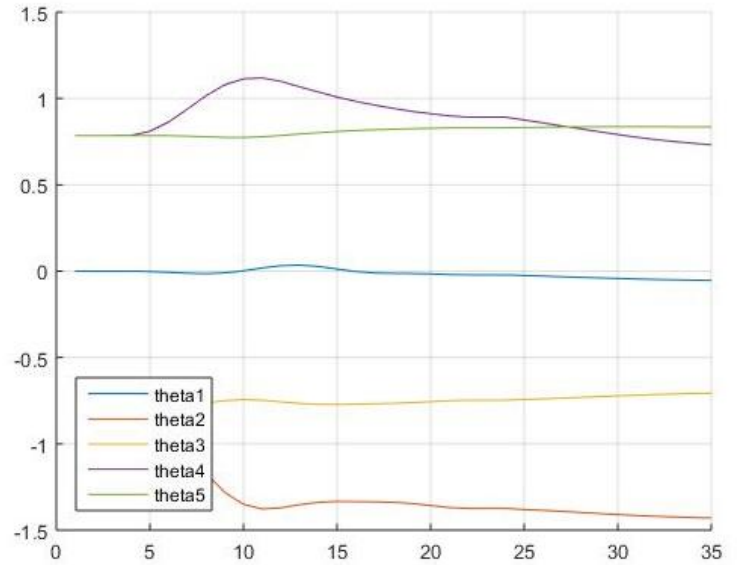


Figure 5 : Evolution des 5 angles des articulations du robot

Positions finales : $PR = \begin{pmatrix} -0.7 \\ 0.9486 \end{pmatrix}, PL = \begin{pmatrix} 0.5001 \\ 1.2814 \end{pmatrix}, CdM = \begin{pmatrix} 0 \\ 0.7729 \end{pmatrix}$

En comparant cette situation à la précédente, on peut déjà constater que le positionnement du robot est beaucoup plus précis sur la position du bras droit. En effet, si nous calculons les erreurs relatives de la même manière que précédemment, nous obtenons un placement exact du bras droit et du centre de masse, mais toujours le même décalage de 0.02% sur la position du bras gauche. Cette hiérarchie est donc plus précise que la précédente en ce qui concerne ce positionnement.

3. Bras droit-Bras gauche-Centre de masse

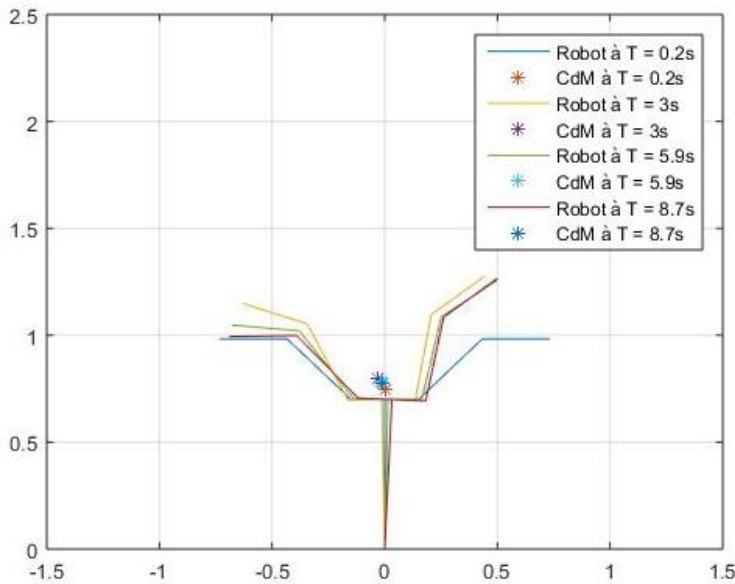


Figure 8 : Représentation du robot à 4 instants différents sous la hiérarchie
Bras droit - Bras gauche - Centre de masse

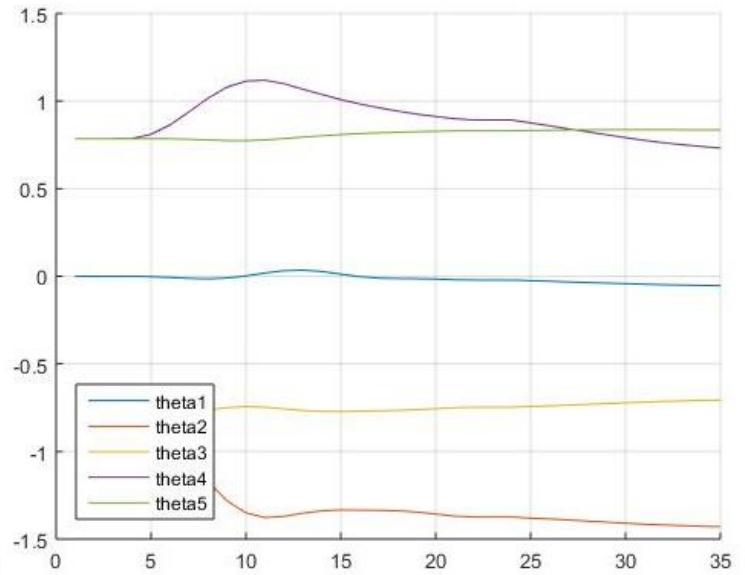


Figure 7 : Evolution des 5 angles des articulations du robot

Positions finales : $PR = \begin{pmatrix} -0.6907 \\ 0.9941 \end{pmatrix}, PL = \begin{pmatrix} 0.5 \\ 1.2664 \end{pmatrix}, CdM = \begin{pmatrix} -0.0073 \\ 0.7780 \end{pmatrix}$

Ici, le décalage des différentes positions est plutôt significatif. En effet, seul le bras gauche atteint la position désirée avec cette hiérarchie, tandis que l'on a une erreur relative de positionnement de 1.3% sur le bras droit et que le centre de masse est décalé de 7.10^{-3} par rapport à l'axe vertical. Cette situation rend donc le robot instable et tend à le faire pencher puis tomber si les efforts résistants appliqués sur le pied du robot n'exercent pas un couple suffisant pour l'en empêcher. Cette hiérarchie n'est donc pas à privilégier.

b. Position B : $PR = -0.9$, $PL = 1$, $CdM = 0$

1. Bras gauche-Centre de masse-Bras droit

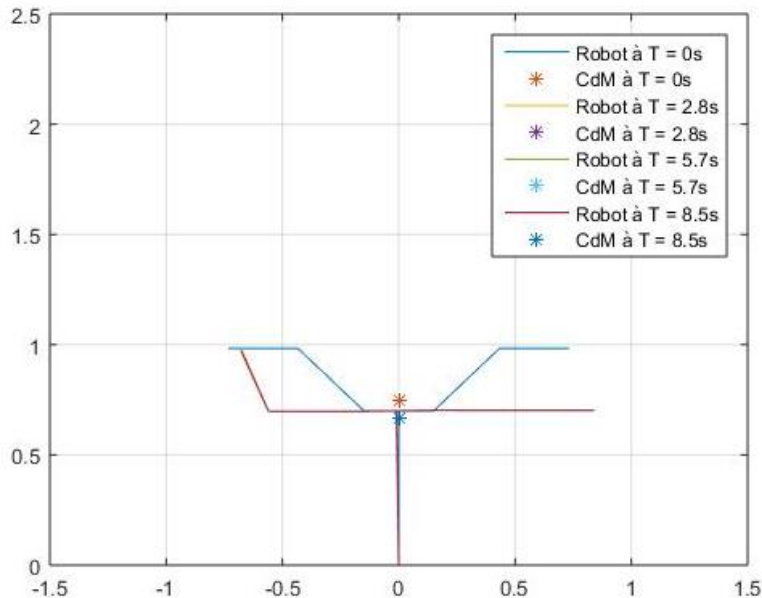


Figure 10 : Représentation du robot à 4 instants différents sous la hiérarchie
Bras gauche - Centre de masse - Bras droit

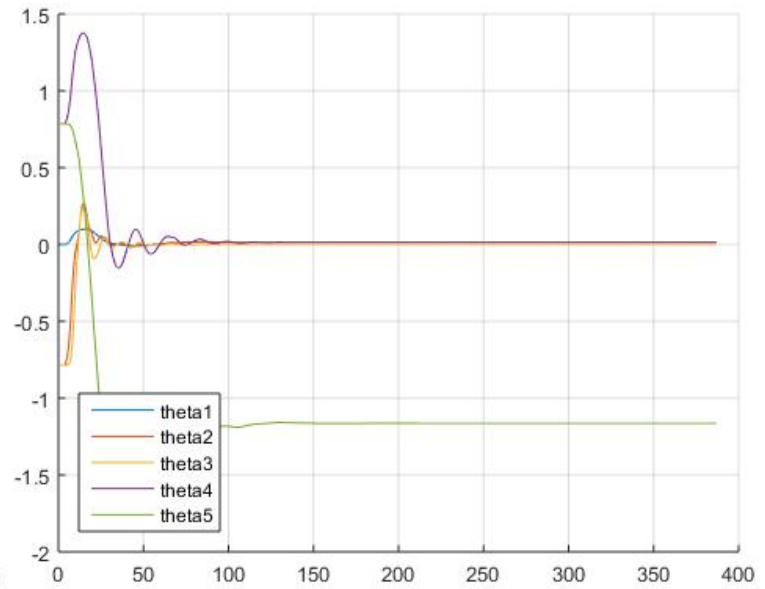


Figure 9 : Evolution des 5 angles des articulations du robot

Positions finales : $PR = \begin{pmatrix} -0.6785 \\ 0.9733 \end{pmatrix}$, $PL = \begin{pmatrix} 0.8402 \\ 0.7018 \end{pmatrix}$, $CdM = \begin{pmatrix} 0 \\ 0.6637 \end{pmatrix}$

En choisissant ce positionnement-ci, il est techniquement impossible d'atteindre plus d'une des positions voulues à la fois. Il est donc normal de s'attendre à ce qu'il y ait de grandes erreurs sur le positionnement des bras et du centre de masse suivant la hiérarchie adoptée.

Dans ce premier cas test, on obtient le positionnement exact du centre de masse, qui se trouve en 2^e place de la hiérarchie des tâches. Le bras gauche est placé tel que l'erreur relative entre sa position finale et la position désirée est de 18.7%, tandis que le bras droit atteint lui 32.6% d'erreur relative sur son placement.

2. Centre de masse-Bras droit-Bras gauche

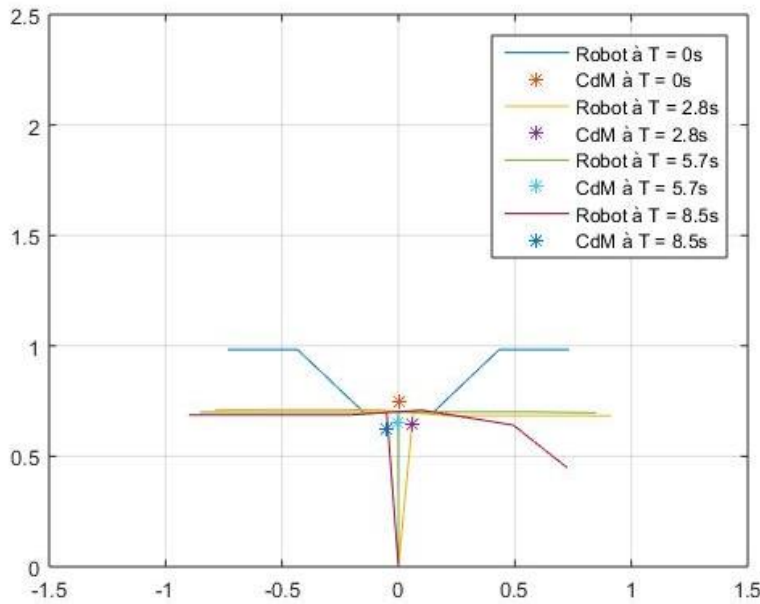


Figure 11 : Représentation du robot à 4 instants différents sous la hiérarchie
Centre de masse - Bras droit - Bras gauche

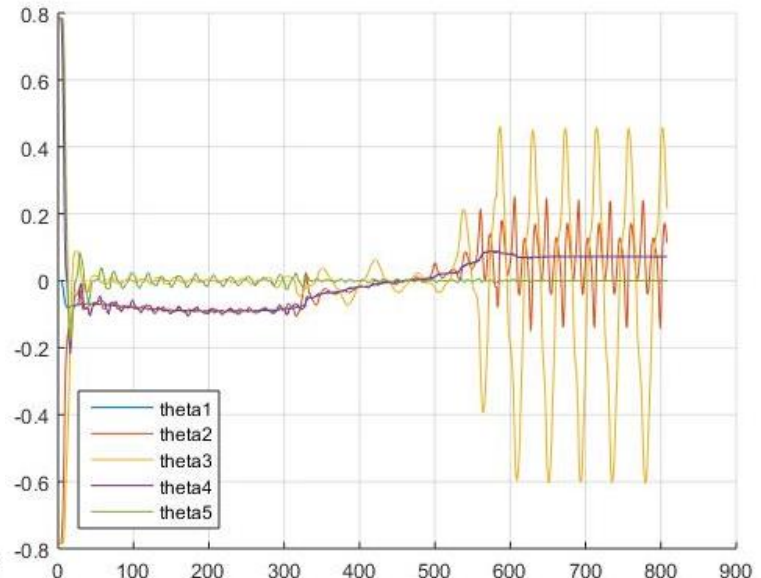


Figure 12 : Evolution des 5 angles des articulations du robot

Positions finales : $PR = \begin{pmatrix} -0.9001 \\ 0.6868 \end{pmatrix}, PL = \begin{pmatrix} 0.7236 \\ 0.4489 \end{pmatrix}, CdM = \begin{pmatrix} -0.0514 \\ 0.6268 \end{pmatrix}$

Ici, c'est le bras droit, en 2° position dans la hiérarchie des tâches, qui est le mieux placé. Son erreur relative de positionnement n'est "que" de 0.01%, contrairement au bras gauche qui lui possède une erreur de 38.2% par rapport à sa position désirée, ce qui n'est pas du tout négligeable quant au choix de la hiérarchie. Le centre de masse est lui décalé de $5 \cdot 10^{-2}$ par rapport à l'axe vertical, ce qui le rend une fois encore instable. Il n'est donc pas conseillé de choisir cette hiérarchie sans s'être assuré au préalable que le robot ne risque pas de basculer dans cette situation.

On peut remarquer que dans cette situation, les angles θ_2 et θ_3 varient sans atteindre de point de convergence. Cela implique que le bras gauche va bouger indéfiniment une fois le bras droit stabilisé. Ceci va faire varier le positionnement du centre de masse, rendant ainsi encore plus instable le robot. Il n'est donc pas conseillé de prendre cette hiérarchie de tâches si on cherche à avoir une position finale qui soit parfaitement fixe.

3. Bras droit-Bras gauche-Centre de masse

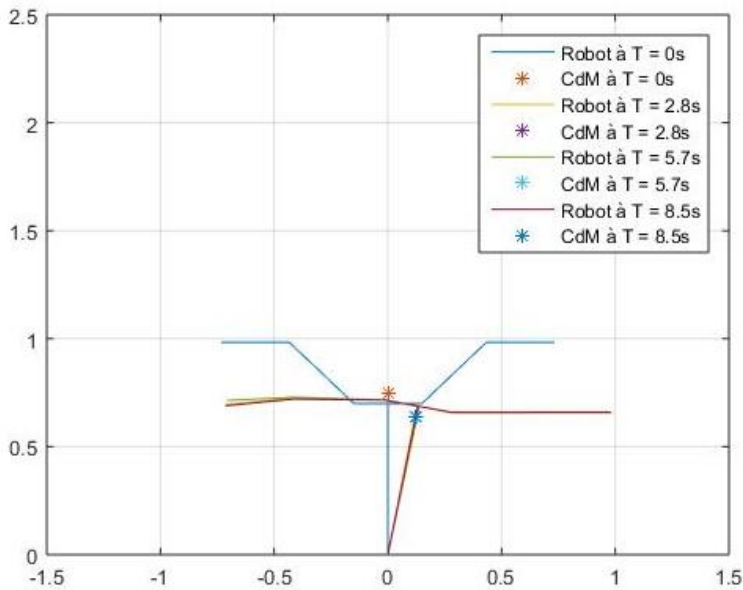


Figure 13 : Représentation du robot à 4 instants différents sous la hiérarchie Bras droit - Bras gauche - Centre de masse

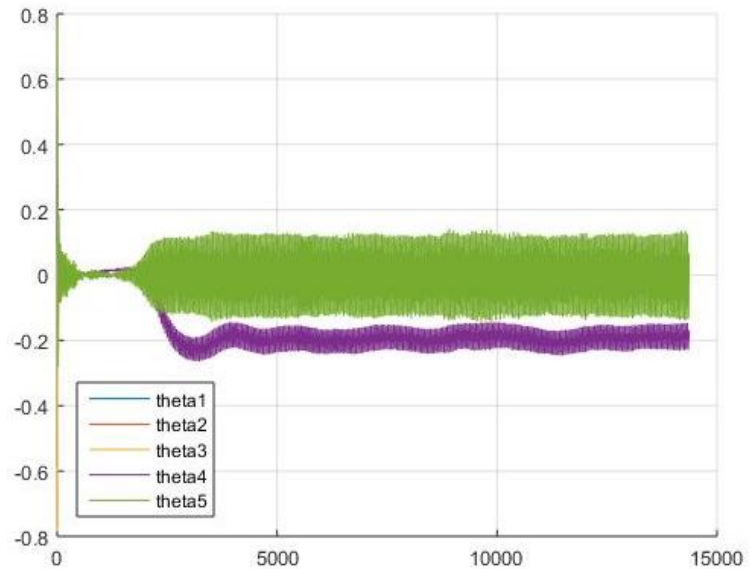


Figure 14 : Evolution des 5 angles des articulations du robot

Positions finales : $PR = \begin{pmatrix} -0.7132 \\ 0.6894 \end{pmatrix}, PL = \begin{pmatrix} 0.9797 \\ 0.6595 \end{pmatrix}, CdM = \begin{pmatrix} 0.1230 \\ 0.6376 \end{pmatrix}$

Finalement, pour ce qui est de la configuration Bras droit – Bras gauche – Centre de masse, on obtient les pires valeurs de positionnement. En effet, le bras gauche atteint les 2% d'erreur de positionnement, et le bras droit possède quant à lui une différence de 26.1% entre sa position finale et la position désirée. Le centre de masse est lui décalé de 0.12 par rapport à l'axe vertical passant par son pied, ce qui le rend très instable et prompt à tomber. Cette hiérarchie est donc à éviter ou à n'utiliser qu'en dernier recours.

On peut remarquer qu'ici aussi la position finale n'est pas stabilisée : en effet, les angles θ_4 et θ_5 du bras droit oscillent sans atteindre de valeur de convergence, montrant que cette configuration n'est pas du tout stable dans sa hiérarchie.

IV. Conclusion

Dans le cas où certaines positions finales sont impossibles à atteindre, c'est la tâche en 2° position dans la hiérarchie qui sera privilégiée, suivie de celle en 1° position puis de celle en 3°. Cela peut sembler contre-intuitif du fait que ce ne soit pas la tâche en première position qui soit considérée comme prioritaire.

Ainsi, la configuration qui prend la position du centre de masse en 2° position dans sa hiérarchie de tâche est celle où le robot sera le plus proche de l'équilibre, du fait que son centre de masse soit aligné avec la droite verticale passant par son pied. C'est aussi sous cette configuration que l'on a les écarts les plus faibles entre les positions désirées des bras et leurs positions finales, ce qui en fait le choix de hiérarchie le plus sûr pour avoir un positionnement exact du robot.

De plus, avoir un bras qui ne soit pas parfaitement positionné peut entraîner une oscillation de celui-ci, du fait du robot qui va essayer de satisfaire la condition sur ce bras tout en respectant les contraintes déjà remplies. Ceci peut déboucher sur la création de vibrations dans le robot, ce qui le déstabilisera et pourra entraîner sa chute ou la rupture de certaines de ses articulations.

V. Annexes

a. Détail du bloc *MATLAB Function* du schéma Simulink

```
function [CdM, PL, PR, u1, u2, u3, u4, u5] = fcn(eCDM, eL, eR, th1, th2, th3, th4, th5)

le = 0.3; l1 = 0.7; l2 = 0.4; l3 = 0.3; l4 = 0.4; l5 = 0.3;
c1 = 0.6; c2 = 0.2; c3 = 0.15; c4 = 0.2; c5 = 0.15;
m1 = 20; m2 = 6; m3 = 4; m4 = 6; m5 = 4;

%calcul des jacobienness
JL = [-l1*cos(th1)-(le/2)*sin(th1)+l2*sin(-th1+th2)+l3*sin(-th1+th2-th3) -l2*sin(-th1+th2)-
l3*sin(-th1+th2-th3) l3*sin(-th1+th2-th3) 0 0];
JR = [-l1*cos(th1)+(le/2)*sin(th1)-l4*sin(-th1+th4)-l5*sin(-th1+th4-th5) 0 0 l4*sin(-
th1+th4)+l5*sin(-th1+th4-th5) -l5*sin(-th1+th4-th5)];

cdm1 = [-c1*cos(th1) 0 0 0 0];
cdm2 = [-l1*cos(th1)-(le/2)*sin(th1)+c2*sin(-th1+th2) -c2*sin(-th1+th2) 0 0 0];
cdm3 = [-l1*cos(th1)-(le/2)*sin(th1)+l2*sin(-th1+th2)+c3*sin(-th1+th2-th3) -l2*sin(-th1+th2)-
c3*sin(-th1+th2-th3) c3*sin(-th1+th2-th3) 0 0];
cdm4 = [-l1*cos(th1)+(le/2)*sin(th1)-c4*sin(-th1+th4) 0 0 c4*sin(-th1+th4) 0];
cdm5 = [-l1*cos(th1)+(le/2)*sin(th1)-l4*sin(-th1+th4)-c5*sin(-th1+th4-th5) 0 0 l4*sin(-
th1+th4)+c5*sin(-th1+th4-th5) -c5*sin(-th1+th4-th5)];
Jcdm = (m1*cdm1+m2*cdm2+m3*cdm3+m4*cdm4+m5*cdm5)/(m1+m2+m3+m4+m5);

%hiérarchisation
J1 = JR; J2 = JL; J3 = Jcdm;
e1 = eR; e2 = eL; e3 = eCDM;

P1 = eye(5) - pinv(J1)*J1;
P2 = eye(5) - pinv(J2)*J2;

%détermination des valeurs de theta_point
th_1 = pinv(J1)*e1;
th_2 = pinv(J2*P1)*(e2-J2*th_1);
th_3 = pinv(J3*P2)*(e3-J3*th_2);

thetaP = th_1+th_2+th_3;
u1 = thetaP(1); u2 = thetaP(2); u3 = thetaP(3); u4 = thetaP(4); u5 = thetaP(5);

%calcul des positions actuelles
PL = -l1*sin(th1)+(le/2)*cos(th1)+l2*cos(-th1+th2)+l3*cos(-th1+th2-th3);
PR = -l1*sin(th1)-(le/2)*cos(th1)-l4*cos(-th1+th4)-l5*cos(-th1+th4-th5);

oc1 = -c1*sin(th1);
oc2 = -l1*sin(th1)+(le/2)*cos(th1)+c2*cos(-th1+th2);
ocL = -l1*sin(th1)+(le/2)*cos(th1)+l2*cos(-th1+th2)+c3*cos(-th1+th2-th3);
oc4 = -l1*sin(th1)-(le/2)*cos(th1)-c4*cos(-th1+th4);
ocR = -l1*sin(th1)-(le/2)*cos(th1)-l4*cos(-th1+th4)-c5*cos(-th1+th4-th5);
CdM = (m1*oc1+m2*oc2+m3*ocL+m4*oc4+m5*ocR)/(m1+m2+m3+m4+m5);

end
```