



UNIVERSITÉ
TOULOUSE III
PAUL SABATIER



Université
de Toulouse

Compte Rendu : Systèmes Linéaires et Invariants

Régulation d'un pendule Inversé

HAMMOUD Moustafa
Encadrant : M.Sylvain Durola
M1 ISTR-RODECO

23 mai 2021

Table des matières

1	Objectifs	2
2	Présentation de la pendule inversée	3
3	Modèle non linéaire	4
4	Modèle linéarisé	6
5	Correction du procédé	9
6	Mode Switching	13
7	Conclusion	14
8	Script MATLAB	15

1 Obectifs

Les objectifs du TP sont les suivants :

- Savoir établir un modèle linéarisé autour d'un point d'équilibre.
- Savoir établir les propriétés structurelle d'un système linéaire.
- Savoir synthétiser une commande par retour d'état.
- Savoir faire la simulation de modèles linéaires et non linéaires.
- Savoir utiliser les différentes commandes MATLAB pour étudier un système.
- Savoir utiliser Matlab et sa toolbox simulink.

2 Présentation de la pendule inversée

Le pendule inversé de l'entreprise QUANSER est composé d'un moteur SVR02, qui entraîne un bras disposant à son extrémité un pendule (tige) lié l'un à l'autre par une liaison pivot. Ce système possède deux capteurs (une roue codeuse chacune) qui nous renseignent sur l'angle du pendule par rapport au bras et sur l'angle du bras par rapport au bâti. Les différents éléments du pendule sont décrits par la Figure 3.

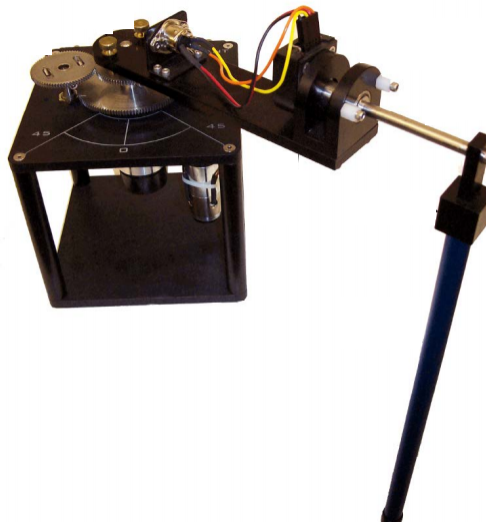


FIGURE 1 – Modèle de la pendule

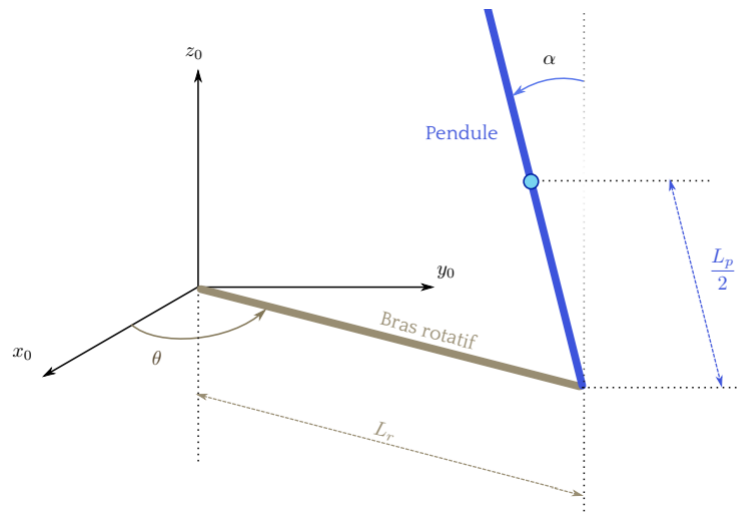


FIGURE 2 – Bras rotatif

3 Modèle non linéaire

On considère le système modélisé dans la figure 1. On note que l'entrée est la tension V_m exercée par le moteur. Le pendule est relié à un bras qui est entraîné par un moteur. L'angle entre le bras et le moteur est noté θ et α l'angle entre le pendule et la verticale. Les équations du mouvement, déterminées à l'aide de la mécanique Lagrangienne, sont données par les deux équations différentielles suivantes :

D'après les équations différentielles ; nous déduisons le vecteur d'état x suivant :

$$\begin{aligned} & \left(m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos(\alpha)^2 + J_r \right) \ddot{\theta} - \left(\frac{1}{2} m_p L_p L_r \cos(\alpha) \right) \ddot{\alpha} + \left(\frac{1}{2} m_p L_p^2 \sin(\alpha) \cos(\alpha) \right) \dot{\theta} \dot{\alpha} + \\ & \left(\frac{1}{2} m_p L_p L_r \sin(\alpha) \right) \dot{\alpha}^2 = \frac{\eta_g K_g \eta_m k_t (V_m - K_g k_m \dot{\theta})}{R_m} - B_r \dot{\theta} \\ & - \frac{1}{2} m_p L_p L_r \cos(\alpha) \ddot{\theta} + \left(J_p + \frac{1}{4} m_p L_p^2 \right) \ddot{\alpha} - \frac{1}{4} m_p L_p^2 \cos(\alpha) \sin(\alpha) \dot{\theta}^2 - \frac{1}{2} m_p L_p g \sin(\alpha) = -B_p \dot{\alpha} \end{aligned}$$

Symbole	Description	Valeur
B_p	Coefficient de frottement visqueux du pendule	0.0024 kg/m ²
B_r	Coefficient de frottement visqueux du bras	0.0024 kg/m ²
η_g	Rendement du réducteur	0.9
η_m	Rendement du moteur	0.69
K_g	Rapport de vitesse	70
k_m	Gain f.e.m	7.68 · 10 ⁻³ V/(rad/s)
k_t	Gain couple moteur/courant	7.68 · 10 ⁻³ N · m/A
J_p	Moment d'inertie du pendule au centre de masse	0.0012 kg/m ²
J_r	Moment d'inertie du bras au centre de masse	9.98 · 10 ⁻⁴ kg/m ²
L_m	Inductance du moteur	0.18 mH
L_p	Longueur du pendule	0.337 m
L_r	Longueur du bras, du pivot à l'extrémité	0.216 m
m_p	Masse du pendule	0.127 kg
m_r	Masse du bras	0.257 kg
R_m	Résistance du moteur	2.60Ω
V_m	Tension envoyé au moteur	À régler

FIGURE 3 – Valeurs des différents paramètres du pendule

$$x = \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \theta \\ \alpha \end{bmatrix}$$

Avec $\dot{\alpha}$ et $\dot{\theta}$ les vitesses angulaires.

Etude des points d'équilibre

Le point d'équilibre (repos) du système est lorsque notre vecteur d'état $x=0$.

Le point d'équilibre est décrit alors par un pendule immobile (vitesse angulaire nulle) avec un angle $\theta = 0$ marqué sur le moteur.

Remarque : Si la vitesse est nulle, l'accélération l'est aussi, $\ddot{\alpha} = 0$.

Nous obtenons alors :

$$\frac{\mu_g K_g \mu_m K_t V_m}{R_m} = 0 \quad (1)$$

$$\frac{1}{2} m_g L_p g \sin(\alpha) = 0 \quad (2)$$

$\sin(\alpha) = 0$ pour $\alpha = \pi$ et $\alpha = 0$, position vertical vers le haut et vers le bas.

On constate alors d'après (1) et (2) que le point d'équilibre est atteint lorsque la tension V_m est nulle et pour $\alpha = \pi$ ou $\alpha = 0$. Par contre, si $\alpha = 0$, la tension V_m peut-elle être égale à 0 ?

Quelle est la force qui va tenir le pendule vers le haut ?

Sur le dispositif réel, le pendule semble chercher à se stabiliser toujours au points bas pour $\alpha = \pi$.

Est-ce que le point d'équilibre haut est instable ?

Nous allons vérifier si cette analyse est correcte, en faisant des simulations via SIMULINK sur le modèle non linéaire.

Pour $\alpha = 0$, nous remarquons que θ et α reconverge vers le point de repos bas ($\alpha = \pi$ et θ diverge)

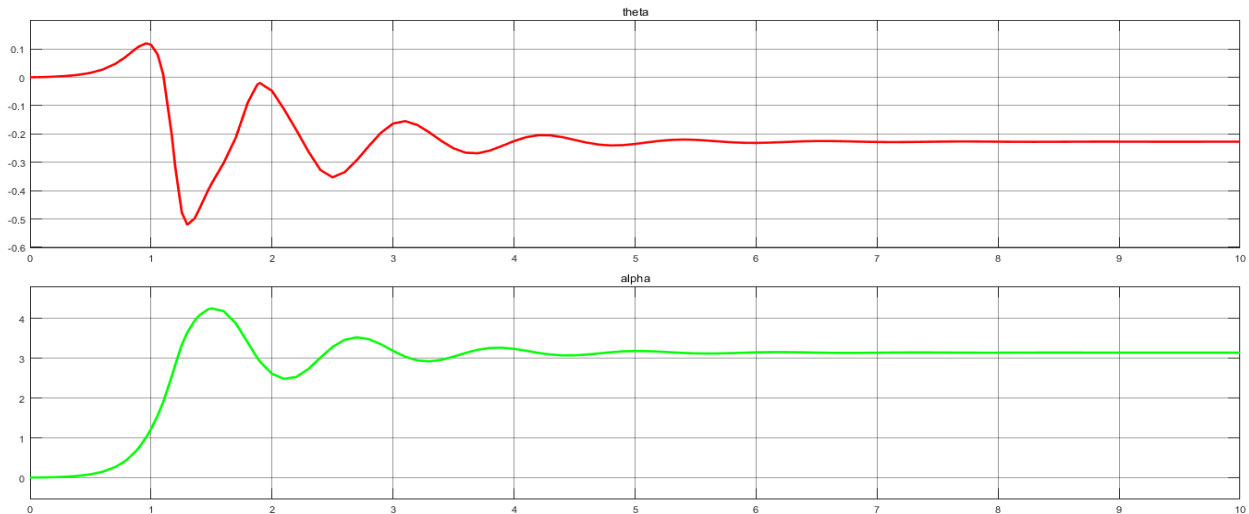


FIGURE 4 – Point d'équilibre haut

On remarque bien que pour $\alpha = 0$, θ , $\dot{\alpha}$ et $\dot{\theta}$ sont quasiment égales à 0, nous sommes donc au point d'équilibre, les amplitudes des oscillations sont de l'ordre de 10^{-16} , cela est due à des petites perturbations externes qui peuvent être rejeté facilement. A l'œil nu, en observant le pendule nous ne pouvons pas voir ces oscillations, il est clairement à l'équilibre.

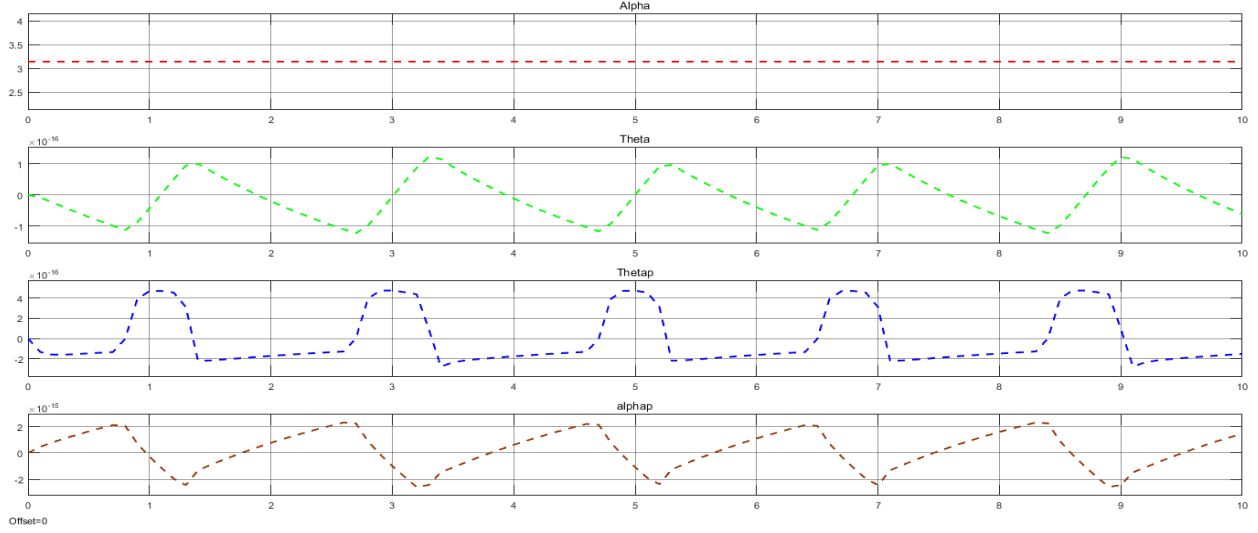


FIGURE 5 – Point d'équilibre bas

4 Modèle linéarisé

Pour obtenir un modèle linéarisé, nous avons effectué une décomposition en série de Taylor arrêtée à l'ordre 1, nous obtenons alors les matrices suivantes :

$$A_{haut} = \frac{1}{J_t} \begin{bmatrix} 0 & 0 & J_t & 0 \\ 0 & 0 & 0 & J_t \\ 0 & \frac{1}{4}m_p^2L_p^2L_rg & \left(J_p + \frac{1}{4}m_pL_p^2\right)(-B_r - K_gk_m\gamma) & -\frac{1}{2}m_pL_pL_rB_p \\ 0 & \frac{1}{2}m_pL_pg(J_r + m_pL_r^2) & \left(\frac{1}{2}m_pL_pL_r\right)(-B_r - K_gk_m\gamma) & -(J_r + m_pL_r^2)B_p \end{bmatrix}$$

$$B_{haut} = \frac{1}{J_t} \begin{bmatrix} 0 \\ 0 \\ \left(J_p + \frac{1}{4}m_pL_p^2\right)\gamma \\ \frac{1}{2}m_pL_pL_r\gamma \end{bmatrix} \quad C_{haut} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

avec

$$\gamma = \frac{\eta_n\eta_mK_gk_t}{R_m} \quad J_t = J_pm_pL_r^2 + J_rJ_p + \frac{1}{4}J_rm_pL_p^2.$$

Analyse du modèle linéarisé

Nous allons analyser le modèle linéarisé au point d'équilibre supérieur (analyse boucle ouverte) :

1. Analyse de la stabilité
2. Analyse de la commandabilité et de l'observabilité

Sur MATLAB, nous pouvons effectuer cette analyse, le script ci dessous explique cela.

```

clear all
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Modele linarise suprieur%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A = [0 0 1 0 ;
      0 0 0 1 ;
      0 Mp^2*(Lp/2)^2*Lr*g/Jt ((Jp+Mp*(Lp/2)^2)*(-Dr-Kg*km*gamma))/Jt -Mp*(Lp/2)*Lr*Dp/Jt ;
      0 Mp*g*(Lp/2)*(Jr+Mp*Lr^2)/Jt (Mp*(Lp/2)*Lr)*(-Dr-gamma*Kg*km)/Jt
      -Dp*(Jr+Mp*Lr^2)/Jt] ;

B = [0 ; 0 ; ((Jp+Mp*(Lp/2)^2)*gamma)/Jt ; (Mp*(Lp/2)*Lr)*gamma/Jt] ;
C = eye(2,4) ;
D = zeros(2,1) ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Espace d'tat%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sys=ss(A,B,C,D)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Diagonalisation de la matrice dynamique A%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[V,D]=eig(A)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Commandabilit\Observabilit%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Co=ctrb(sys)
Ob=obsv(sys)
r=rank(Co)
r1=rank(Ob)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Fonction de Transfert%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
G=tf(sys)

```

On obtient les résultats suivants :

$$Co = \begin{bmatrix} 0 & 0.0001 & -0.0015 & 0.0499 \\ 0 & 0.0000 & -0.0015 & 0.0509 \\ 0.0001 & -0.0015 & 0.0499 & -1.6077 \\ 0.0000 & -0.0015 & 0.0509 & -1.6384 \end{bmatrix}$$

Le rang de la matrice de commandabilité est de 4, donc le couple (A,B) est commandable.

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -32.3450 & 0 & 0 \\ 0 & 0 & 7.3932 & 0 \\ 0 & 0 & 0 & -5.2704 \end{bmatrix}$$

Les valeurs propres de la matrice dyanmique A sont : [0 -32.3450 7.3932 -5.2704]

$$Ob = \begin{bmatrix} 0.0010 & 0 & 0 & 0 \\ 0 & 0.0010 & 0 & 0 \\ 0 & 0 & 0.0010 & 0 \\ 0 & 0 & 0 & 0.0010 \\ 0 & 0.0814 & -0.0288 & -0.0009 \\ 0 & 0.1221 & -0.0277 & -0.0014 \\ 0 & -2.4602 & 0.8567 & 0.1096 \\ 0 & -2.4284 & 0.8383 & 0.1499 \end{bmatrix}$$

Le rang de la matrice d'observabilité est de 4, donc le couple (A,C) est observable. Le modèle en boucle ouverte s'écrit :

$$\frac{51.83s^2 + 25.94s - 2266}{s^4 + 30.22s^3 - 107.6s^2 - 1260s} \quad (3)$$

$$\frac{49.88s + 2.229.10^{-13}}{s^4 + 30.22s^3 - 107.6s^2 - 1260s} \quad (4)$$

Tout système d'ordre 2 est régi par une équation différentielle de l'ordre 2.

5 Correction du procedé

Pour régler l'instabilité du système au point critique haut, nous allons proposer un retour d'état avec un vecteur de gain K .

Cependant, nous cherchons des valeurs propres à partie réelle négative et une paire de pôle complexe

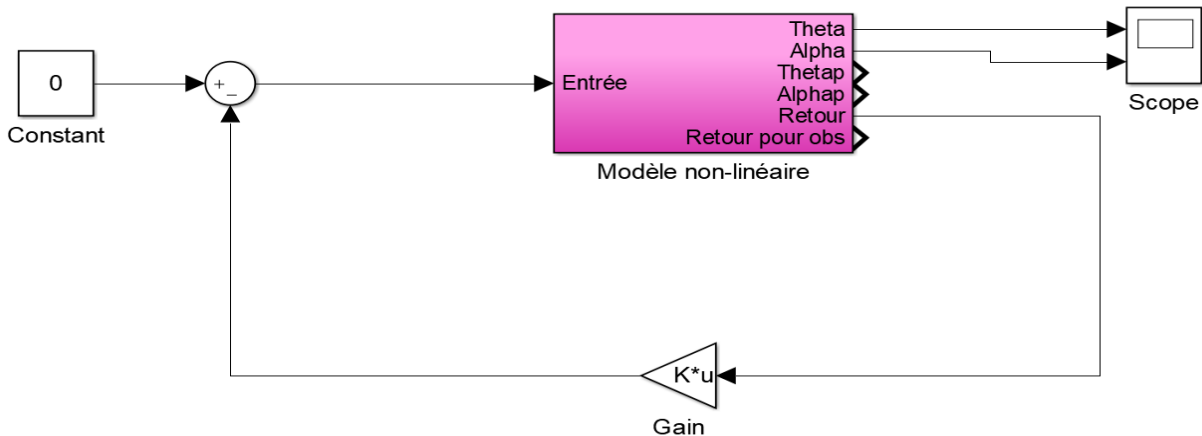


FIGURE 6 – Simulation avec le modèle non linéaire

conjugué prévoyant des oscillations au cas où le pendule n'arrive pas à rester maintenu vers le haut. Pour cela, nous avons choisi les pôles suivants $[-108 \quad -0.3 \quad -4.3 + 2.2i \quad -4.3 - 2.2i]^T$

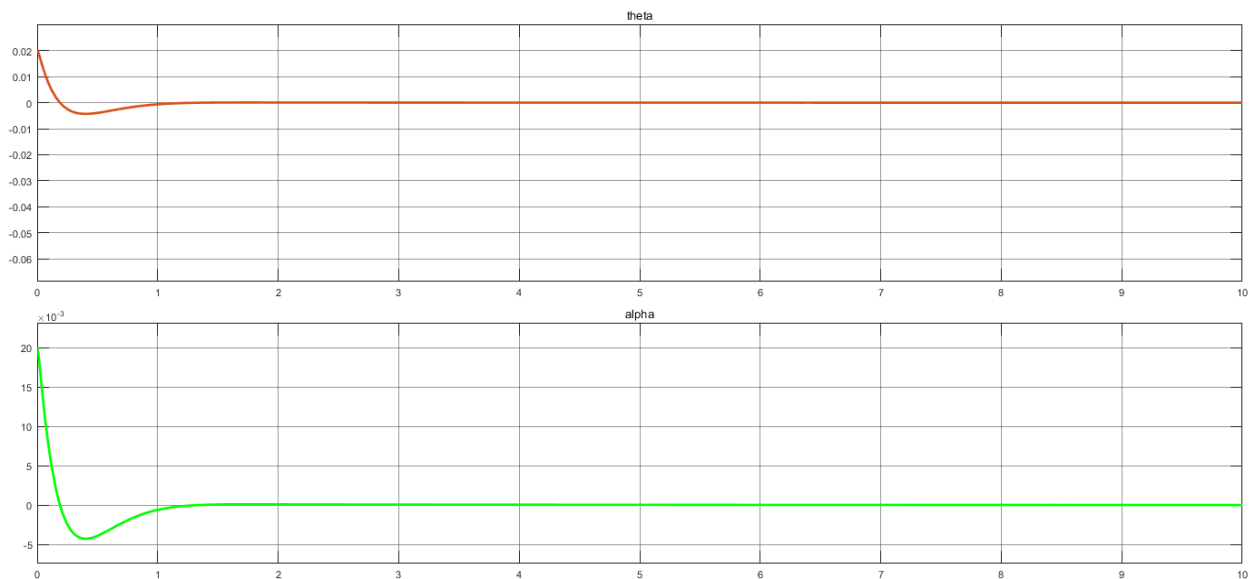


FIGURE 7 – Simulation avec le modèle linéarisé

α et θ semblent converger vers 0, notre modèle nous semble être stable. Après avoir simulé sur le modèle non linéaire, nous pouvons tester sur le procédé physique.

Résultat : Le pendule maintient sa position vers le haut mais cette stabilité n'est pas maintenue à

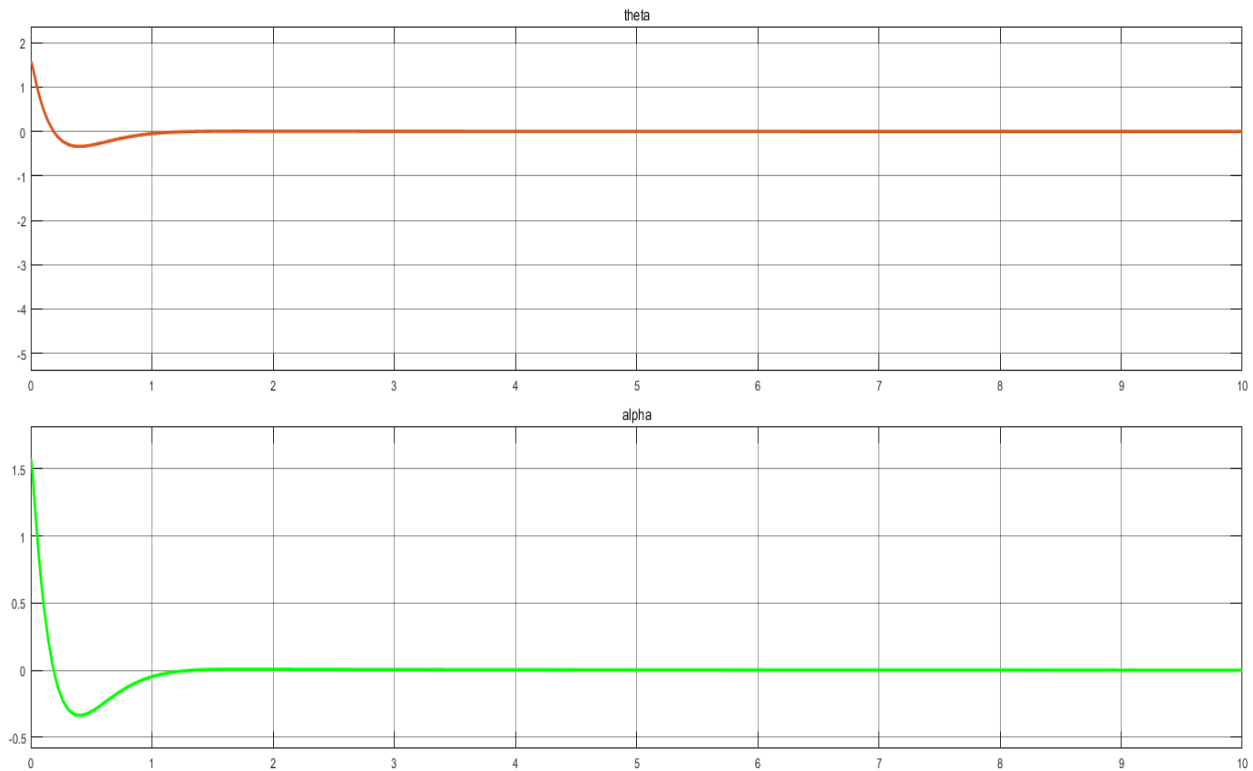


FIGURE 8 – Simulation avec le modèle non linéaire

partir d'une valeur de l'angle θ qui diverge.

Solution : Nous allons mettre un intégrateur en place pour régler ce problème.

```

%%%%%%%%%%%%%% Retour d'tat - point haut%%%%%%%%%%%%%%
Pdes_up=[-108 ; -0.3 ; -4.3+2.2i ; -4.3-2.2i] ;
Kup=acker(A,B,Pdes_up)
%% Rglage de Te et de l'initialisation
Te=0.02 ;
Theta_init=0 ;
Alpha_init_n1=0.02 ; %Origine en haut
Alpha_init_sup=0.002 ; %Origine en haut
Alpha_init_inf=0 ; %Origine en bas

```

Ajout d'un intégrateur

Nous allons ajouter un intégrateur sur θ pour assurer une stabilité plus robuste au point haut. Nous serons ramener alors à ajouter un état dans notre vecteur d'état.

Notre système s'écrit alors :

$$\begin{cases} \dot{x} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} x + \begin{bmatrix} B & 0 \\ 0 & 1 \end{bmatrix} u \\ y = \begin{bmatrix} C & 0 \end{bmatrix} x \end{cases}$$

Sur SIMULINK, notre modèle deviendra le suivant :

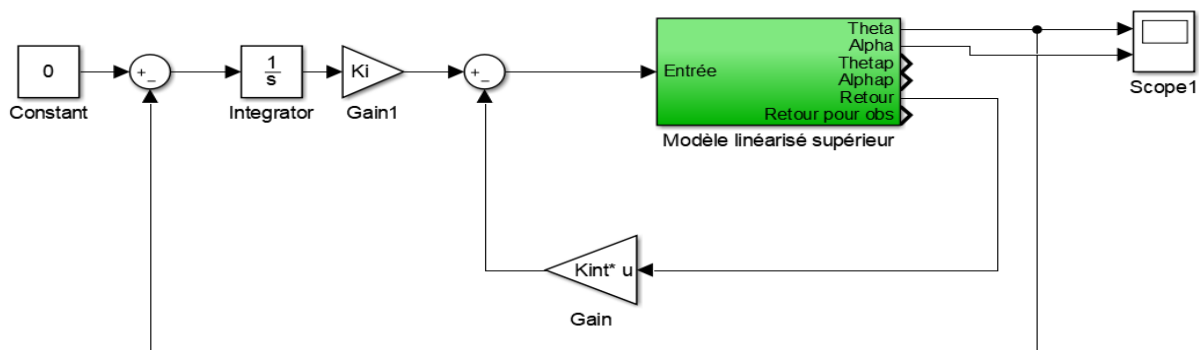


FIGURE 9 – Simulation avec le modèle non linéaire

```
% Avec intgrateur :
Aint= [Aup zeros(4,1) ;-1 0 0 0 0]
Bint=[Bup ;0]
```

En executant ce code sur Maltab, nous aurons les nouvelles matrices suivantes :

$$A_{int} = \begin{bmatrix} 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 81.4033 & -28.8249 & -0.9319 & 0 \\ 0 & 122.0545 & -27.7372 & -1.3972 & 0 \\ -1.0000 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_{int} = \begin{bmatrix} 0 \\ 0 \\ 51.8323 \\ 49.8764 \\ 0 \end{bmatrix}$$

```
%% Retour d'tat avec int
```

```

Pdes_up_int = [-108 ; -4.3+2.25i ; -4.3-2.25i ; -0.4+0.35i ; -0.4-0.35i]
Kint=acker(Aint,Bint,Pdes_up_int)
Ki=Kint(length(Kint)) ;
Kint=Kint(1 :(length(Kint)-1)) ;

```

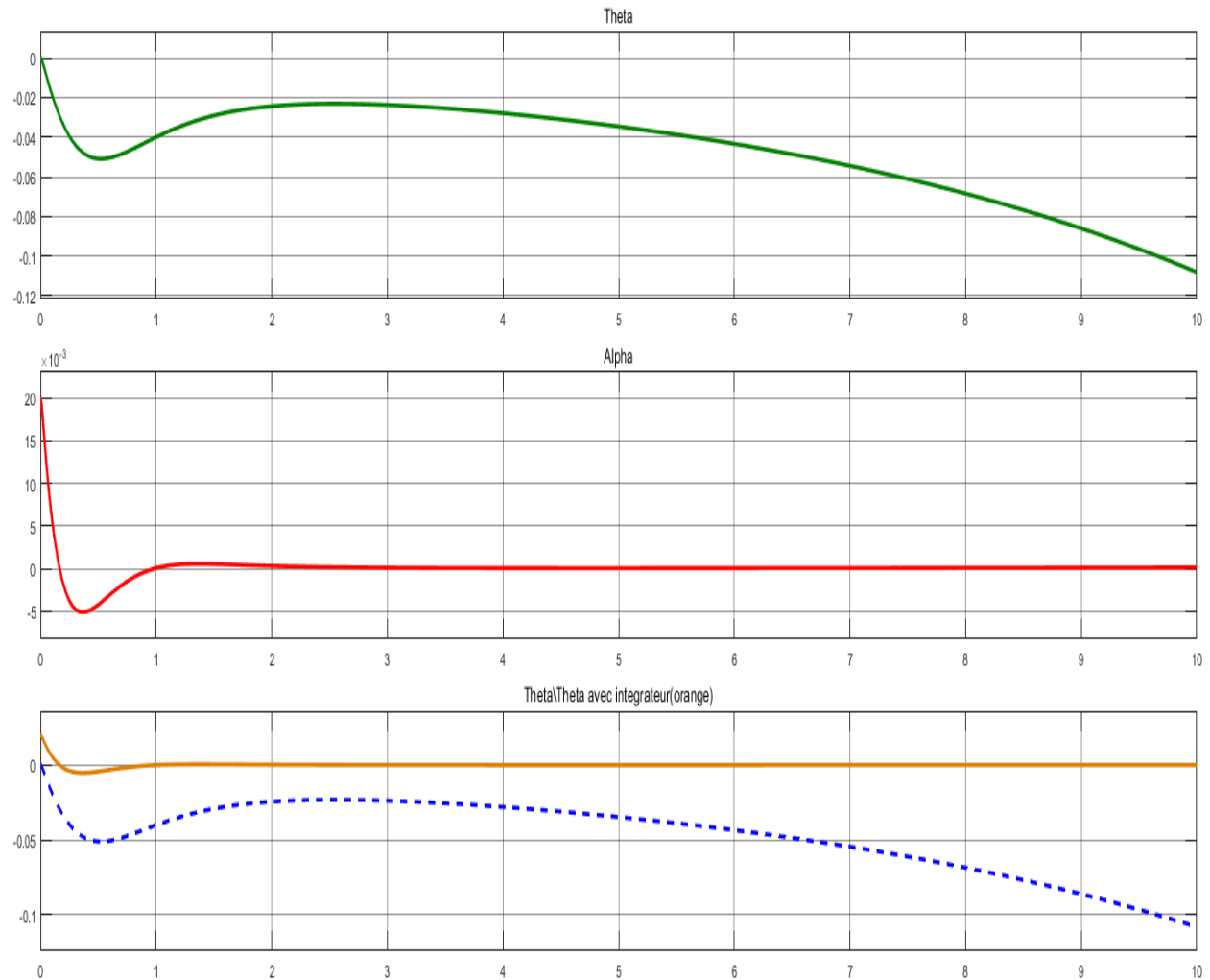


FIGURE 10 – Retour d'état avec intégrateur

Nous remarquons que θ finit par converger vers 0.

En simulant sur le procédé réel, nous remarquons que le pendule maintient sa position vers le haut et que θ ne diverge plus.

Nous avons réussi à obtenir une stabilité robuste avec rejet de perturbation.

6 Mode Switching

Pour cette partie, nous allons essayer de passer du point critique bas au point critique bas. On va alors faire un switch vers la position d'équilibre supérieur.

Ce phénomène était présent avant de mettre en place le retour d'état, le point d'équilibre haut était instable, nous faisons alors un switch vers le point d'équilibre bas stable. Nous avons décidé de faire le contraire et de déstabiliser le point bas pour stabiliser le point haut.

Pour cela, nous allons chercher à déstabiliser le pendule en position bas pour le stabiliser en position haut.

Nous devons alors mettre une paire de pôles complexes conjugués à partie réelle strictement positive, en but de déstabiliser le point haut. D'un autre côté nous allons mettre des pôles à partie réelle négative pour stabiliser en haut. Nous avons procédé en simulant divers valeurs propres.

Finalement, nous avons choisi comme pôles désirés : $[-108 \ -0.3 \ 3+10i \ 3-10i]^T$

```
%% Swing
Pdes_swing=[-108 -0.3 3+10i 3-10i] ;
Kswing=[1.5583 11.5857 4.5848 3.2635]
```

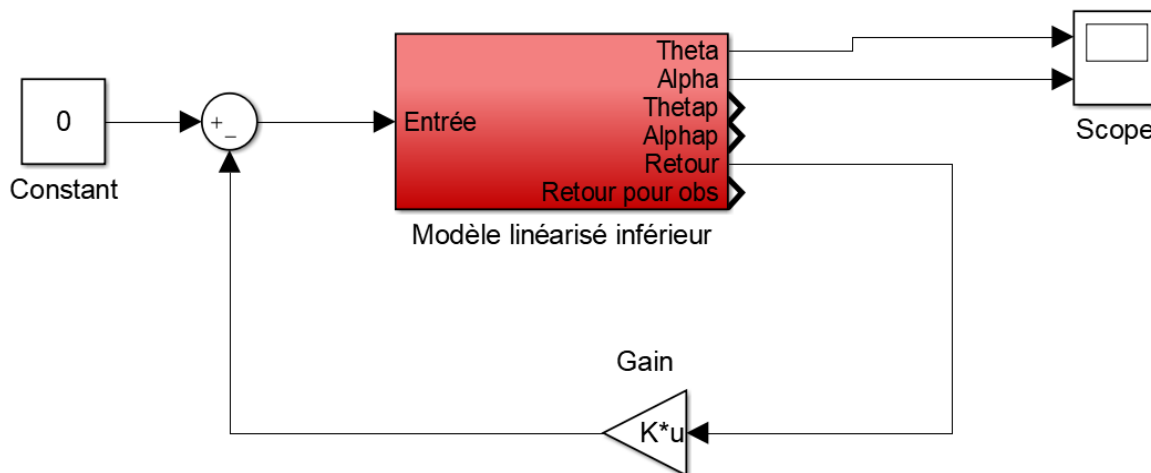
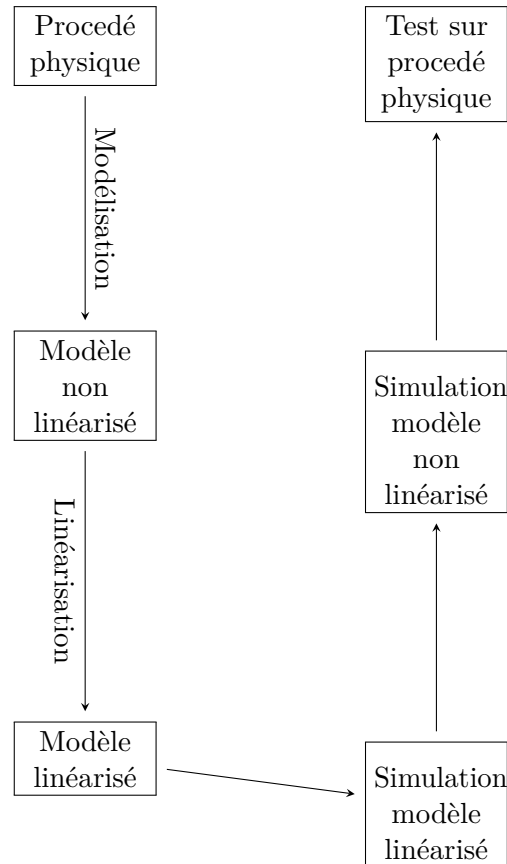


FIGURE 11 – Retour d'état avec intégrateur

7 Conclusion

A la fin de ce TP, nous avons mis en place une stratégie de travail face à un modèle non linéaire.



Nous avons dans un premier temps réalisé une analyse du procédé réel, en essayant de prévoir les divers points d'équilibres, et nous avons vérifiés notre intuition à travers la théorie et différentes simulations, mais nous nous sommes rendu compte que les 2 approches n'étaient pas compatibles.

Pour essayer de coller la théorie et la pratique, nous avons réalisé une analyse du système en boucle ouverte pour repérer ses défauts. Nous avons constaté qu'un des points d'équilibres n'étaient pas stable.

Pour régler cette instabilité, nous avons réalisé une commande par retour d'état. En testant sur le procédé physique, nous avons remarqué que le point haut était stable mais pas robuste, on arrivait à perdre la stabilité.

Pour rendre le point haut plus robustement stable, nous avons ajouté un intégrateur, sur notre ancien modèle. Par conséquent, nous avons obtenu un modèle stable robuste, les 2 points d'équilibre sont vérifiés pour $\alpha = 0$ et $\alpha = \pi$.

Enfin, dans la dernière partie nous avons vu comment on peut faire un switch entre un mode et un autre. Pour cela nous avons procédé par intuition, cependant ça aurait été plus intéressant de chercher théoriquement des pôles spécifiques qui nous assurent un asservissement assez performant et robuste, en ayant un système rapide mais qui n'oscille pas trop avec une erreur assez faible.

8 Script MATLAB

```

%% Point haut
% Espace d'tat, point d'equilibre haut
Jt = Jr*Jp + Mp*(Lp/2)^2*Jr + Jp*Mp*Lr^2 ;
gamma=(eta_g*eta_m*Kg*kt)/Rm ;

Aup = [0 0 1 0 ;
        0 0 0 1 ;
        0 Mp^2*(Lp/2)^2*Lr*g/Jt ((Jp+Mp*(Lp/2)^2)*(-Dr-Kg*km*gamma))/Jt -Mp*(Lp/2)*Lr*Dp/Jt ;
        0 Mp*g*(Lp/2)*(Jr+Mp*Lr^2)/Jt (Mp*(Lp/2)*Lr)*(-Dr-gamma*Kg*km)/Jt
        -Dp*(Jr+Mp*Lr^2)/Jt] ;

Bup = [0 ; 0 ; ((Jp+Mp*(Lp/2)^2)*gamma)/Jt ; (Mp*(Lp/2)*Lr)*gamma/Jt] ;
Cup = eye(2,4) ;
Dup = zeros(2,1) ;

% Avec intgrateur : A COMPLETER
Aint= [Aup zeros(4,1) ;-1 0 0 0 0]
Bint=[Bup ;0]

%% Point bas
% Espace d'tat, point d'equilibre bas
Adown = [0 0 1 0 ;
          0 0 0 1 ;
          0 Mp^2*(Lp/2)^2*Lr*g/Jt ((Jp+Mp*(Lp/2)^2)*(-Dr-Kg*km*gamma))/Jt
          -Mp*(Lp/2)*Lr*Dp/Jt ;
          0 -Mp*g*(Lp/2)*(Jr+Mp*Lr^2)/Jt (-Mp*(Lp/2)*Lr)*(-Dr-gamma*Kg*km)/Jt
          Dp*(Jr+Mp*Lr^2)/Jt ;]

Bdown = [0 ; 0 ; ((Jp+Mp*(Lp/2)^2)*gamma)/Jt ; (-Mp*(Lp/2)*Lr)*gamma/Jt] ;
Cdown = eye(2,4) ;
Ddown = zeros(2,1) ;

```

```

%% Point haut
% Espace d'tat, point d'equilibre haut
Jt = Jr*Jp + Mp*(Lp/2)^2*Jr + Jp*Mp*Lr^2 ;
gamma=(eta_g*eta_m*Kg*kt)/Rm ;

Aup = [0 0 1 0 ;
       0 0 0 1 ;
       0 Mp^2*(Lp/2)^2*Lr*g/Jt ((Jp+Mp*(Lp/2)^2)*(-Dr-Kg*km*gamma))/Jt -Mp*(Lp/2)*Lr*Dp/Jt ;
       0 Mp*g*(Lp/2)*(Jr+Mp*Lr^2)/Jt (Mp*(Lp/2)*Lr)*(-Dr-gamma*Kg*km)/Jt
       -Dp*(Jr+Mp*Lr^2)/Jt] ;

Bup = [0 ; 0 ; ((Jp+Mp*(Lp/2)^2)*gamma)/Jt ; (Mp*(Lp/2)*Lr)*gamma/Jt] ;
Cup = eye(2,4) ;
Dup = zeros(2,1) ;

% Avec intgrateur : A COMPLETER
Aint= [Aup zeros(4,1) ; -1 0 0 0 0]
Bint=[Bup ; 0]

%% Point bas
% Espace d'tat, point d'equilibre bas
Adown = [0 0 1 0 ;
         0 0 0 1 ;
         0 Mp^2*(Lp/2)^2*Lr*g/Jt ((Jp+Mp*(Lp/2)^2)*(-Dr-Kg*km*gamma))/Jt
         -Mp*(Lp/2)*Lr*Dp/Jt ;
         0 -Mp*g*(Lp/2)*(Jr+Mp*Lr^2)/Jt (-Mp*(Lp/2)*Lr)*(-Dr-gamma*Kg*km)/Jt
         Dp*(Jr+Mp*Lr^2)/Jt ;]

Bdown = [0 ; 0 ; ((Jp+Mp*(Lp/2)^2)*gamma)/Jt ; (-Mp*(Lp/2)*Lr)*gamma/Jt] ;
Cdown = eye(2,4) ;
Ddown = zeros(2,1) ;

```

```

%% Etude du pendule invers
%
clear all ;
%
%% Configuration de la maquette
%% Configuration du SRV02
% Configuration des engrenages : choisir 'HIGH' ou 'LOW'
EXT_GEAR_CONFIG = 'HIGH' ;
% Type du codeur : choisir 'E' ou 'EHR'
ENCODER_TYPE = 'E' ;
% Le SRV02 est-il quip d'un tachymtre ? : choisir 'YES' ou 'NO'
TACH_OPTION = 'NO' ;
% Type de charge : choisir 'NONE', 'DISC', ou 'BAR'
LOAD_TYPE = 'ROTPEN' ;
% Gain d'amplificateur utilis :
% VoltPAQ-X2 choisir K_AMP 3
% Sinon 1
K_AMP = 1 ;
% Type d'amplificateur utilis : dfinir soit 'VoltPAQ', 'UPM_1503', 'UPM_2405', or 'Q3'
AMP_TYPE = 'UPM_1503' ;
%
%% Gnration des paramtres
% Calcul des paramtres du moteur en fonction des options choisies
[ Rm, kt, km, Kg, eta_g, Beq, Jm, Jeq, eta_m, K_POT, K_TACH, K_ENC, VMAX_AMP, IMAX_AMP ]
    = config_srv02( EXT_GEAR_CONFIG, ENCODER_TYPE, TACH_OPTION, AMP_TYPE, LOAD_TYPE ) ;
% Charge les paramtres du bras
[ g, Mr, Lr, lr, Jr, Dr ] = config_sp( 'ROTARY_ARM', 'ROTPEN-E' ) ;
% Charge les paramtres du pendule
[ g, Mp, Lp, lp, Jp, Dp ] = config_sp( 'MEDIUM_12IN', 'ROTPEN-E' ) ;
%
%% Rglage de Te et de l'initialisation
Te=0.02 ;
Theta_init=0 ;
Alpha_init_n1=0.02 ;%Origine en haut
Alpha_init_sup=0.02 ;%Origine en haut
Alpha_init_inf=pi ;%Origine en bas
%% Appel des fichiers

Espace_etat_ABCD ; % A COMPLETER gnre {A,B,C,D}{up,int,down}
Pseudo_derivateur ; % NE PAS MODIFIER
Commande_point_haut ; % A COMPLETER POUR GENERER Kup, Kint et Ki, Kswing

%% Simulations / Rponses temporelles ...

```

```

%% %% Differentes commande pour le point haut

%% Retour d'tat - point haut
Pdes_up=[-108 ; -0.3 ; -4.3+2.2i ; -4.3-2.2i] ;
Kup=[-0.3335 23.2307 -1.7978 3.6061] ;

%% Retour d'tat avec int

```

```
Pdes_up_int =[-108 ; -4.3+2.25i ; -4.3-2.25i ; -0.4+0.35i ; -0.4-0.35i] ;  
Kint = [-1.0203 25.2537 -2.0482 3.8764 0.3171] ;  
Ki=Kint(length(Kint)) ;  
Kint=Kint(1 :(length(Kint)-1)) ;
```

```
%% Swing
```

```
Pdes_swing=[-108 -0.3 3+10i 3-10i] ;  
Kswing=[1.5583 11.5857 4.5848 3.2635]
```
