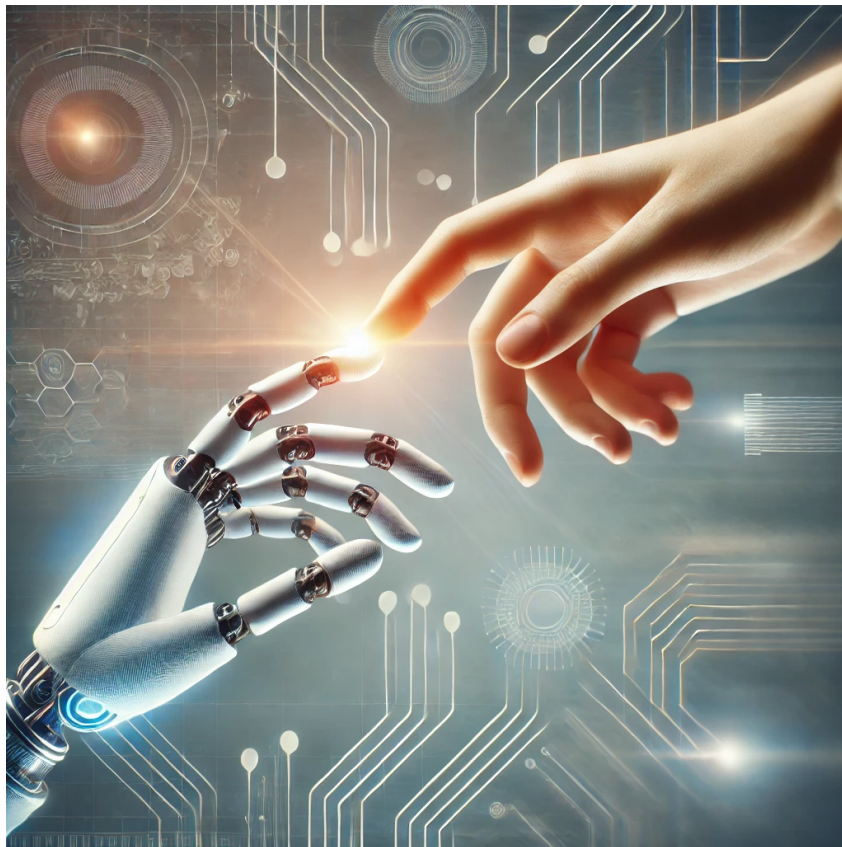


UQAC

Interaction Humain-Robot : Devoir 2



Constance ALOYAU
Erwan MAWART
Benjamin PELLIEUX

16 novembre 2024

ALOC25530200
MAWE14050200
PELB28120100

Table des matières

Introduction	3
Énoncé 1 : Conception du modèle du mécanisme	4
Question 1.1	4
Question 1.2	5
Question 1.3	5
Énoncé 2 : Simulation	9
Question 2.1	9
Question 2.2	10
Question 2.3	11
Discussions	12
Conclusion	12
Annexes	13
Annexe 1 : Liste des valeurs fixes	13
Annexe 2 : Fichier de calculs des plages de K_p et K_H	13
Annexe 3 : Fichier de configuration de la simulation	15
Annexe 4 : Modèle complet	18
Annexe 5 : Modèle humain	19
Annexe 6 : Modèle IAD	19
Annexe 7 : Modèle observateur	20
Lien vers le Dépôt GitHub	20

Table des figures

1	Modèle du mécanisme robotique et de l'humain [1] [2]	4
2	Commande des mécanismes robotique et humain avec observateur	5
3	Critère de Routh-Hurwitz S^2 en fonction de K_p	7
4	Lieu des racines du système	8
5	Évolution de K_H selon une rampe	9
6	Évolution de la réponse avec K_H selon une rampe	9
7	Commande du mécanisme robotique sans humain	10
8	Évolution de la réponse sans humain	10
9	Évolution de la phase de la réponse sans humain	10
10	Évolution de K_H selon trois fonctions step	11
11	Évolution de la réponse avec K_H selon trois fonctions step	11

Liste des tableaux

1	Tableau théorique du critère de Routh-Hurwitz	6
---	---	---

Introduction

Les vibrations présentes dans un mécanisme robotique manipulé par un opérateur humain peuvent affecter de manière significative la précision et la qualité des tâches réalisées. Ces vibrations, générées notamment par la force que l'humain exerce sur l'interface de commande, peuvent devenir problématiques, notamment lorsqu'elles sont amplifiées par les mouvements involontaires ou l'ajustement de la rigidité du bras de l'opérateur. L'interaction entre l'humain et la machine crée ainsi un système complexe où les propriétés mécaniques du robot et de l'opérateur se superposent, entraînant des oscillations et perturbations indésirables. Ces vibrations ne nuisent pas seulement à la performance du système en altérant la précision des actions exécutées, mais elles peuvent également compromettre le confort de l'utilisateur et, à terme, affecter sa santé, en provoquant des tensions musculaires ou des troubles liés aux mouvements répétitifs.

Dans ce contexte, cette étude se propose d'analyser et de compenser ces vibrations. Notre approche repose sur le développement d'un observateur capable de détecter et de compenser efficacement l'impact des vibrations générées par les forces et mouvements de l'opérateur. En intégrant un modèle dynamique de la rigidité humaine dans la boucle d'asservissement, nous visons à réduire les effets indésirables de ces vibrations sur la qualité de la tâche réalisée. L'objectif étant de proposer une solution technologique qui permette d'améliorer la précision et le confort de l'utilisateur tout en garantissant une interaction harmonieuse entre le robot et son opérateur.

$$\Leftrightarrow \begin{cases} \ddot{x}_1 = \frac{F - K_B x_1 + K_B x_2 - C_B \dot{x}_1 + C_B \dot{x}_2}{m_R} \\ \ddot{x}_2 = \frac{K_B x_1 - K_B x_2 + C_B \dot{x}_1 - C_B \dot{x}_2 - C_R \dot{x}_2}{M_R} \end{cases} \quad (1)$$

Grâce à celle-ci, on construit les deux équations de fonctionnement du système :

$$\dot{X} = AX + BU \Leftrightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-K_B}{m_R} & \frac{K_B}{m_R} & \frac{-C_B}{m_R} & \frac{C_B}{m_R} \\ \frac{K_B}{M_R} & \frac{-K_B}{M_R} & \frac{C_B}{M_R} & \frac{-(C_B + C_R)}{M_R} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_R} \\ 0 \end{bmatrix} \cdot F \quad (2)$$

$$Y = CX + DU \Leftrightarrow Y = CX \Leftrightarrow \begin{bmatrix} x_2 \\ \dot{x}_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} \quad (3)$$

L'équation 3 présente la sortie du système. Dans notre cas, on utilise x_2 (la position de la charge) et v_1 (la vitesse des moteurs) afin d'asservir notre système. Où x_2 permet à l'humain de corriger la trajectoire de la charge et v_1 d'asservir les moteurs.

Question 1.2

Lors d'une étude précédente, nous nous penchions sur la détection des vibrations générées par le système lorsque l'humain se rigidifie. Pour cela, nous analysons la vitesse des moteurs pour en extraire des caractéristiques temporelles qui nous permettent de déterminer la présence des vibrations.

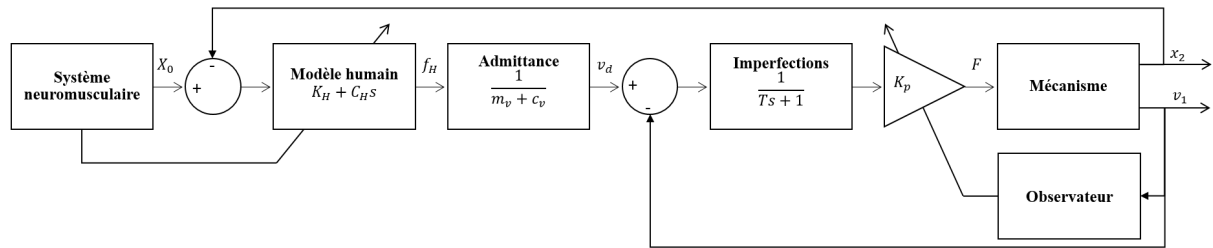


FIGURE 2 – Commande des mécanismes robotique et humain avec observateur

Afin de rendre le système autonome, il est donc nécessaire de rendre automatique cette recherche de caractéristiques. C'est pourquoi, comme le montre la figure 2, on ajoute un observateur qui va influencer sur la valeur du gain K_p , et donc, limiter les vibrations créées par le système.

Dans Simulink, on représentera ce bloc avec l'architecture contenue dans l'annexe 7. Celle-ci consiste à analyser l'écart-type du signal afin de déterminer la présence de vibrations.

Question 1.3

Le critère de Routh-Hurwitz est une méthode mathématique permettant de déterminer rapidement la stabilité d'un système. Elle consiste à produire un tableau à partir des coefficients du dénominateur de la fonction de transfert tel que présenté au tableau 1. Une fois construit,

on observe sa première colonne. Le nombre de changement de signe détermine la quantité de racines étant dans la partie instable du plan-s (c'est-à-dire dans la partie réel positive).

s^n	a_n	a_{n-2}	a_{n-4}	\dots
s^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	\dots
s^{n-2}	b_{n-1}	b_{n-3}	b_{n-5}	\dots
s^{n-3}	c_{n-1}	c_{n-3}	c_{n-5}	\dots
\vdots	\vdots	\vdots	\vdots	\vdots
s^0	h_{n-1}	h_{n-3}	h_{n-5}	\dots

TABLE 1 – Tableau théorique du critère de Routh-Hurwitz

Dans le tableau 1, on retrouve :

- a_{n-i} , les coefficients du dénominateur.
- b_{n-i} , donnés par : $b_{n-i} = \frac{-1}{a_{n-1}} \begin{vmatrix} a_n & a_{n-i-1} \\ a_{n-1} & a_{n-i-2} \end{vmatrix} = \frac{a_{n-1}a_{n-i-1} - a_n a_{n-i-2}}{a_{n-1}}$
- c_{n-i} , donnés par : $c_{n-i} = \frac{-1}{b_{n-1}} \begin{vmatrix} a_n & a_{n-i-2} \\ b_{n-1} & b_{n-i-2} \end{vmatrix} = \frac{b_{n-1}a_{n-i-2} - a_n b_{n-i-2}}{b_{n-1}}$
- etc.

Cette méthode est utilisée pour déterminer l'intervalle de valeurs du gain K_p . Afin de simplifier la construction de cette table, nous utilisons la fonction suivante dans Matlab :

Listing 1 – Fonction Matlab génération du critère de Routh-Hurwitz

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parametres : %
% - TF : la fonction de transfert a analyser %
% - Oldvars : les variables a remplacer %
% - Newvars : les valeurs a implementer %
% - s : la variable de Laplace %
% Sortie : %
% - S : le tableau construit %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function S = calcTabRH(TF, Oldvars, Newvars, s)
% Extraction des coefficients du denominateur =====
[~, Den] = numden(TF);
[an, terms] = coeffs(Den, s);
l_terms = length(terms);
l = round(l_terms/2);

% Creation du tableau =====
S = sym(zeros(l_terms, l));
% Ajout des coefficients du denominateur
for n=1:l
    i = n*2;
    S(l_terms, n) = an(i-1);
    if i < l_terms
        S(l_terms-1, n) = an(i);
    end
end
```

```

end
% Calcul des autres elements du tableau
for k=l_terms-2:-1:1
    for n=1:l-1
        S(k, n) = (S(k+1, 1)*S(k+2, n+1)-S(k+2, 1)*S(k+1,
            n+1))/S(k+1, 1);
    end
end

% Remplacement des valeurs connues =====
S = simplify(subs(S, Oldvars, Newvars));

% Affichage du tableau =====
disp("Tableau du critere de Routh-Hurwitz :")
disp(S)
end

```

Les gains K_p et K_H étant dépendants l'un de l'autre, leur plage de valeurs varie en fonction de l'autre. Par conséquent, il nous est obligatoire de fixer l'un des deux. Ici, nous considérons $K_H = 50N/m$, s'assurant ainsi la stabilité du système.

En remplaçant les variables présentes dans le tableau du critère de Routh-Hurwitz par les valeurs données en annexe 1, il reste cinq coefficients contenant K_p . Parmi ceux-ci, le critère S^2 devient négatif à $K_p = 90$, comme présenté à la figure 3. Ainsi, on déduit que $K_p \in \langle 0, 90 \rangle$.

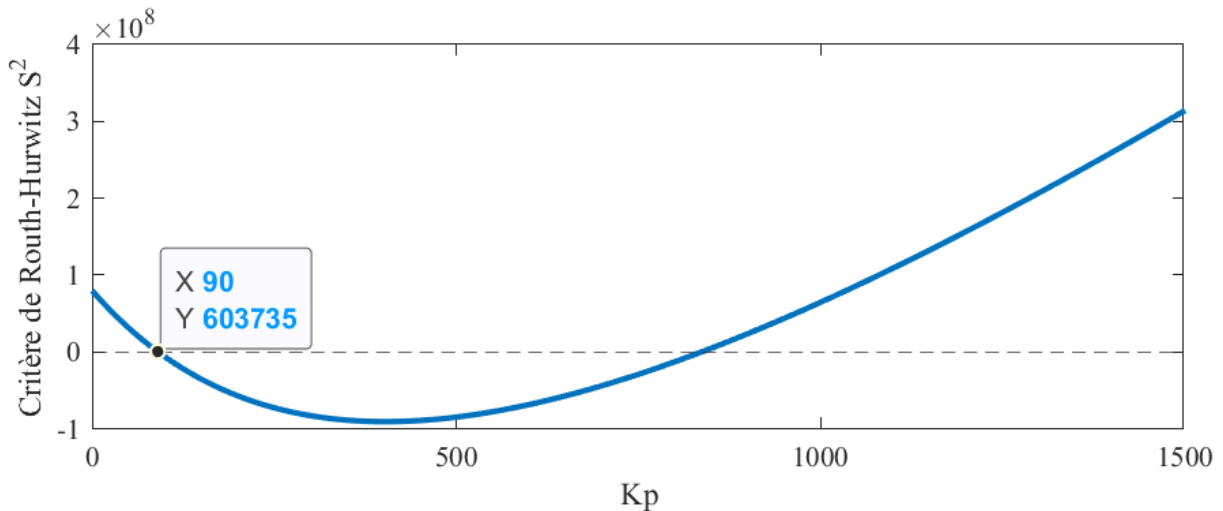


FIGURE 3 – Critère de Routh-Hurwitz S^2 en fonction de K_p

Afin de confirmer cette valeur, nous pouvons utiliser le lieu des racines. Cette méthode consiste à faire varier un gain en entrée du système et ainsi de déterminer le moment auquel ce-dernier devient instable.

Ici, le gain variable est K_H et on fixe $K_p = 90$, condition limite déterminée à la figure 3. En traçant le lieu des racines, comme montré à la figure 4, on remarque que le système est stable jusqu'à $K_H \approx 50$ ($K_H = 46.9532$ au point de mesure noté sur la figure 4). Cela confirme la plage déterminée via le critère de Routh-Hurwitz.

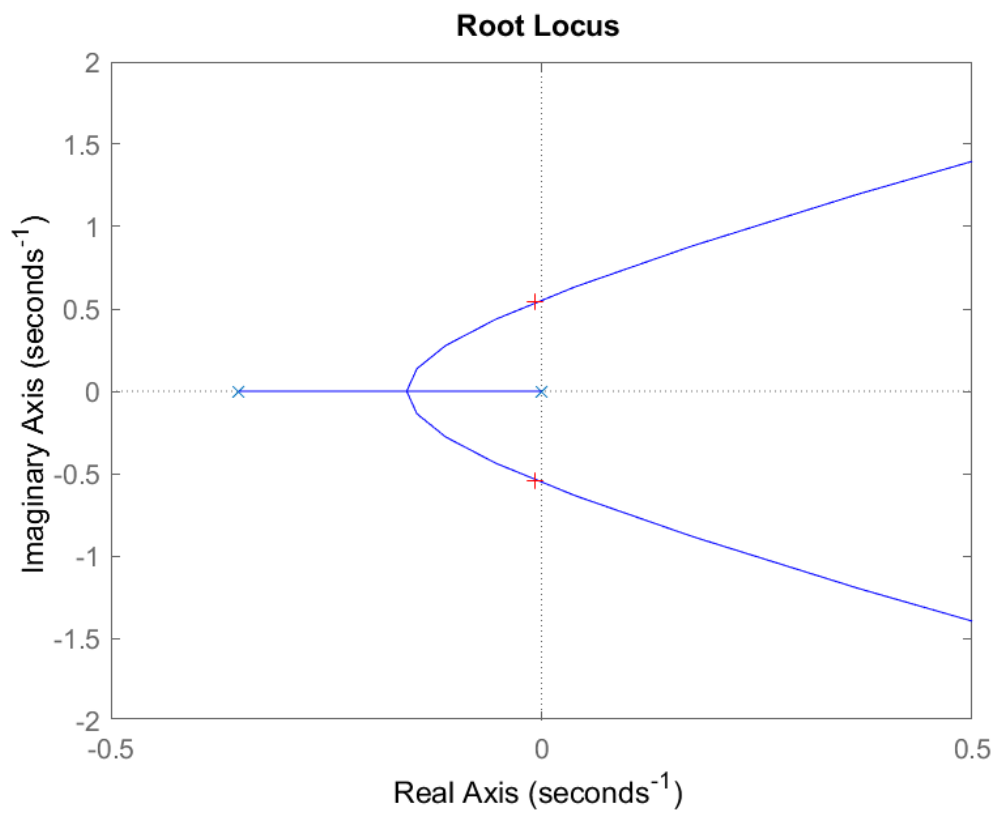


FIGURE 4 – Lieu des racines du système

Énoncé 2 : Simulation

Question 2.1

Dans le but de confirmer les plages de gains admissibles, nous allons faire varier progressivement la valeur de K_H selon une fonction rampe. Cette approche permettra d'observer l'effet de l'augmentation graduelle de la rigidité sur le système, et de déterminer précisément le seuil auquel une instabilité se manifeste.

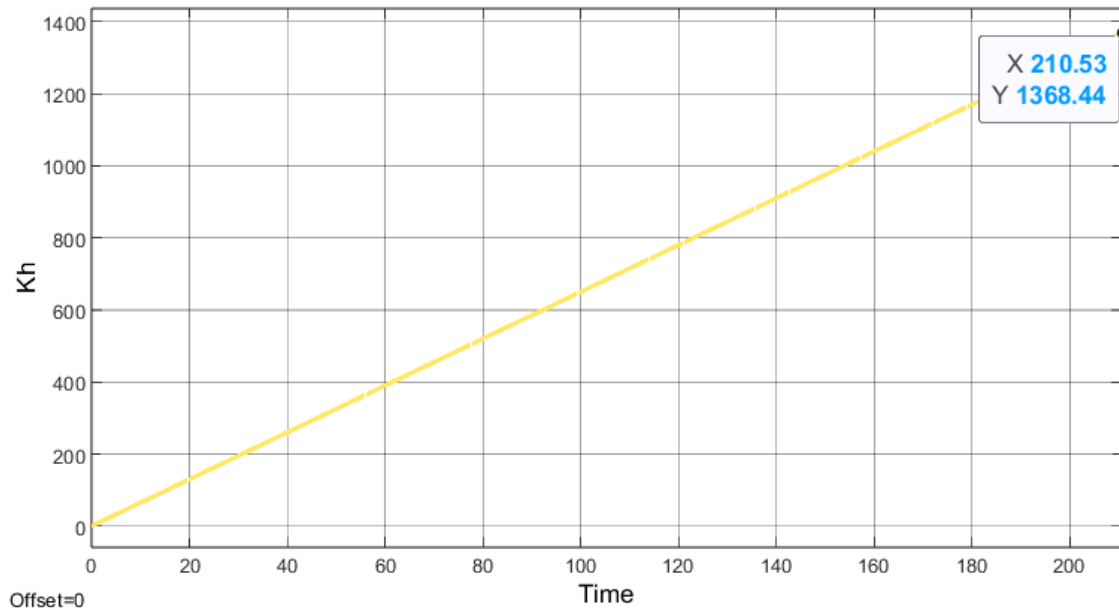


FIGURE 5 – Évolution de K_H selon une rampe

La figure 6 présente la réponse du système pour cette simulation, tandis que la figure 5 illustre l'évolution de la rigidité K_H au fil du temps. On observe que la réponse du système tend vers l'infini lorsque $K_H \approx 1368$, ce qui indique une instabilité. Cette observation permet de conclure que la plage de valeurs de K_H admettant un comportement stable est $K_H \in \langle 0, 1350 \rangle$.

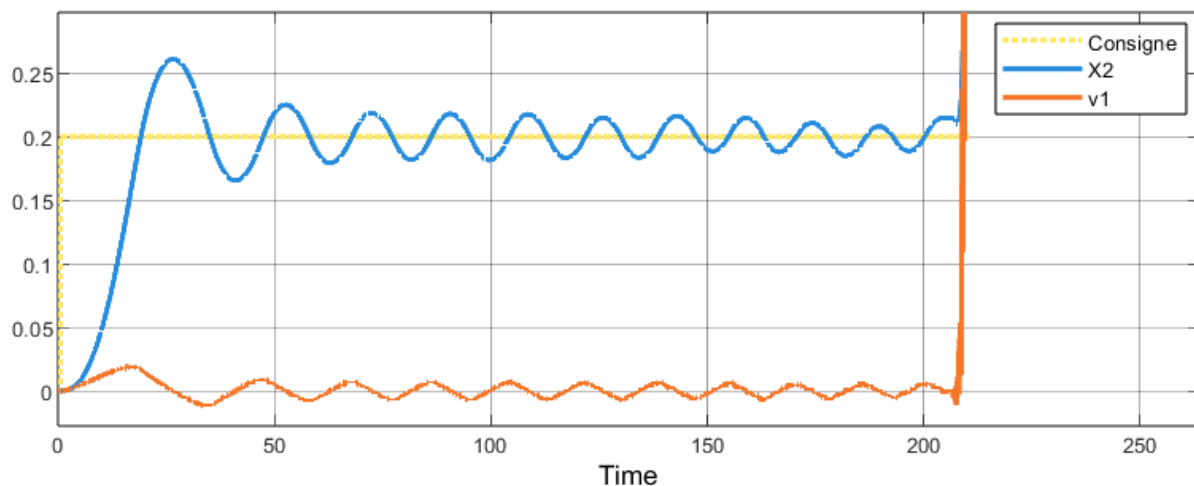


FIGURE 6 – Évolution de la réponse avec K_H selon une rampe

Question 2.2

Afin de confirmer la stabilité du système, on retire le modèle humain, comme le montre la figure 7. Cela nous permettra d'observer le fonctionnement du système sans perturbations en entrée.

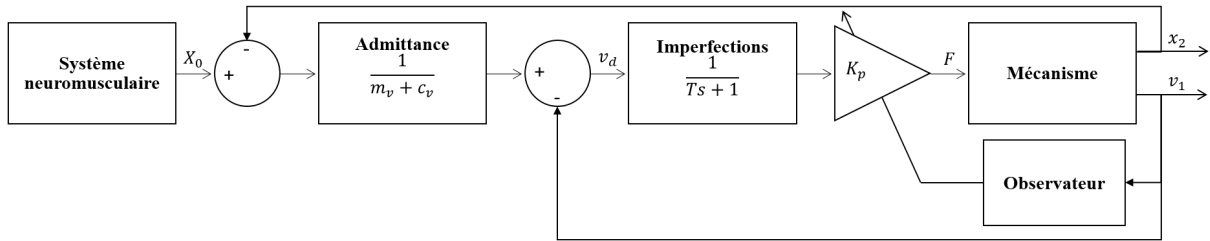


FIGURE 7 – Commande du mécanisme robotique sans humain

La réponse obtenue présente un gain (cf. figure 8) sans oscillation, ce qui témoigne d'une stabilité satisfaisante du système. En effet, l'absence d'oscillations suggère que le système atteint son état d'équilibre sans dépasser ou fluctuer autour de la consigne, caractéristique d'un comportement stable et bien contrôlé.

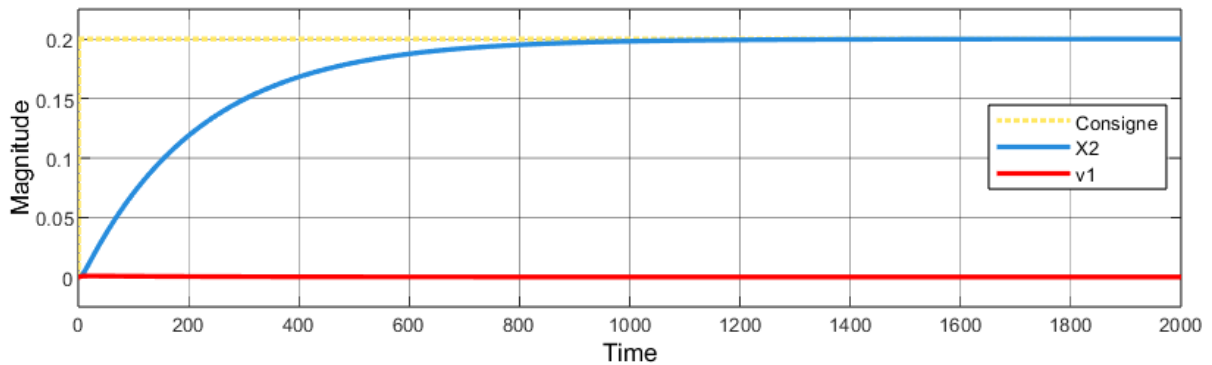


FIGURE 8 – Évolution de la réponse sans humain

Cette observation est renforcée par l'analyse du déphasage, illustrée dans la figure 9. Au cours de la simulation, aucun déphasage significatif n'est observé, ce qui confirme que la phase de la sortie suit de manière quasi-linéaire l'entrée, sans délai ou retard perceptible.

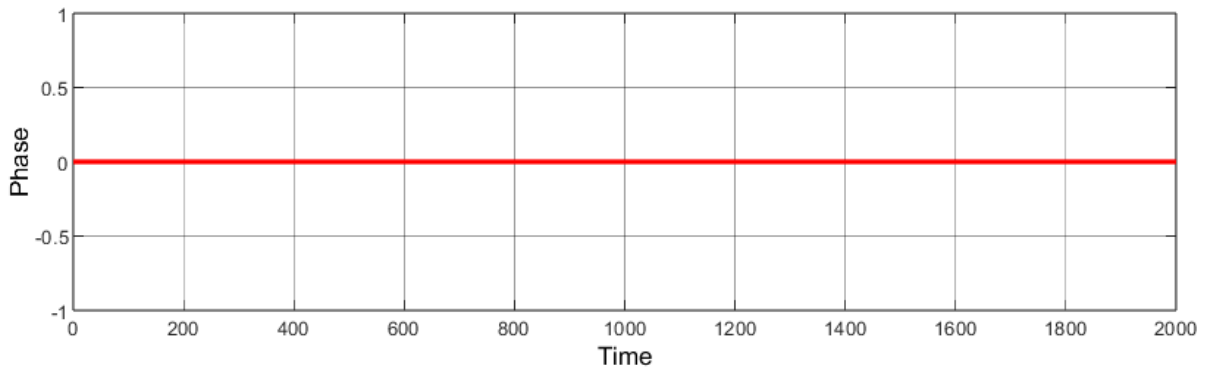


FIGURE 9 – Évolution de la phase de la réponse sans humain

Question 2.3

Notre dernière simulation consistera à appliquer différentes rigidités humaine afin d'observer la réponse de notre système. Pour cela, on fait varier K_H selon trois valeurs définies par trois fonctions "step" configurées telles que présenté dans l'annexe 5.

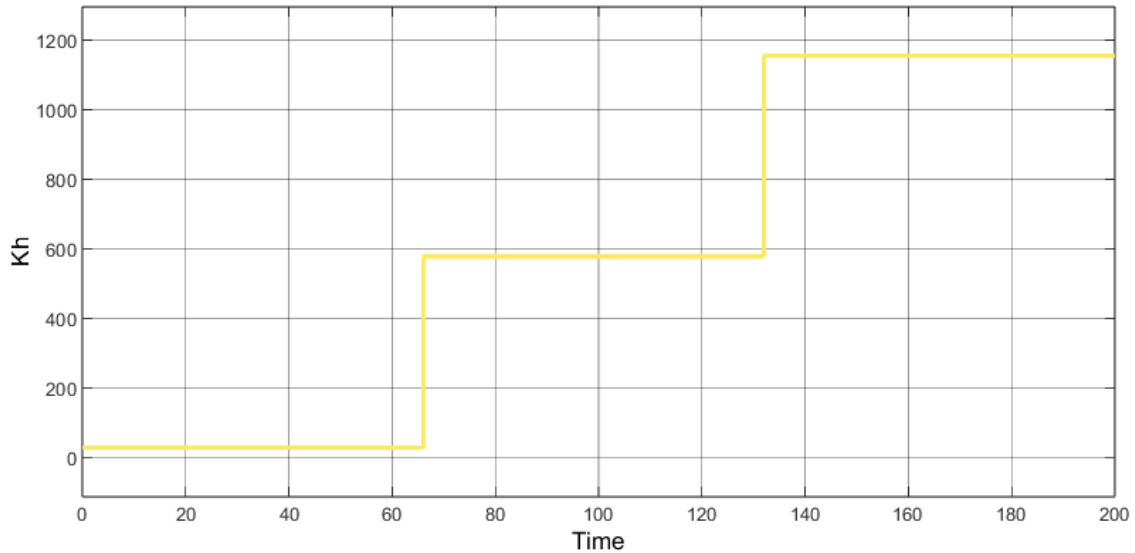


FIGURE 10 – Évolution de K_H selon trois fonctions step

Sur la réponse présentée dans la figure 11, on observe que l'amortissement du système est satisfaisant tant que la valeur de K_H reste proche de zéro. Cependant, lors de la première augmentation de la rigidité, on note l'apparition progressive d'ondulations. Dès que K_H augmente pour sa valeur finale, ces ondulations deviennent plus prononcées. De plus, des vibrations commencent à se manifester sur le signal v_1 , qui représente la vitesse des moteurs.

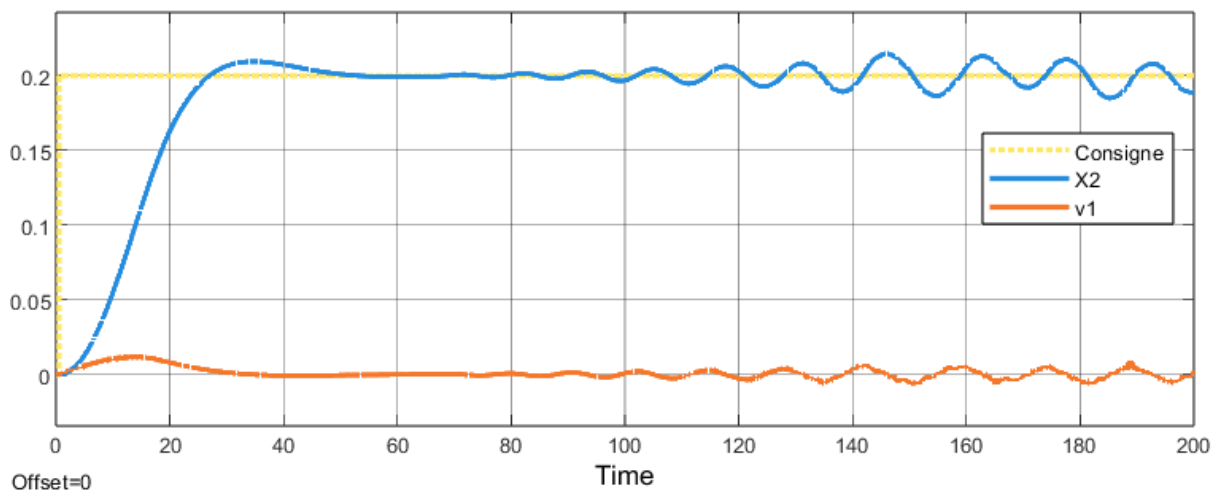


FIGURE 11 – Évolution de la réponse avec K_H selon trois fonctions step

Discussions

Cette étude a permis de concevoir un système de compensation des vibrations générées par la rigidité du corps humain.

Cependant, le système présente certaines limites. En effet, il a été observé qu'au-delà d'un certain seuil de rigidité, le système rencontre des difficultés à amortir efficacement les oscillations et à compenser les vibrations. Si la rigidité continue d'augmenter, le système perd sa capacité à suivre les mouvements, ce qui conduit à une instabilité complète.

Ces résultats soulignent qu'il est essentiel pour l'opérateur de maintenir une vigilance constante lors de l'utilisation de cet outil, notamment en veillant à limiter l'impact de la rigidité corporelle sur le système.

Conclusion

En conclusion, ce système offre une solution prometteuse pour compenser les effets de la rigidité corporelle, contribuant ainsi à optimiser les performances du système dans des conditions d'utilisation variées.

Après avoir validé les performances du système par simulation, la prochaine étape consistera à concevoir un prototype fonctionnel pour tester son efficacité dans des conditions réelles. Si cette phase de prototypage se révèle concluante, une phase de validation expérimentale pourra être lancée, ouvrant ainsi la voie à une commercialisation éventuelle du produit.

Annexes

Annexe 1 : Liste des valeurs fixes

$M_R = 500 \text{ kg}$	(masse de la charge)
$m_R = 50 \text{ kg}$	(masse du système)
$m_v = 20 \text{ kg}$	(masse ressentie)
$K_B = 40000 \text{ N/m}$	(constante de raideur des courroies)
$C_B = 40 \text{ N} \cdot \text{s/m}$	(coefficient d'amortissement des courroies)
$C_H = 23.45 \text{ N} \cdot \text{s/m}$	(coefficient d'amortissement de l'humain)
$C_R = 100 \text{ N} \cdot \text{s/m}$	(coefficient de frottement)
$c_v = 20 \text{ N} \cdot \text{s/m}$	(coefficient d'amortissement vituel)
$T = 0.1 \text{ s}$	(temps d'échantillonnage)

Annexe 2 : Fichier de calculs des plages de K_p et K_H

Listing 2 – Fonction simulink de calcul du compensateur

```
% Initialisation =====
close all
system_config
clear Kp Kh
FontName = 'Times';

% Calcul des boucles =====
[Boucle_ouverte, Boucle] = Calc_Sys();

% Creation de la table de Routh-Hurwitz =====
syms Kp Kh s
S_boucle = calcTabRH(Boucle, [sym('MR') sym('mR') sym('Kb')
    sym('Cb') sym('CR') sym('T') sym('mv') sym('cv')], [MR mR Kb
    Cb CR T mv cv], s);

% Recherche de la plage de valeurs Kp =====
Kp = 0:1500;
Kh = 50;
figure;
hold on
for i = 1:length(S_boucle(:,1))
    if ~isempty(find(symvar(S_boucle(i,1)) == sym('Kp'), 1))
        S_boucle(i,1) = subs(S_boucle(i,1), sym('Kh'), Kh);
        S_calc = eval(subs(S_boucle(i,1), sym('Kp'), Kp));

        subplot(3, 2, i)
        plot(Kp, S_calc, 'LineWidth', 2)
        yline(0, '--')
        xlabel('Kp')
        ylabel("Critere de Routh-Hurwitz S^" + i)
        fontname(FontName)
    end
end
end
```



```

% - Newvars : les valeurs a implementer %
% - s : la variable de Laplace %
% Sortie : %
% - S : le tableau construit %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function S = calcTabRH(TF, Oldvars, Newvars, s)
    % Extraction des coefficients du denominateur =====
    [~, Den] = numden(TF);
    [an, terms] = coeffs(Den, s);
    l_terms = length(terms);
    l = round(l_terms/2);

    % Creation du tableau =====
    S = sym(zeros(l_terms, l));
    % Ajout des coefficients du denominateur
    for n=1:l
        i = n*2;
        S(l_terms, n) = an(i-1);
        if i < l_terms
            S(l_terms-1, n) = an(i);
        end
    end
    % Calcul des autres elements du tableau
    for k=l_terms-2:-1:1
        for n=1:l-1
            S(k, n) = (S(k+1, 1)*S(k+2, n+1)-S(k+2, 1)*S(k+1,
                n+1))/S(k+1, 1);
        end
    end

    % Remplacement des valeurs connues =====
    S = simplify(subs(S, Oldvars, Newvars));

    % Affichage du tableau =====
    disp("Tableau du critere de Routh-Hurwitz :")
    disp(S)
end

```

Annexe 3 : Fichier de configuration de la simulation

Listing 3 – Fonction simulink de calcul du compensateur

```

clear;
clc;

%Temps de simulation
sim_time = 200; % s
% Taille de la memoire tampon de l'observateur
buffer_size = 128;

% Bloc "Modele humain" =====

```



```

Kh = 550;                % N/m
Ch = 23.45;              % N*s/m

% Admittance =====
cv = 20;                 % N*s/m
mv = 20;                 % kg

% Imperfections =====
T = .1;                  % s

% Mecanisme =====
mR = 50;                  % kg
MR = 500;                 % kg
CR = 100;                 % N*s/m
Cb = 40;                  % N*s/m
Kb = 40000;               % N/m
[A, B, C, D] = calcIAD(Kb, Cb, CR, mR, MR);

% Observateur =====
% Ecart-type limite avant vibrations
ec_max = 1e-3;

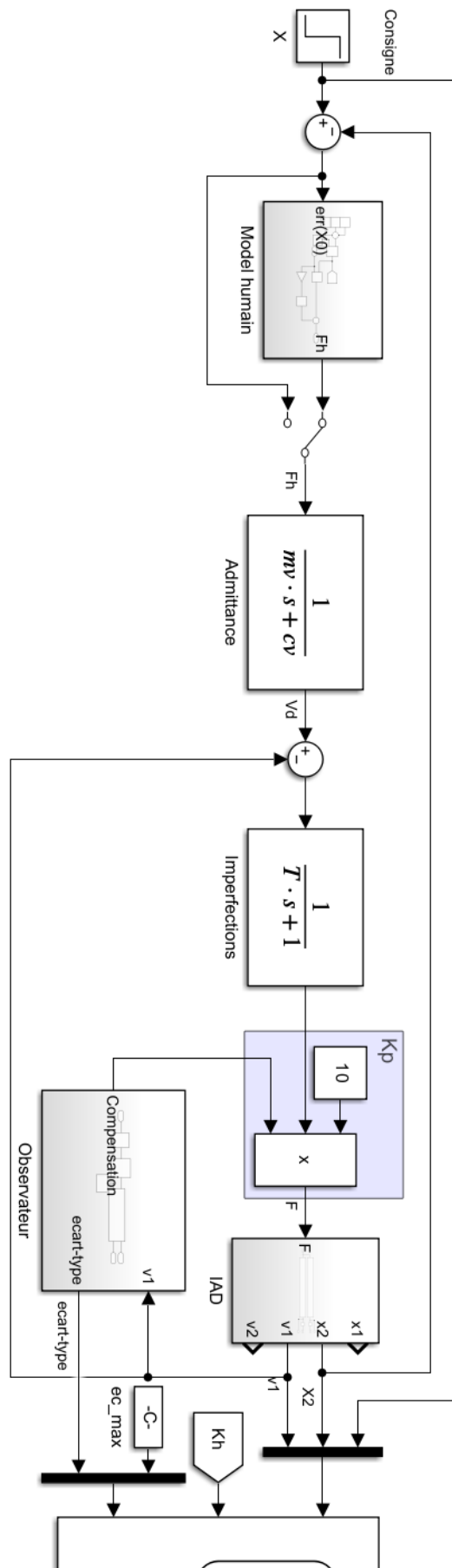
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parametres :
%   - K : ici Kb
%   - C1 : ici Cb
%   - C2 : ici CR
%   - m : ici mR
%   - M : ici MR
% Sortie :
%   - A : la matrice d'etat
%   - B : la matrice de commande
%   - C : la matrice d'observation
%   - D : la matrice d'action directe
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [A, B, C, D] = calcIAD(K, C1, C2, m, M)
    mK = K / m;
    mC = C1 / m;
    MK = K / M;

    A = [
        0      0      1      0;
        0      0      0      1;
        -mK     mK     -mC    mC;
        MK     -MK     C1/M   -(C1+C2)/M
    ];
    B = [
        0;
        0;
        1/m;
    ]

```

```
        0
    ];
C = [
    0      1      0      0;
    0      0      1      0;
    ];
D = zeros(2, 1);
end
```

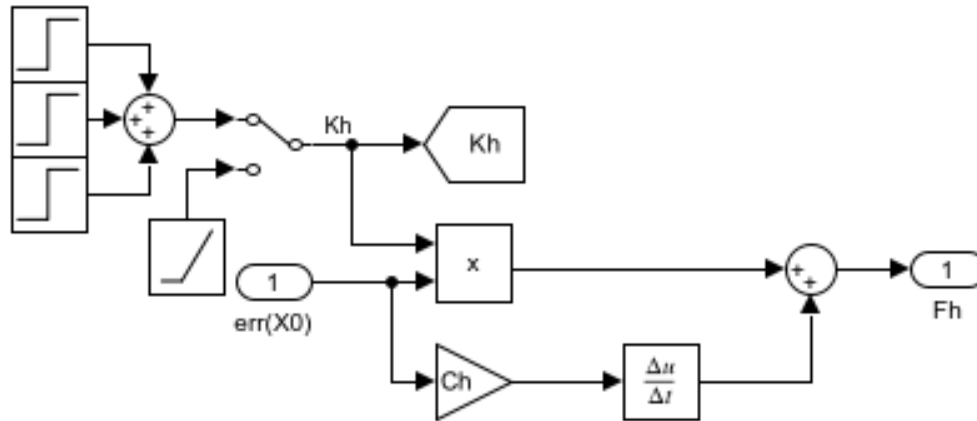
Annexe 4 : Modèle complet



Où le bloc "step" (X) est paramétré avec :

- "step time" = 0.5
- "initial value" = 0
- "final value" = 0.2

Annexe 5 : Modèle humain



Où :

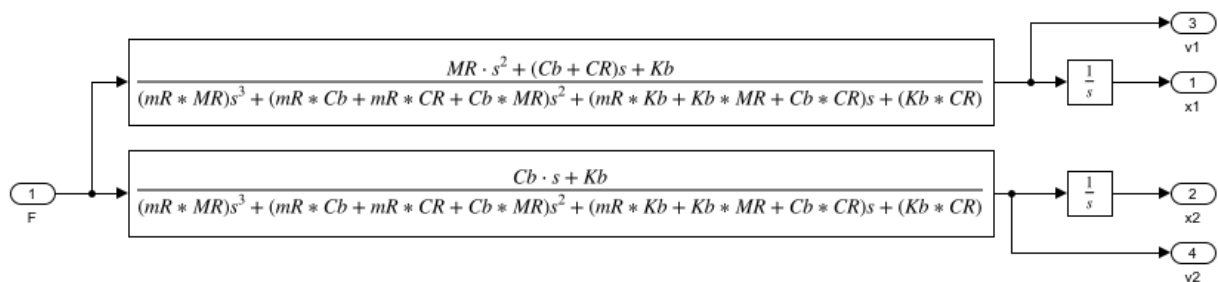
Le bloc "rate transition" est paramétré avec :

- "slope" = 6.5
- "start time" = 0
- "initial output" = 0

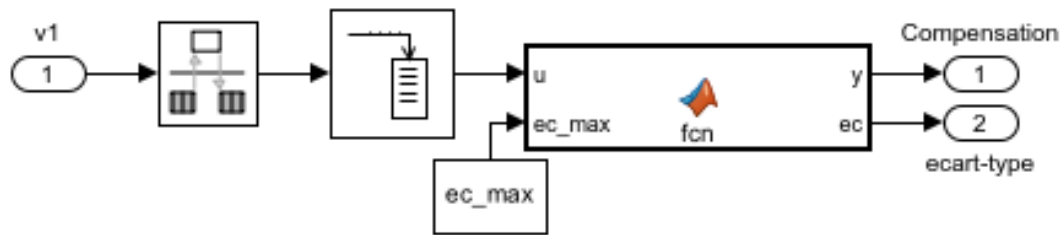
Et les blocs "step" sont paramétrés avec :

Step haut	Step milieu	Step bas
• "step time" = 0	• "step time" = 66	• "step time" = 132
• "initial value" = 0	• "initial value" = 0	• "initial value" = 0
• "final value" = 27.5	• "final value" = 550	• "final value" = 577.5

Annexe 6 : Modèle IAD



Annexe 7 : Modèle observateur



Où :

Le bloc "rate transition" est paramétré avec :

- "initial conditions" = 0
- "ouput port sample time" = 0.01

Le bloc "buffer" est paramétré avec :

- "ouput buffer size (per channel)" = 128
- "buffer overlap" = 64
- "initial conditions" = 0

Et le bloc "MATLAB Function" contient :

Listing 4 – Fonction simulink de calcul du compensateur

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parametres : %
% - u : segment de signal %
% - ec_max : ecart-type limite avant vibration %
% Sortie : %
% - y : le coefficient de compensation %
% - ec : l'ecart-type du segment %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [y, ec] = fcn(u, ec_max)
    % Calcul de l'ecart-type =====
    ec = std(u);

    % Calcul de la compensation =====
    y = 1 - ec * (1/ec_max);
end

```

Lien vers le Dépôt GitHub

https://github.com/BlueWan14/Cours_IHR/tree/main/Devoir_2

Références

- [1] Martin J-D OTIS et Clément GOSSELIN ALEXANDRE CAMPEAU-LECOURS. *Modeling of physical human–robot interaction : Admittance controllers applied to intelligent assist devices with large payload*. URL : <https://journals.sagepub.com/doi/epub/10.1177/1729881416658167>.
- [2] Martin J-D OTIS et Alexandre CAMPEAU-LECOURS MOHAMED AMIR SASSI. *Active stability observer using artificial neural network for intuitive physical human–robot interaction*. URL : <https://journals.sagepub.com/doi/epub/10.1177/1729881417727326>.