

Rapport TP/DM  
UE Estimation et Identification pour la Robotique

---

**Identification d'un robot plan RR en tant que rigide**

---

**Auteurs :**

Viviane Bao-3800857

Hao Yuan-21117163

Binôme 17

Année Universitaire 2023-2024

07 janvier 2024

# Table des matières

I.	Introduction-----	p2
II.	Modélisation du système mécanique-----	p3-p8
III.	Traitement des signaux-----	p9-p21
IV.	Conclusion-----	p21

## I. Introduction

En robotique, l'identification des caractéristiques essentielles d'un système est cruciale pour les contrôleurs basés sur des modèles. MATLAB est couramment utilisé pour cette tâche. Dans ce contexte, notre objectif est d'illustrer cette méthode en utilisant un bras robotique à deux degrés de liberté (RR orthogonal). Nous cherchons à identifier les propriétés inertielles et géométriques de ce bras en analysant les signaux d'angles et de couples articulaires générés lors d'un mouvement périodique le long d'une trajectoire donnée.

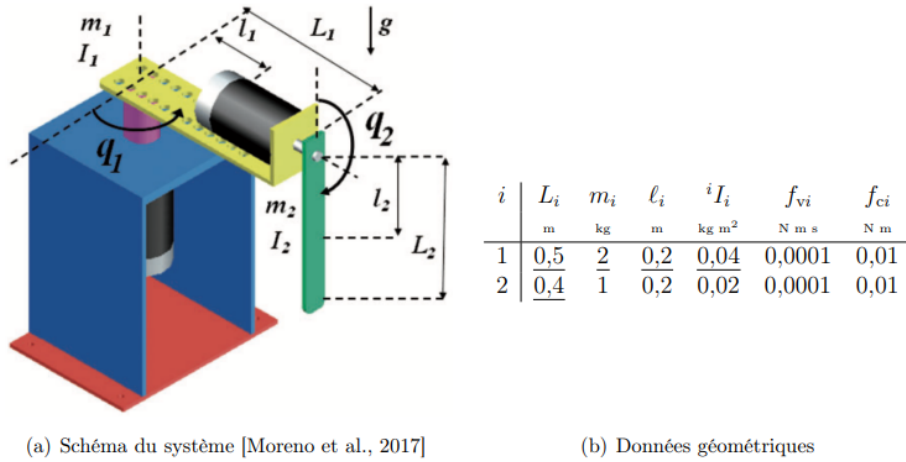


Figure 1- Robot RR orthogonal

Le système étudié est un robot plan composé de deux segments rigides, connectés par deux articulations motorisées, chacun des composants étant considéré comme parfaitement rigide. Nous possédons pour ce robot des informations de base sur la géométrie (longueurs  $L_i, i = 1 \dots 2$ ) et des approximations des caractéristiques inertielles (masses  $m_i$ , positions des centres de masses  $\ell_{ci}$ , moments d'inertie  $I_i$ ). De plus, les coefficients de frottement visqueux  $f_v$  et sec  $f_c$  sont connus. L'objectif est d'obtenir des valeurs plus précises des paramètres d'inertie et de frottement. Pour y parvenir, des trajectoires périodiques sont effectuées et nous enregistrons les signaux de position angulaire et de couple pour chaque articulation, représentées par  $q_i(t)$  et  $\tau_i(t)$  respectivement.

En analysant les signaux de position et de couple durant des mouvements répétitifs, nous obtenons des données reflétant la dynamique du robot. Ces données, traitées par des méthodes d'estimation paramétrique, précisent les propriétés inertielles et les frottements. Ces informations affinent le modèle du robot, crucial pour optimiser le contrôle et simuler avec exactitude son comportement, améliorant ainsi performance et fiabilité.

## II. Modélisation du système mécanique

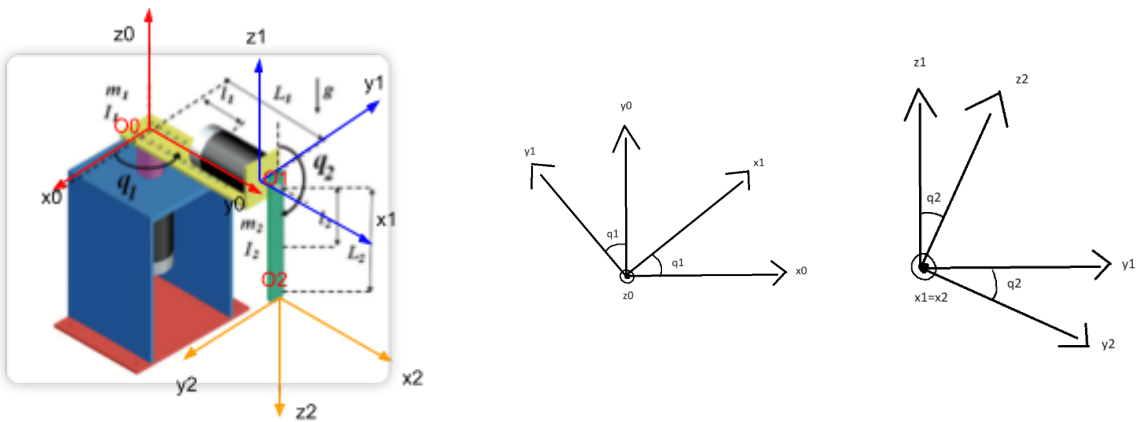
Le modèle dynamique du robot à deux degrés de liberté est décrit par les équations

d'Euler-Lagrange:  $M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + f_v \dot{q} + f_c(\dot{q}) + g(q) = \tau \quad (I)$

avec 
$$\left\{ \begin{array}{l} M(q) = \begin{bmatrix} I_1 + L_1^2 m_2 + l_1^2 m_1 + \frac{l_2^2 m_2}{2} (1 - \cos(2q_2)) & L_1 l_2 m_2 \cos(q_2) \\ L_1 l_2 m_2 \cos(q_2) & m_2 l_2^2 + I_2 \end{bmatrix} \\ C(q, \dot{q}) = l_2 m_2 \sin(q_2) \begin{bmatrix} l_2 \dot{q}_2 \cos(q_2) & l_2 \dot{q}_1 \cos(q_2) - L_1 \dot{q}_2 \\ -l_2 \dot{q}_1 \cos(q_2) & 0 \end{bmatrix} \end{array} \right.$$

où  $q, \dot{q}, \ddot{q} \in \mathbb{R}^2$  sont respectivement les vecteurs de position, de vitesse et d'accélération des articulations,  $M(q) \in \mathbb{R}^{2 \times 2}$  est la matrice d'inertie symétrique définie positive,  $C(q, \dot{q}) \in \mathbb{R}^2$  est le vecteur des couples de Coriolis et centrifuges,  $g(q) \in \mathbb{R}^2$  est le vecteur des couples gravitationnels,  $f_v \in \mathbb{R}^{2 \times 2}$  est une matrice diagonale qui contient les coefficients de frottement visqueux des articulations,  $f_c \in \mathbb{R}^2$  est une version continue du frottement de Coulomb à chaque articulation avec  $r > 0$  assez grand, et  $\tau \in \mathbb{R}^2$  est le vecteur des couples appliqués.

1/ Dans un premier temps, nous souhaitons définir les positions et les vitesses cartésiennes des points particuliers  $O_i$  et des centres de gravité  $C_i$  à partir des positions et des vitesses articulaires. Le système de référence a été configuré comme suit :



Nous rappelons que la position articulaire  $q_1(t)$  correspond à un alignement sur un axe horizontal, tandis que la position articulaire  $q_2(t)$  est en référence à un axe vertical et est considérée comme nulle lorsque le segment est en position verticale haute.

$$\text{Nous avons: } \begin{cases} \vec{x}_1 = \cos(q_1) \vec{x}_0 + \sin(q_1) \vec{y}_0 \\ \vec{y}_1 = -\sin(q_1) \vec{x}_0 + \cos(q_1) \vec{y}_0 \\ \vec{z}_1 = \vec{z}_0 \end{cases}$$

$$\text{Et } \begin{cases} \vec{y}_2 = \cos(q_2) \vec{y}_1 + \sin(q_2) \vec{z}_1 \\ \quad = -\sin(q_1) \cos(q_2) \vec{x}_0 + \cos(q_1) \cos(q_2) \vec{y}_0 + \sin(q_2) \vec{z}_0 \\ \vec{z}_2 = -\sin(q_2) \vec{y}_1 + \cos(q_2) \vec{z}_1 \\ \quad = \sin(q_1) \sin(q_2) \vec{x}_0 - \cos(q_1) \sin(q_2) \vec{y}_0 + \cos(q_2) \vec{z}_0 \\ \vec{x}_2 = \vec{x}_1 \end{cases}$$

Ceci aboutit aux positions et vitesses cartésiennes déterminées :

$$\vec{0_0 0_1} = L_1 \vec{x}_1 = L_1 (\cos(q_1) \vec{x}_0 + \sin(q_1) \vec{y}_0) = \begin{pmatrix} L_1 \cos(q_1) \\ L_1 \sin(q_1) \\ 0 \end{pmatrix}$$

$$\frac{d\vec{0_0 0_1}}{dt} = \begin{pmatrix} -L_1 \dot{q}_1 \sin(q_1) \\ L_1 \dot{q}_1 \cos(q_1) \\ 0 \end{pmatrix}$$

$$\overrightarrow{0_0 0_2} = \overrightarrow{0_0 0_1} + \overrightarrow{0_1 0_2} = \begin{pmatrix} L_1 \cos(q_1) \\ L_1 \sin(q_1) \\ 0 \end{pmatrix} + L_2 \vec{z}_2 = \begin{pmatrix} L_1 \cos(q_1) \\ L_1 \sin(q_1) \\ 0 \end{pmatrix} + L_2 \begin{pmatrix} \sin(q_1) \sin(q_2) \\ -\cos(q_1) \sin(q_2) \\ \cos(q_2) \end{pmatrix}$$

$$= \begin{pmatrix} L_1 \cos(q_1) + L_2 \sin(q_1) \sin(q_2) \\ L_1 \sin(q_1) - L_2 \cos(q_1) \sin(q_2) \\ L_2 \cos(q_2) \end{pmatrix}$$

$$\frac{d\overrightarrow{0_0 0_2}}{dt} = \begin{pmatrix} -L_1 \dot{q}_1 \sin(q_1) + L_2 \dot{q}_1 \cos(q_1) \sin(q_2) + L_2 \dot{q}_2 \sin(q_1) \cos(q_2) \\ L_1 \dot{q}_1 \cos(q_1) + L_2 \dot{q}_1 \sin(q_1) \sin(q_2) - L_2 \dot{q}_2 \cos(q_1) \cos(q_2) \\ -L_2 \dot{q}_2 \sin(q_2) \end{pmatrix}$$

Pour ce qui est des centres de gravité, nous obtenons :

$$\overrightarrow{0_0 C_1} = \begin{pmatrix} l_1 \cos(q_1) \\ l_1 \sin(q_1) \\ 0 \end{pmatrix}$$

$$\frac{d\overrightarrow{0_0 C_1}}{dt} = \begin{pmatrix} -l_1 \dot{q}_1 \sin(q_1) \\ l_1 \dot{q}_1 \cos(q_1) \\ 0 \end{pmatrix}$$

$$\overrightarrow{0_0 C_2} = \begin{pmatrix} L_1 \cos(q_1) + l_2 \sin(q_1) \sin(q_2) \\ L_1 \sin(q_1) - l_2 \cos(q_1) \sin(q_2) \\ l_2 \cos(q_2) \end{pmatrix}$$

$$\frac{d\overrightarrow{0_0 C_2}}{dt} = \begin{pmatrix} -L_1 \dot{q}_1 \sin(q_1) + l_2 \dot{q}_1 \cos(q_1) \sin(q_2) + l_2 \dot{q}_2 \sin(q_1) \cos(q_2) \\ L_1 \dot{q}_1 \cos(q_1) + l_2 \dot{q}_1 \sin(q_1) \sin(q_2) - l_2 \dot{q}_2 \cos(q_1) \cos(q_2) \\ -l_2 \dot{q}_2 \sin(q_2) \end{pmatrix}$$

2/ A présent, nous souhaitons formuler les équations de mouvement reliant les positions, les vitesses et accélérations articulaires aux couples exercés par la motorisation. Il est donc nécessaire de formuler l'expression de l'énergie potentielle du système afin d'en déduire ensuite l'expression de  $g(q)$ .

$$E_p(totale) = E_p(S_1) + E_p(S_2)$$

$$E_p(S_1) = \text{constante}, E_p(S_2) = m_2 g l_2 \cos(q_2) + \text{constante}$$

$$\text{Ainsi, } E_p(totale) = m_2 g l_2 \cos(q_2) + \text{constante}$$

$$\text{Nous pouvons donc déduire que: } g(q) = \left( \frac{\partial E_p}{\partial q} \right)^T = \begin{pmatrix} 0 \\ -m_2 g l_2 \sin(q_2) \end{pmatrix}$$

Pour déplacer un joint robotique, le couple généré par le moteur doit être suffisant pour gérer à la fois la tâche externe ( $\tau_{load}$ ) et les forces de frottement internes au système ( $f_v \omega$ ,  $f_c \text{sign}(\omega)$ ). L'expression générale du couple total appliqué sur le joint est donnée par:

$$\tau = \tau_{load} + f_v \omega + f_c \text{sign}(\omega).$$

$$\text{Au final: } \begin{bmatrix} I_1 + L_1^2 m_2 + l_1^2 m_1 + \frac{l_2^2 m_2}{2} (1 - \cos(2q_2)) & L_1 l_2 m_2 \cos(q_2) \\ L_1 l_2 m_2 \cos(q_2) & m_2 l_2^2 + I_2 \end{bmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} +$$

$$l_2 m_2 \sin(q_2) \begin{bmatrix} l_2 \dot{q}_2 \cos(q_2) & l_2 \dot{q}_1 \cos(q_2) - L_1 \dot{q}_2 \\ -l_2 \dot{q}_1 \cos(q_2) & 0 \end{bmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} + \begin{pmatrix} 0 \\ -m_2 g l_2 \sin(q_2) \end{pmatrix} = \begin{pmatrix} \tau_1 - f_{v1} \dot{q}_1 - f_{c1} \text{sign}(\dot{q}_1) \\ \tau_2 - f_{v2} \dot{q}_2 - f_{c2} \text{sign}(\dot{q}_2) \end{pmatrix}$$

$$\text{Soit: } \begin{bmatrix} I_1 + L_1^2 m_2 + l_1^2 m_1 + \frac{l_2^2 m_2}{2} (1 - \cos(2q_2)) & L_1 l_2 m_2 \cos(q_2) \\ L_1 l_2 m_2 \cos(q_2) & m_2 l_2^2 + I_2 \end{bmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} +$$

$$l_2 m_2 \sin(q_2) \begin{bmatrix} l_2 \dot{q}_2 \cos(q_2) & l_2 \dot{q}_1 \cos(q_2) - L_1 \dot{q}_2 \\ -l_2 \dot{q}_1 \cos(q_2) & 0 \end{bmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{v1} & 0 \\ 0 & f_{v2} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} + \begin{pmatrix} f_{c1} \tanh(r \dot{q}_1) \\ f_{c2} \tanh(r \dot{q}_2) \end{pmatrix} +$$

$$\begin{pmatrix} 0 \\ -m_2 g l_2 \sin(q_2) \end{pmatrix} = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix}$$

Nous pouvons poser:

Paramètre	Définition
$\theta_1$	$I_1 + m_1 l_1^2 + m_2 L_1^2$
$\theta_2$	$l_2^2 m_2$
$\theta_3$	$L_1 l_2 m_2$
$\theta_4$	$m_2 l_2^2 + I_2$
$\theta_5$	$m_2 g l_2$
$\theta_6$	$f_{v1}$
$\theta_7$	$f_{v2}$
$\theta_8$	$f_{c1}$
$\theta_9$	$f_{c2}$

De plus, il convient de noter les propriétés suivantes :  $\frac{1-\cos(2q_2)}{2} = \sin^2(q_2)$ ,  
 $\sin(q_2)\cos(q_2) = \frac{1}{2}\sin(2q_2)$

Par conséquent, le système dynamique peut être reformulé comme suit :

$$\begin{pmatrix} \theta_1 + \theta_2 \sin^2(q_2) & \theta_3 \cos(q_2) \\ \theta_3 \cos(q_2) & \theta_4 \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} + \begin{pmatrix} \frac{1}{2}\theta_2 \dot{q}_2 \sin(2q_2) & \frac{1}{2}\theta_2 \dot{q}_1 \sin(2q_2) - \theta_3 \dot{q}_2 \sin(q_2) \\ -\frac{1}{2}\theta_2 \dot{q}_1 \sin(2q_2) & 0 \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} +$$

$$\begin{pmatrix} \theta_6 & 0 \\ 0 & \theta_7 \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} + \begin{pmatrix} \theta_8 \tanh(r\dot{q}_1) \\ \theta_9 \tanh(r\dot{q}_2) \end{pmatrix} + \begin{pmatrix} 0 \\ -\theta_5 \sin(q_2) \end{pmatrix} = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix}$$

3/ Dans cette partie, nous voulons devons exprimer les inconnues à identifier (masses, moments quadratiques, ...) de manière à formuler l'opérateur de régression. Le système défini précédemment peut être réécrit comme suit :



$$\begin{pmatrix} \theta_1 \ddot{q}_1 + \theta_2 \sin(q_2)^2 \ddot{q}_1 + \theta_3 \cos(q_2) \ddot{q}_2 + \frac{1}{2} \theta_2 \dot{q}_2 \sin(2q_2) \dot{q}_1 - \theta_3 \dot{q}_2^2 \sin(q_2) + \frac{1}{2} \theta_2 \dot{q}_1 \sin(2q_2) \dot{q}_2 + \theta_6 \dot{q}_1 + \theta_8 \tanh(r \dot{q}_1) \\ \theta_3 \cos(q_2) \ddot{q}_1 + \theta_4 \ddot{q}_2 - \frac{1}{2} \theta_2 \dot{q}_1 \sin(2q_2) \dot{q}_1 - \theta_5 \sin(q_2) + \theta_7 \dot{q}_2 + \theta_9 \tanh(r \dot{q}_2) \end{pmatrix} = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix}$$

$$\begin{pmatrix} \ddot{q}_1 & \sin(q_2)^2 \ddot{q}_1 + \dot{q}_1 \dot{q}_2 \sin(2q_2) & \cos(q_2) \ddot{q}_2 - \dot{q}_2^2 \sin(q_2) & 0 & 0 & \dot{q}_1 & 0 & \tanh(r \dot{q}_1) & 0 \\ 0 & -\frac{1}{2} \dot{q}_1^2 \sin(q_2) & \cos(q_2) \ddot{q}_1 & \ddot{q}_2 - \sin(q_2) & 0 & \dot{q}_2 & 0 & \tanh(r \dot{q}_2) \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \end{pmatrix} = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix}$$

Le modèle dynamique du manipulateur robotique à deux degrés de liberté dans l'équation **(1)** satisfait la propriété de linéarité dans les paramètres; donc, il peut être exprimé comme

un modèle de régression linéaire:  $Y(q, \dot{q}, \ddot{q}) \theta = \tau$

où  $Y(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{2 \times 9}$  est la matrice de régression,  $\theta \in \mathbb{R}^9$  est le vecteur des paramètres.

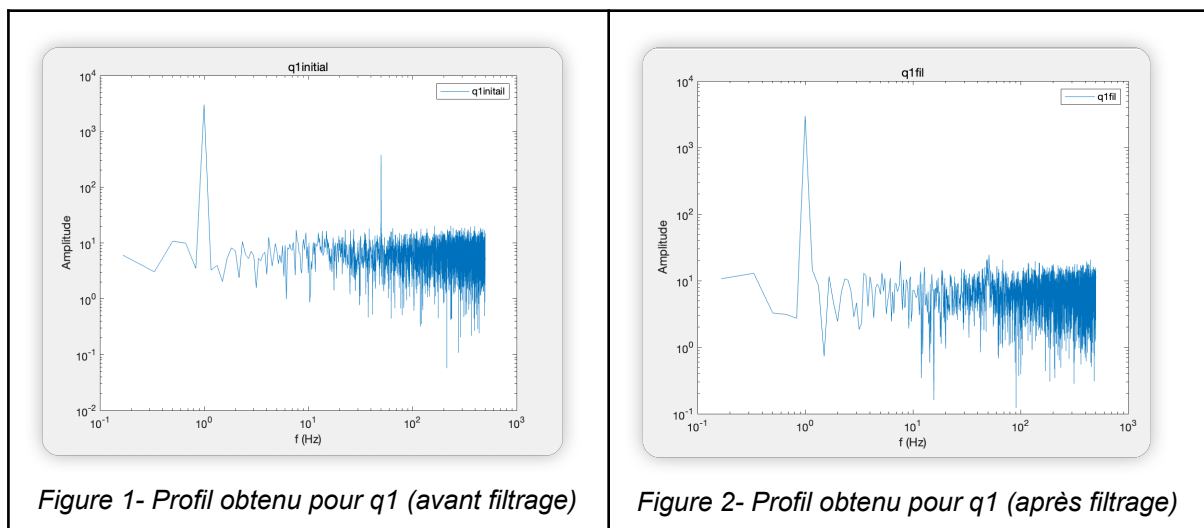
### III. Traitement des signaux

Le traitement des signaux récupérés par le robot nécessite des interfaces spécifiques pour l'enregistrement des mesures d'angles et de couples. La capture des données angulaires se réalise à une fréquence d'échantillonnage de 1 kHz, tandis que celle des couples se fait à 2.5 kHz. Un décalage temporel fixe est à noter entre le début de l'acquisition des données de couples et d'angles. Par ailleurs, divers types de bruits affectent la précision des mesures expérimentales :

- Un bruit de mesure, assimilable à du bruit blanc gaussien, interfère avec les mesures de couples et d'angles.
- Un bruit parasite est également induit par le réseau électrique à 50 Hz.

Pour pallier ces contraintes, il est nécessaire d'entreprendre les démarches suivantes:

1/ Nous avons appliqué un filtre coupe-bande pour enlever au maximum le bruit parasite. Ce filtre est conçu pour atténuer les fréquences autour de 50 Hz. La largeur de bande ( $bw$ ) du filtre est déterminée par  $f_0/Q$ , ce qui donne la plage de fréquences à atténuer autour de la fréquence centrale. Les fréquences normalisées ( $Wn_q$  et  $Wn_{tau}$ ) sont calculées en fonction des fréquences d'échantillonnage  $fs_1$  et  $fs_2$ . Le filtre est ensuite appliqué aux quatre signaux ( $q_1$ ,  $q_2$ ,  $\tau_1$ ,  $\tau_2$ ) en utilisant la fonction `filter` de MATLAB, qui filtre les données avec une réponse en fréquence déterminée par les coefficients  $[b, a]$ . Enfin, la fonction `fft_plot_log` est utilisée pour afficher le spectre fréquentiel des 4 signaux.



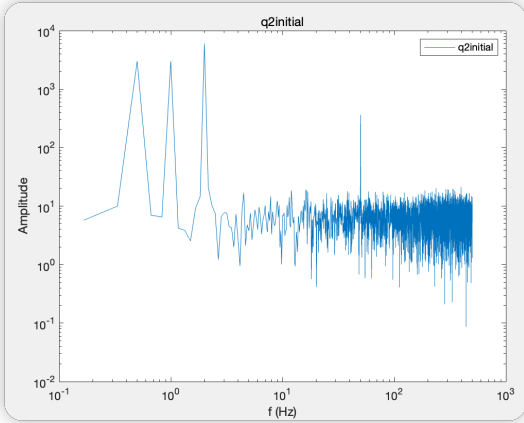


Figure 3- Profil obtenu pour  $q_2$  (avant filtrage)

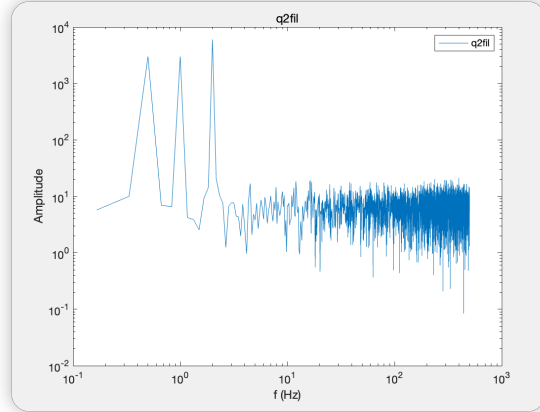


Figure 4- Profil obtenu pour  $q_2$  (après filtrage)

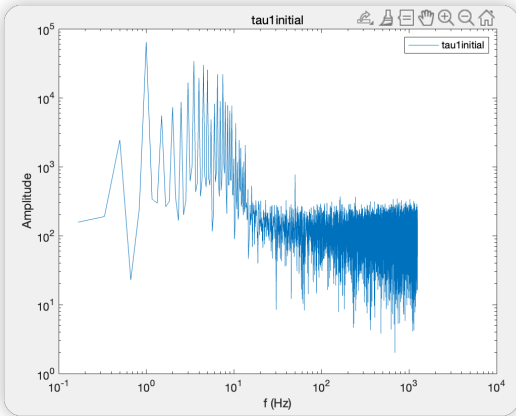


Figure 5- Profil obtenu pour  $\tau_1$  (avant filtrage)

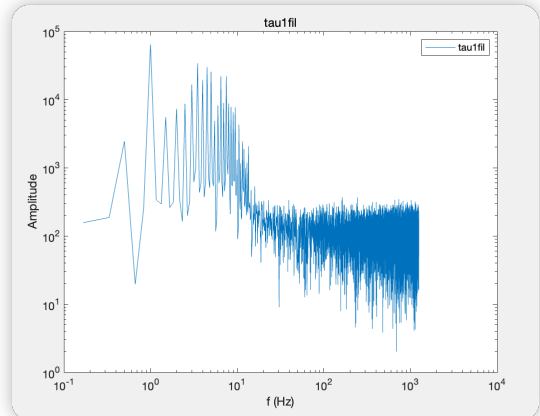


Figure 6- Profil obtenu pour  $\tau_1$  (après filtrage)

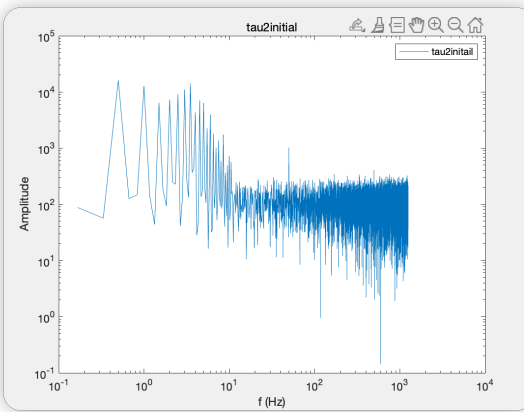


Figure 7- Profil obtenu pour  $\tau_2$  (avant filtrage)

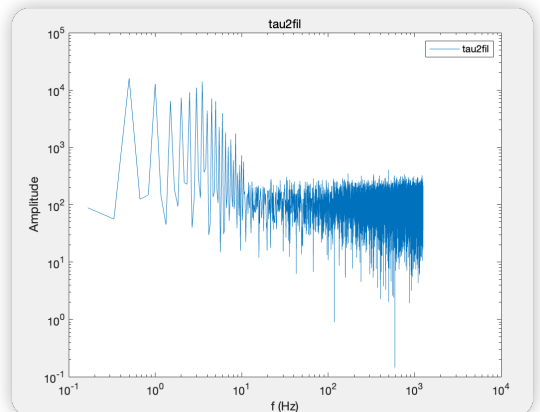


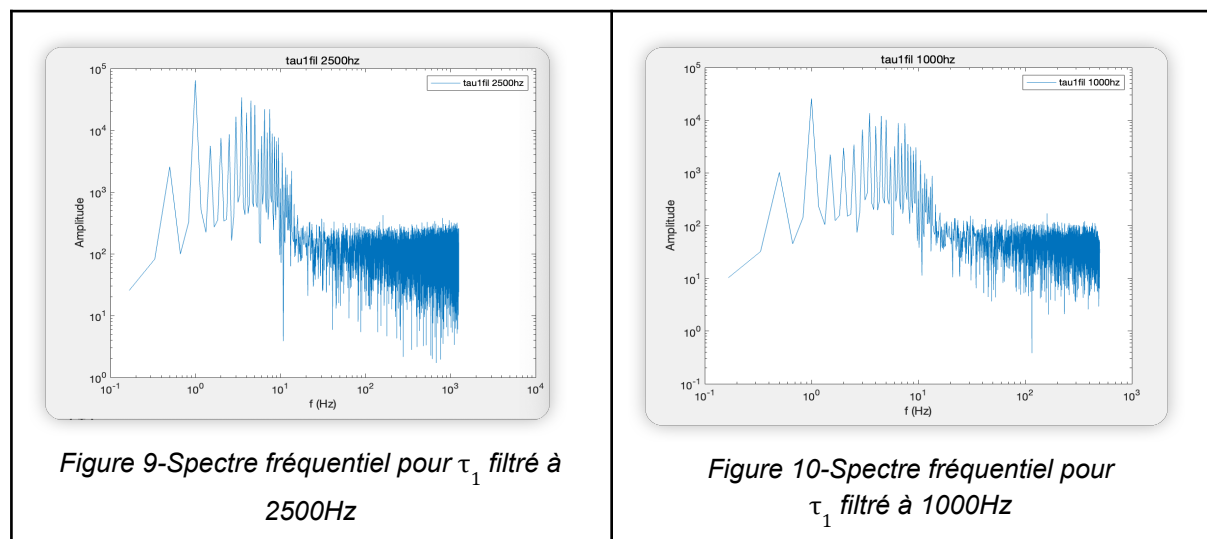
Figure 8- Profil obtenu pour  $\tau_2$  (après filtrage)

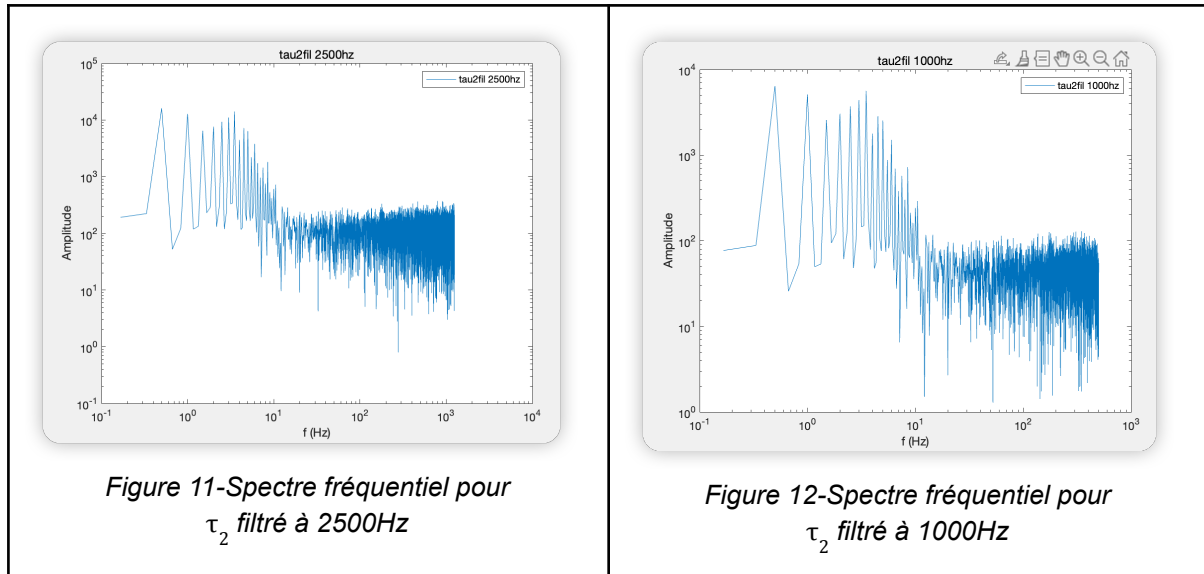
Les figures 1 et 2 montrent un avant et après filtrage. Après filtrage, nous voyons une atténuation claire autour de 50 Hz, comme indiqué par une réduction de l'amplitude dans cette région fréquentielle. Les figures 3 et 4 montrent des résultats similaires, avec une atténuation des fréquences autour de 50 Hz. Les figures 5 et 6 et les figures 7 et 8 montrent également les signaux avant et après filtrage, avec une réduction dans la région de fréquence de 50 Hz.

Les pics de fréquence à 50 Hz dans les spectres de fréquence initiaux ont été significativement atténués dans les spectres filtrés, ce qui montre l'efficacité du filtre utilisé. Par ailleurs, les graphiques après filtrage présentent moins d'énergie (moins d'amplitude) dans la plage de fréquences ciblée par le filtre. La qualité du filtrage peut être appréciée par l'absence de modifications significatives en dehors de la plage ciblée, indiquant que le filtre n'a pas affecté les fréquences en dehors de la bande d'atténuation.

2/ Nous proposons ensuite une méthode pour que les signaux de couple et d'angle soient à la même fréquence d'échantillonnage. Pour ce faire, nous avons utilisé la fonction 'resample' pour rééchantillonner les signaux filtrés  $\tau_{1\_fil}$  et  $\tau_{2\_fil}$  afin qu'ils aient la même fréquence d'échantillonnage. Cette fonction gère automatiquement le rééchantillonnage lorsque les taux de suréchantillonnage et de sous-échantillonnage ne sont pas entiers, et elle intègre un filtre passe-bas pour prévenir l'aliasing, assurant ainsi la conservation de l'intégrité du spectre du signal.

Le fréquence d'échantillonnage pour l'angle est de 1000Hz, tandis que celle pour le couple est de 2500Hz. Nous avons deux options: augmenter la fréquence d'échantillonnage de l'angle ou réduire celle du couple. Nous avons opté pour la réduction de la fréquence d'échantillonnage du couple. Ainsi, le code que nous avons développé pour cette partie change la fréquence d'échantillonnage de 2500 Hz à 1000 Hz en utilisant un facteur de suréchantillonnage de 2 et un facteur de sous-échantillonnage de 5. Ceci est une technique commune pour convertir un signal d'une fréquence d'échantillonnage à une autre. Enfin, la fonction 'fft\_plot\_log' permet de visualiser les résultats et de comparer visuellement les spectres de fréquences avant et après rééchantillonnage pour évaluer l'impact de l'opération sur les signaux.





En comparant les figures 9 et 10, nous pouvons évaluer l'impact du rééchantillonnage sur 'tau1'. De même, les figures 11 et 12 permettent cette évaluation pour 'tau2'.

Le spectre sur la figure 9 montre l'ensemble des composantes fréquentielles du signal. Les pics dans ce spectre représentent les fréquences dominantes ou les harmoniques du signal. Le bruit blanc gaussien est également visible. Après rééchantillonnage (Figure 10), en réduisant la fréquence d'échantillonnage à 1000 Hz, nous nous attendons à ce que le signal conserve ses caractéristiques principales, mais avec une résolution fréquentielle plus faible. Ici, le rééchantillonnage est bien réalisé, il n'y a pas de distorsion notable ou d'introduction d'aliasing. Le spectre est plus « propre », avec moins de bruit de haute fréquence, grâce à l'effet de lissage du rééchantillonnage.

Le spectre sur la figure 11 présente des caractéristiques similaires à celles de la figure 9, avec ses propres pics fréquentiels et niveaux de bruit. Après rééchantillonnage (Figure 12), le signal tau2 rééchantillonné à 1000 Hz montre une diminution des hautes fréquences en raison du sous-échantillonnage (cela peut être bénéfique pour atténuer le bruit de haute fréquence, mais il y a aussi un risque de perdre des informations si des composantes significatives du signal se trouvent dans les fréquences qui sont réduites).

3/ Pour estimer le décalage temporel entre les deux interfaces d'acquisition, nous devons commencer par calculer la corrélation: nous combinons les signaux filtrés 'q1\_fil' et 'q2\_fil' dans un vecteur 'combined\_q' (ce qui implique que nous les traitons en un seul ensemble de données). De même, le vecteur 'combined\_tau' est créé en empilant 'tau1\_resample' et 'tau2\_resample'. Ensuite, nous utilisons la fonction xcorr pour calculer la corrélation croisée des deux groupes de signaux (cette fonction évalue comment les deux ensembles de signaux se correspondent l'un à l'autre en introduisant des décalages temporels successifs entre eux, le résultat de xcorr est un vecteur corrélation qui contient les coefficients de corrélation pour chaque décalage et un vecteur lags qui indique le décalage correspondant à chaque coefficient). Nous pouvons déterminer le décalage temporel via 'time\_delay = lags(l);', l'indice 'l' du maximum absolu de la corrélation, indique le décalage où la similitude

est la plus élevée. Cela signifie que si nous décalons l'un des ensembles de signaux par 'time\_delay = lags(l);', ils seront alignés dans le temps. Ce décalage est converti en secondes en le divisant par la fréquence d'échantillonnage, supposée ici être de 1000 Hz.

```
time delay before the signal moves
434
```

Après avoir détecté un retard de 400 unités (ou échantillons), une correction est appliquée en supprimant les 400 premiers échantillons des signaux q1\_fil et q2\_fil. Cette opération vise à synchroniser les signaux q avec les signaux de couple tau. Une fois cette correction effectuée, la corrélation croisée est recalculée pour vérifier que les signaux sont maintenant mieux alignés en termes de temps.

```
time delay after the signal moves
0
```

4/ Pour calculer les vitesses ('q1\_dot','q2\_dot') et accélérations angulaires ('q1\_dotdot','q2\_dotdot') à partir des signaux de position ('q1\_fil','q2\_fil'), nous avons utilisé une méthode de différentiation numérique.

Chaque signal de position est d'abord filtré par un filtre Butterworth à l'aide de la fonction 'filtfilt' pour atténuer le bruit de haute fréquence, ce qui est essentiel avant la différenciation afin de ne pas amplifier le bruit.

Les signaux filtrés sont ensuite différenciés numériquement en utilisant la fonction 'diff' pour obtenir les vitesses angulaires. Cela consiste à calculer la différence entre des échantillons consécutifs et à diviser par l'intervalle de temps entre les échantillons pour obtenir la vitesse instantanée.

Après la première différentiation, un autre filtre Butterworth est appliqué pour lisser les vitesses angulaires et réduire le bruit qui aurait pu être introduit par la différenciation.

Les vitesses filtrées sont différenciées une seconde fois pour obtenir les accélérations angulaires.

Un dernier filtrage est appliqué aux accélérations angulaires pour minimiser davantage l'impact du bruit amplifié par la seconde différenciation.

Cette méthode permet de calculer les dérivées temporelles des signaux de position tout en minimisant l'effet du bruit grâce à des étapes de filtrage répétées. Le processus de filtrage avant et après chaque différenciation s'assure que le signal résultant est le plus lisse possible tout en étant fidèle aux changements dynamiques du signal d'origine.

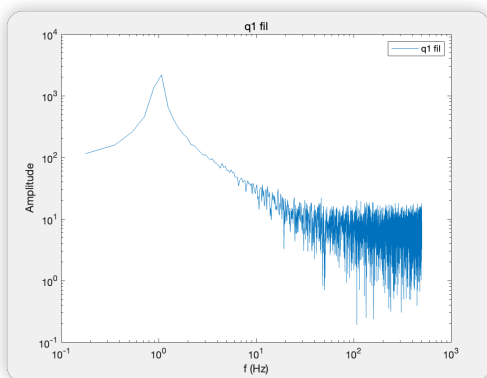


Figure 13-Profil pour  $q1\_fil$

Voici le spectre après suppression du bruit électrique. Nous observons que les informations principales du signal résident dans les basses fréquences, alors que le bruit blanc gaussien est principalement distribué dans les hautes fréquences, qui sont plus marquées. La fréquence de coupure de notre filtre est fixée à 2 Hz pour conserver les informations essentielles et éliminer le bruit blanc gaussien.

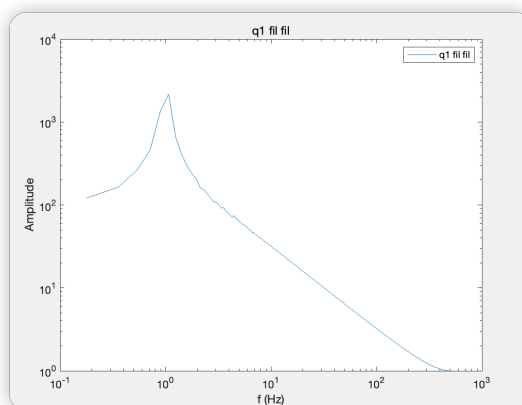


Figure 14- Profil pour  $q1\_fil\_fil$  (après filtrage de type Butterworth)

Voici le spectre après suppression du bruit parasite et du bruit blanc gaussien.

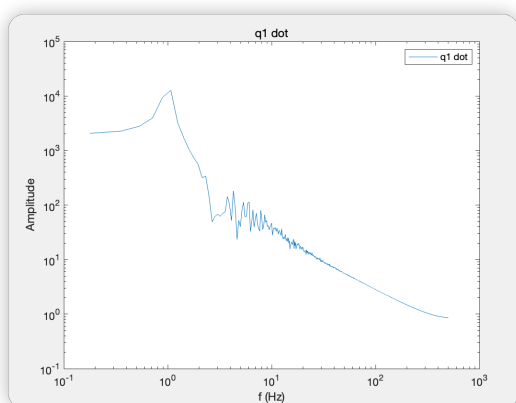


Figure 15- Profil pour  $q1\_dot$

Malgré le filtrage du bruit haute fréquence, une fraction résiduelle qui n'a pas été entièrement éliminée a été amplifiée lors de la différenciation. Néanmoins, grâce à la réduction préalable de l'amplitude du bruit haute fréquence, l'amplification subséquente du bruit n'a pas altéré la qualité du signal.

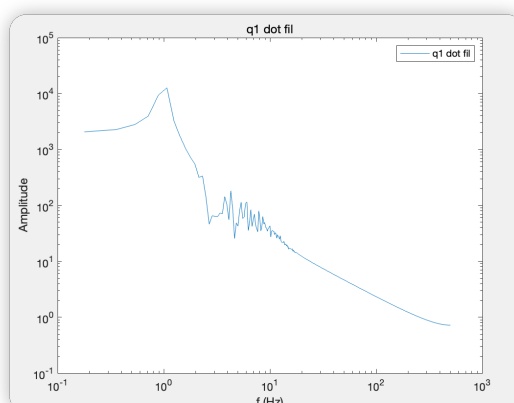


Figure 16- Profil pour  $q1\_dot\_fil$  (après filtrage de type Butterworth)

Après le processus de débruitage, nous obtenons la vitesse angulaire avec une atténuation de l'amplitude des composantes haute fréquence. De manière similaire, en différenciant la vitesse angulaire débruitée, nous obtenons le spectre de l'accélération sur la figure 17.

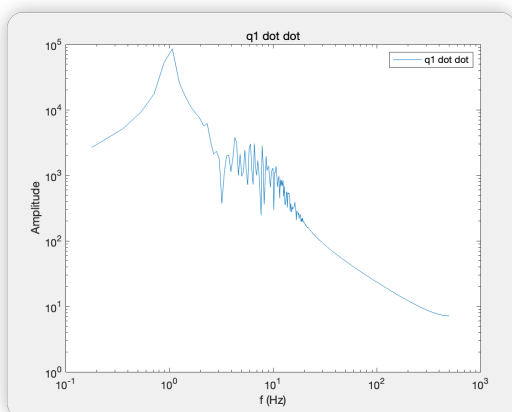


Figure 17- Profil pour  $q1\_dotdot$

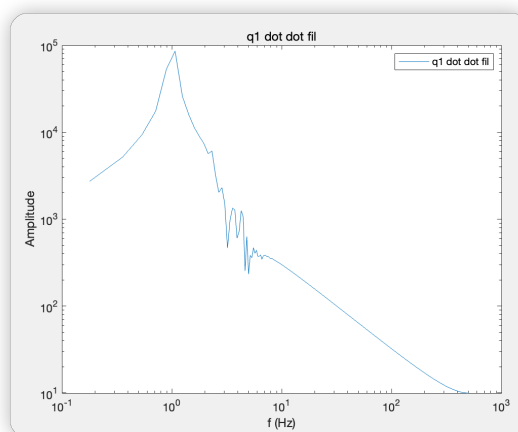


Figure 18- Profil pour  $q1\_dotdot\_fil$  (après filtrage)

Nous avons constaté que le signal de haute fréquence dans le spectre de l'accélération avait un impact mineur sur la qualité du signal, ce qui nous a conduit à appliquer un processus de débruitage à l'accélération également.

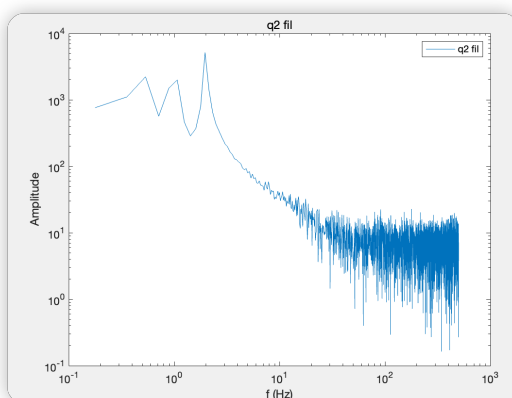


Figure 19-Profil pour  $q2\_fil$

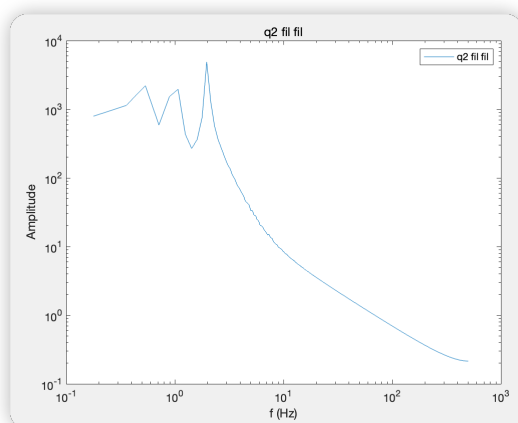


Figure 20- Profil pour  $q2\_fil\_fil$  (après filtrage de type Butterworth)

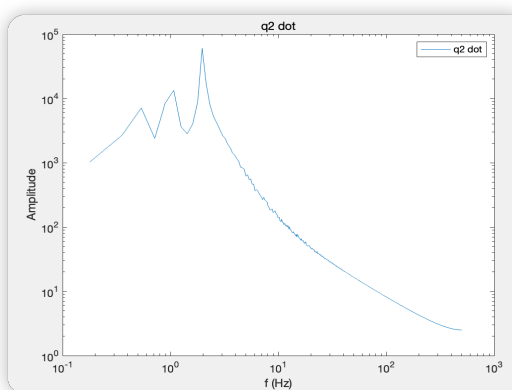


Figure 21- Profil pour  $q2\_dot$

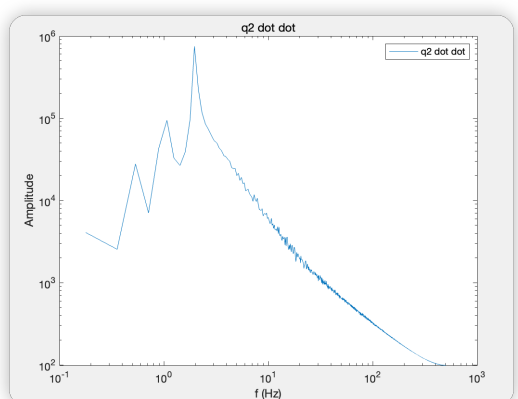


Figure 22- Profil pour  $q2\_dotdot$



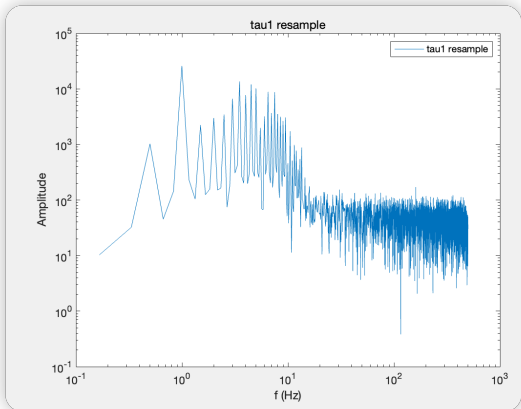


Figure 23- Profil pour tau1\_resample

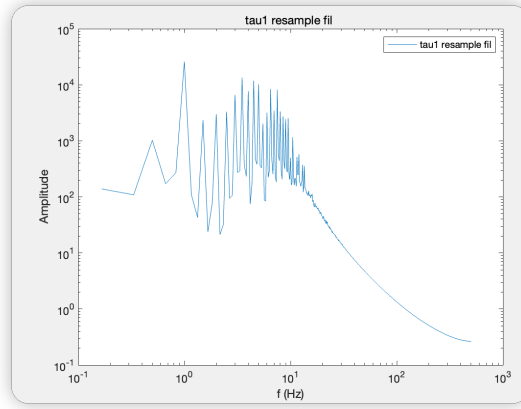


Figure 24- Profil pour tau1\_resample\_fil

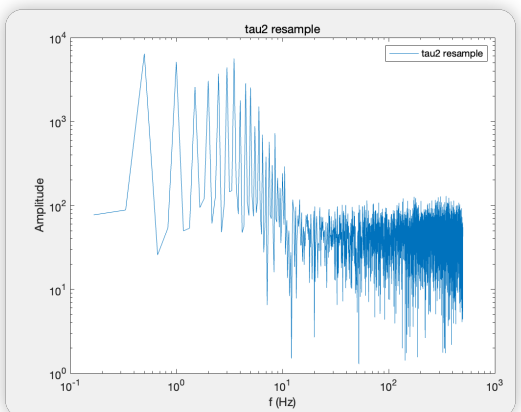


Figure 24- Profil pour tau2\_resample

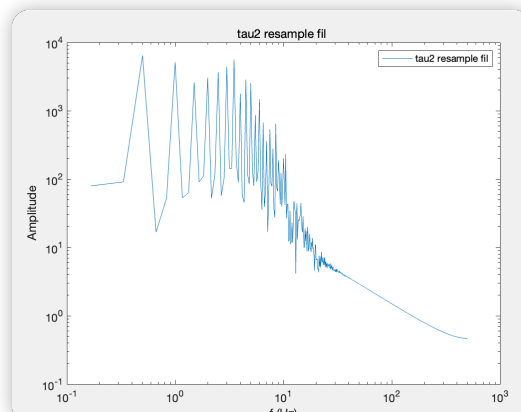


Figure 25- Profil pour tau2\_resample\_fil

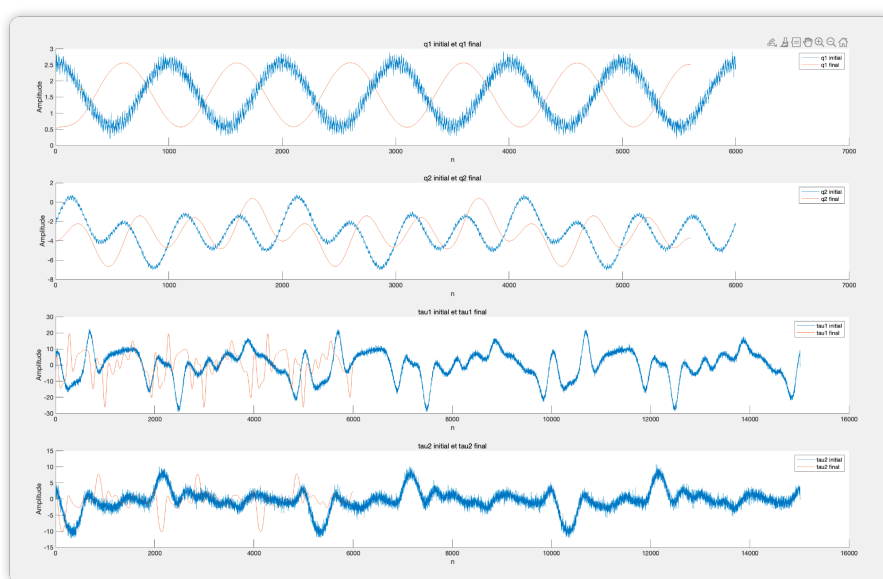
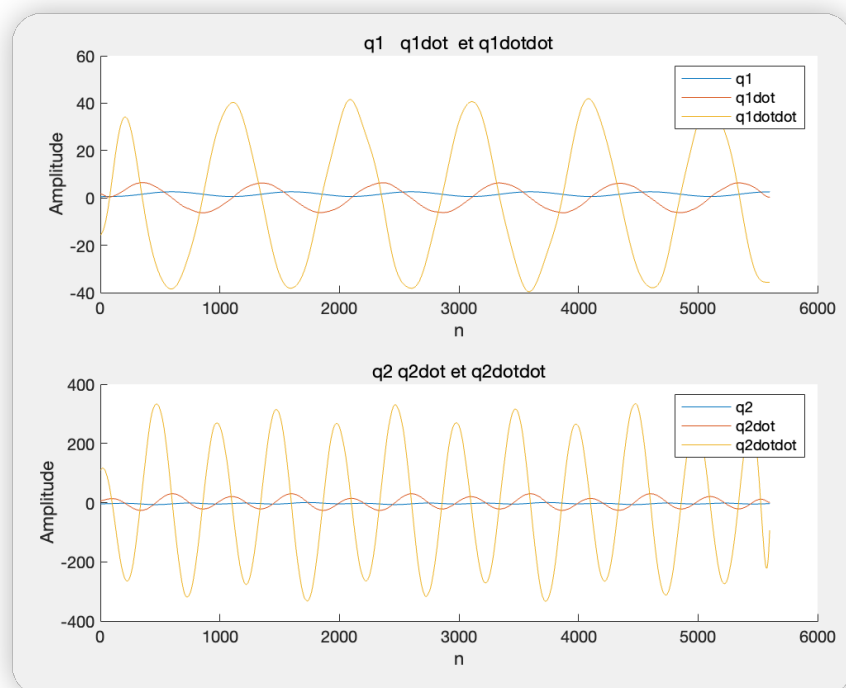


Figure 26-Résultat final du traitement

La figure 26 représente le résultat final de notre traitement. Nous avons d'abord appliqué un processus de réduction du bruit parasite et du bruit gaussien aux données d'angle, puis nous avons tronqué les 400 premiers points de données afin d'aligner temporellement les deux interfaces d'échantillonnage. Ensuite, nous avons appliqué un traitement similaire de réduction du bruit aux données de couple, suivi d'une opération de sous-échantillonnage avec un facteur de 2,5. En fin de compte, nous avons réussi à éliminer efficacement le bruit des données d'angle et nous avons constaté un décalage vers la gauche dans le signal final pour compenser le décalage temporel entre les interfaces d'échantillonnage. Les données finales de couple ont été comprimées, résultant en une réduction de la résolution due à la sous-échantillonnage, tout en préservant une élimination réussie du bruit.



Nous avons représenté graphiquement les données relatives aux angles, aux vitesses angulaires et aux accélérations angulaires de chaque articulation. L'une de nos observations significatives a été que les amplitudes des vitesses angulaires sont considérablement supérieures à celles des angles correspondants. Cette disparité s'explique par le fait que nous ne pouvons pas simplement interpréter les résultats de la dérivation des angles comme des vitesses angulaires directes. Pour obtenir des valeurs de vitesses angulaires véritablement précises, nous avons dû normaliser ces résultats en les divisant par l'intervalle d'échantillonnage, ce qui donne  $q\_dot = \text{diff}(q)/0,001$ . Le même processus a été appliqué pour calculer les accélérations angulaires, soit  $q\_dotdot = \text{diff}(q\_dot)/0,001$ . La normalisation par l'intervalle d'échantillonnage pour le calcul des vitesses et accélérations est une pratique standard pour obtenir des valeurs précises lors de la différenciation numérique. Cela permet

de compenser le fait que la fonction `diff` calcule seulement la différence entre les échantillons successifs sans tenir compte de l'intervalle de temps qui les sépare.

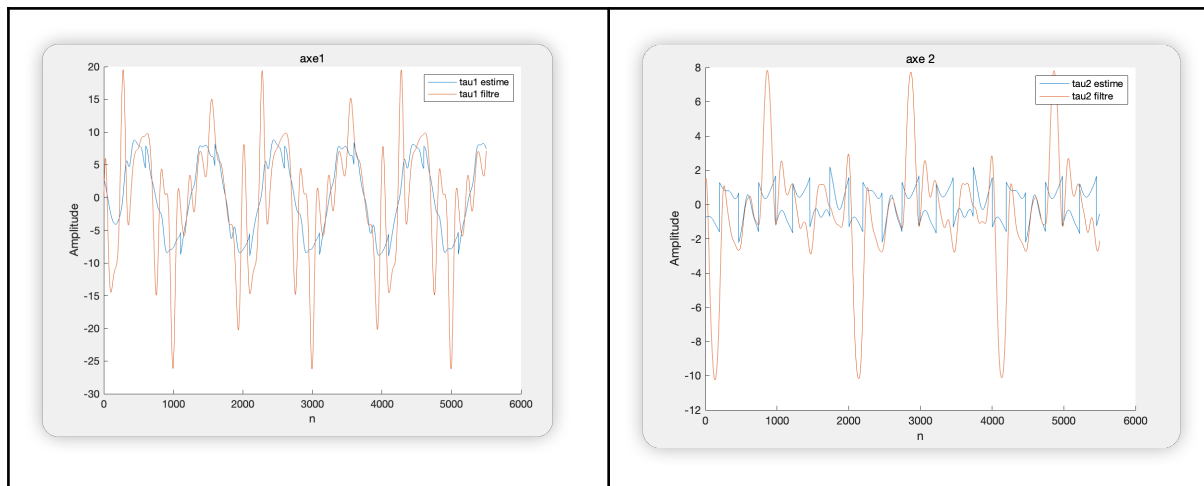
Après le traitement des données, nous disposons des données estimées de tous les paramètres  $[q1, q1dot, q1dotdot]$   $[q2, q2dot, q2dotdot]$   $\tau_1$  et  $\tau_2$ .

Nous avons d'abord envisagé d'utiliser un estimateur linéaire des moindres carrés sans aucune contrainte. Le choix découle de plusieurs considérations. Tout d'abord, notre problème est essentiellement linéaire et implique un bruit blanc gaussien. De plus, notre modèle n'est pas conçu pour fonctionner en temps réel, ce qui signifie que les données ne sont pas mises à jour en temps réel. Au lieu de cela, nous avons collecté toutes les données avant de procéder à l'estimation, ce qui rend inutile l'utilisation d'un filtre de Kalman par exemple.

Le principe de cette méthode consiste à construire une matrice augmentée en fonction du modèle linéaire  $Y\theta = b$ . Chaque jeu de données  $Y$  et  $b$  ajoute deux lignes à la matrice augmentée. Une fois que toutes les données sont introduites dans notre modèle, théoriquement, le modèle contient déjà les informations sur les paramètres que nous souhaitons estimer. Notre seule tâche est d'extraire explicitement les informations des paramètres du modèle, ce qui implique l'utilisation de la pseudo-inverse de Moore-Penrose.

Voici les résultats que nous avons obtenus :

```
>> estimation_lms
resultat lms sans contrainte
-0.1741
 0.0113
 0.0002
-0.0016
-0.7580
 1.1377
 0.0459
-1.7127
-1.4580
```



Cependant, nous attendions les résultats suivants :

$$\theta = [0.37 \ 0.04 \ 0.1 \ 0.06 \ 1.962 \ 0.0001 \ 0.0001 \ 0.01 \ 0.01]$$

Nous pouvons remarquer que les résultats obtenus par l'estimateur comportent des valeurs négatives, ce qui est clairement peu réaliste, car selon l'expression de  $\theta$ , toutes les valeurs estimées devraient théoriquement être non négatives. Pour remédier à cela, nous avons cherché à utiliser un algorithme d'optimisation qui permettrait d'imposer des contraintes sur les paramètres estimés. Dans la boîte à outils d'optimisation de MATLAB, il existe de nombreux algorithmes. Ces algorithmes fonctionnent selon des principes similaires : tout d'abord, ils construisent une fonction de coût. Dans notre cas, nous définissons la fonction de coût comme  $F_1 = \text{norm}(Y\theta - b)$ , où  $Y$  et  $b$  sont des matrices augmentées contenant toutes les valeurs de données. Pour 6000 points de données,  $Y$  et  $b$  contiennent 12000 lignes.

En réalité, pour garantir strictement que les valeurs estimées des paramètres soient positives, nous avons deux méthodes :

- Nous appliquons une transformation exponentielle aux paramètres  $\theta$ , ce qui est indiqué par  $\exp(\theta)$ . Cette transformation assure que les valeurs estimées pour  $\theta$ , nommées  $r_i$ , seront toujours positives, car l'exponentielle d'un nombre réel est toujours positive. En définissant la fonction de coût comme  $F_2 = \text{norm}(Y\exp(\theta) - b)$ , nous créons un problème d'optimisation où le but est de minimiser la norme (souvent la norme euclidienne) de la différence entre les valeurs prédites par le modèle  $Y\exp(\theta)$  et les données observées  $b$ . En trouvant le minimum de cette fonction de coût grâce à un algorithme d'optimisation, nous obtenons les valeurs des paramètres transformés  $r_i$ , qui correspondent à  $\exp(\theta_i)$ .

Pour retrouver les paramètres d'origine  $\theta$ , nous prenons le logarithme naturel des paramètres transformés  $r_i$ .

- Nous utilisons une fonction intégrée à MATLAB pour définir des limites.

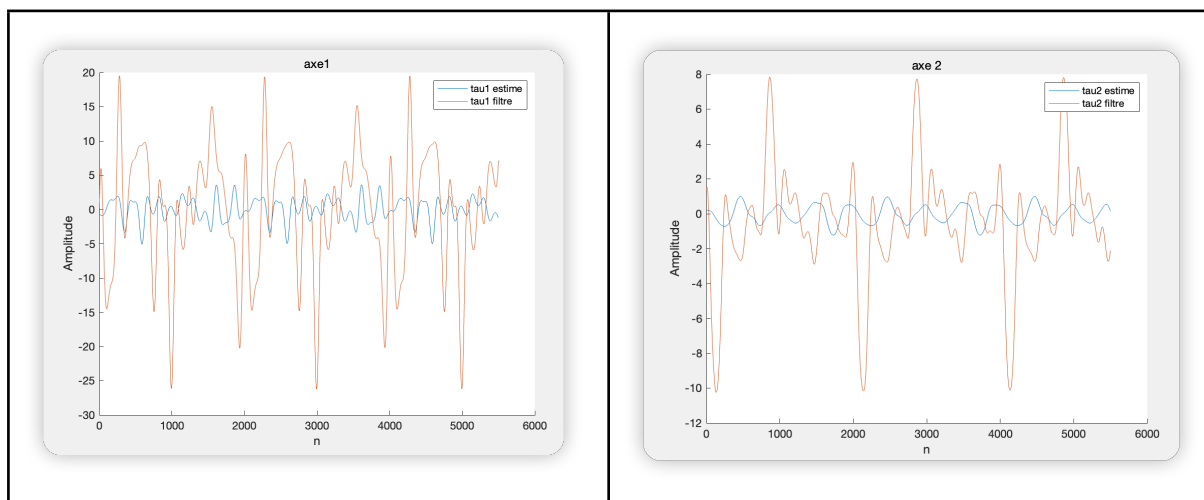
Nous avons utilisé la fonction `lsqnonlin` de la boîte à outils d'optimisation de MATLAB. La fonction `lsqnonlin` de MATLAB fait partie de la boîte à outils d'optimisation (Optimization Toolbox) et est utilisée pour résoudre des problèmes de moindres carrés non linéaires.

Nous commençons par fixer un vecteur initial `x0` et définissons des bornes inférieures `lb` et supérieures `ub`. L'exécution de la commande :

`[x, resnorm, residual, exitflag, output] = lsqnonlin(@function_cout_optimization, x0, lb, ub, options);` lance l'algorithme qui, par le biais de techniques telles que la descente de gradient ou d'autres approches algorithmiques, recherche les valeurs des paramètres qui réduisent au minimum la fonction de coût tout en respectant les contraintes imposées.

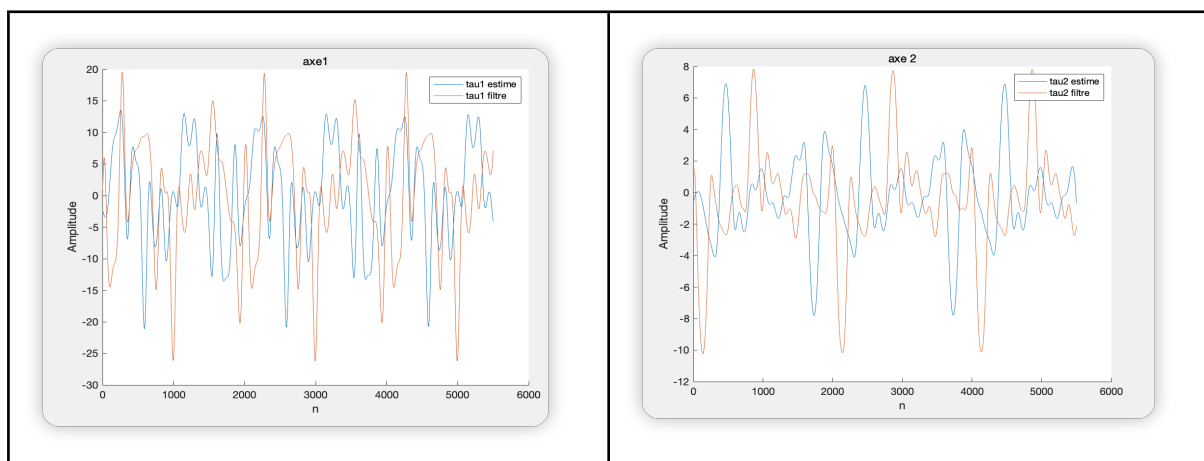
Pour `'F=Y*[x(1);x(2);x(3);x(4);x(5);x(6);x(7);x(8);x(9)]-b'`, nous avons obtenu :

$$\theta_{\text{estimé}} = [0.0160 \ 0.0020 \ 0.0050 \ 0.0030 \ 0.2725 \ 0.0002 \ 0.0000 \ 0.0160 \ 0.0005]$$



Pour `'F=Y*[exp(x(1)); exp(x(2)); exp(x(3)); exp(x(4)); exp(x(5)); exp(x(6)); exp(x(7)); exp(x(8)); exp(x(9))]-b'`, nous avons obtenu :

$$\theta_{\text{estimé}} = [0.1353 \ 0.0067 \ 0.0183 \ 0.0183 \ 3.3324 \ 0.8420 \ 0.0001 \ 0.0257 \ 0.0012].$$



La seconde série d'estimations s'est avérée être globalement plus proche des valeurs cibles, indiquant une meilleure précision dans l'estimation des paramètres.

Cependant, l'analyse des séries de paramètres estimés par rapport aux valeurs cibles révèle des différences significatives, suggérant la nécessité d'affiner le modèle ou la méthode d'estimation. L'observation que les valeurs estimées de  $\tau_1$  et  $\tau_2$  ne correspondent pas aux profils de  $\tau_1$  et  $\tau_2$  filtrés est particulièrement préoccupante. Cela suggère une déconnexion entre les signaux traités et les résultats de l'estimation, ce qui pourrait être attribué à plusieurs facteurs. Premièrement, cela peut indiquer que le filtrage n'a pas été suffisamment efficace pour éliminer le bruit tout en préservant les caractéristiques intrinsèques des signaux, ce qui a pu entraîner des estimations erronées ou biaisées. Deuxièmement, cela pourrait signifier que le modèle ou la méthode d'estimation utilisée n'est pas entièrement adapté ou calibré aux données spécifiques en question, nécessitant un examen ou une modification plus poussés.

L'inadéquation entre les signaux filtrés et les valeurs estimées peut également indiquer des problèmes sous-jacents des erreurs dans la mise en œuvre du processus de filtrage ou d'estimation.

#### **IV. Conclusion**

Ce projet a exploré divers aspects du traitement de données et de l'estimation de paramètres en utilisant MATLAB. Nous avons mis en œuvre plusieurs stratégies pour imposer des contraintes lors d'optimisations, mettant l'accent sur l'importance de filtrer et de prétraiter les données pour minimiser le bruit avant d'effectuer toute différentiation numérique. Particulièrement, nous avons utilisé la fonction `lsqnonlin` pour aborder efficacement les problèmes de moindres carrés non linéaires, tout en intégrant des transformations spécifiques pour assurer que les estimations obtenues soient toujours positives. En comparant des valeurs estimées à des valeurs cibles, nous avons confirmé l'importance de sélectionner des techniques d'optimisation et de transformation adéquates, afin d'améliorer la précision des estimations. Ce projet met en exergue le besoin crucial d'une approche rigoureuse et méthodique dans le traitement du signal pour obtenir des informations précises et fiables, conduisant à des conclusions significatives et opérationnelles.