



LA-IIS-STI Laboratoire d'automatique

A Linear State Feedback Controller for a SpiderCrane

Dominick Vanthienen

Assistant : Davide Bucciari
Professeur : Dominique Bonvin

Semestre d'hiver 2006/2007

Table des matières

1	Introduction	3
2	Description de la grue	3
3	Création d'un modèle mathématique	4
3.1	Notations	4
3.1.1	Paramètres du système	4
3.1.2	Coordonnées du système	4
3.2	Choix d'un système de coordonnées	4
3.3	Modèle dynamique : les formules de Lagrange	5
3.3.1	Formulation générale	5
3.3.2	Lagrangien	6
3.3.3	Expression des forces	6
3.3.4	Résoudre les équations	7
4	Simulation sans rétroaction	9
5	Linéarisation	9
6	Gouvernabilité	10
7	Commande par placement de valeurs propres	11
7.1	Régulateur proportionnel-différentiel PD	11
7.1.1	Théorie	11
7.1.2	Implémentation en Matlab	11
7.1.3	Pôles	12
7.1.4	Points de simulation	12
7.1.5	Code des couleurs	13
7.1.6	Premier simulation	13
7.1.7	Influence de la longueur du câble	15
7.1.8	Seconde simulation	15
7.2	Régulateur proportionnel-intégral-différentiel PID	17
7.2.1	Théorie	17
7.2.2	Implémentation en Matlab	17
7.2.3	Pôles	18
7.2.4	Points de simulation	18
7.2.5	Premier simulation	18
7.2.6	Deuxième simulation	19
7.2.7	Troisième simulation	19

8	Les diagrammes de Bode	21
8.1	Les fonctions de transfert	21
8.2	Diagrammes de Bode	21
9	Conclusion	23

1 Introduction

On a développé une nouvelle grue efficace, SpiderCrane, pour améliorer le taux de transfert des conteneurs. Le contrôle de cette grue était développé par la Laboratoire d'Automatique de l'EPFL. Afin de tester la comportement de la grue, on a construit une modèle de cette grue à la Laboratoire d'Automatique.

Les buts de ce projet sont :

- de construire un modèle mathématique de la grue en utilisant les formules de Lagrange
- le synthse d'une loi de contrôle linéarisée avec rétroaction et la comparaison avec la loi de contrôle non-linéaire existante

2 Description de la grue

La grue se compose de trois pylônes avec au sommet de chaque pylône une poulie permettant de faire coulisser un câble. Ces trois câbles sont rattachés à un anneau. En jouant sur la longueur de ces câbles, on peut déplacer la charge entre les pylônes. La potence qui était utilisé dans une version antérieure n'est pas considéré dans ce projet. Tous les câbles sont commandés en force et sont équipés d'encodeurs permettant de mesurer leur longueur et vitesse.

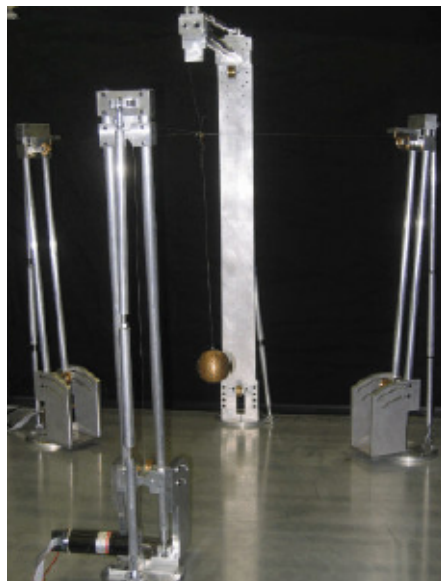


FIG. 1 – SpiderCrane

3 Création d'un modèle mathématique

3.1 Notations

3.1.1 Paramètres du système

m	=	masse de la charge
m_0	=	masse de l'anneau
$x_{pa1}, x_{pa2}, x_{pa3}$	=	position de la poulie fixe a
$x_{pb1}, x_{pb2}, x_{pb3}$	=	position de la poulie fixe b
$x_{pc1}, x_{pc2}, x_{pc3}$	=	position de la poulie fixe c

3.1.2 Coordonnées du système

$x_1(t), x_2(t), x_3(t)$	=	position de la charge
$x_{10}(t), x_{20}(t), x_{30}(t)$	=	position initial de la charge
$x_{1ref}(t), x_{2ref}(t), x_{3ref}(t)$	=	position de la charge où on veut aller
$\bar{x}_1(t), \bar{x}_2(t), \bar{x}_3(t)$	=	position de la charge autour du quel on linéarise
$x_{01}(t), x_{02}(t), x_{03}(t)$	=	position de l'anneau
L_1	=	longueur du câble reliant la poulie a à l'anneau
L_2	=	longueur du câble reliant la poulie b à l'anneau
L_3	=	longueur du câble reliant la poulie c à l'anneau
L_5	=	longueur du câble reliant l'anneau à la charge

3.2 Choix d'un système de coordonnées

Le premier système de coordonnées considérées, utilise des coordonnées sphériques. Le vecteur concerné est projeté orthogonal sur le plan (x_1, x_2) . Les coordonnées sont x_1, x_2, x_3 , l'angle θ entre le vecteur L_5 et sa projection et l'angle ϕ entre la projection et x_1 . (figure 2) Ce système est facile à interpréter mais il y a une singularité quand θ est nonante degrés, l'angle qu'on veut acquérir quand le système est stable. En ce cas on peut changer ϕ sans qu'il y ait un effet sur la position.

Un deuxième système étudié utilise des coordonnées sphériques orthogonales. Ce système projette orthogonal le vecteur sur deux plans, (x_1, x_3) et (x_2, x_3) . Les coordonnées sont x_1, x_2, x_3 , l'angle θ entre le premier projection et x_1 et l'angle ϕ entre la deuxième projection et x_2 . (figure 2) L'avantage de ce deuxième système est qu'on n'a pas la singularité comme dans le cas des coordonnées sphériques. Par contre, le système est difficile à

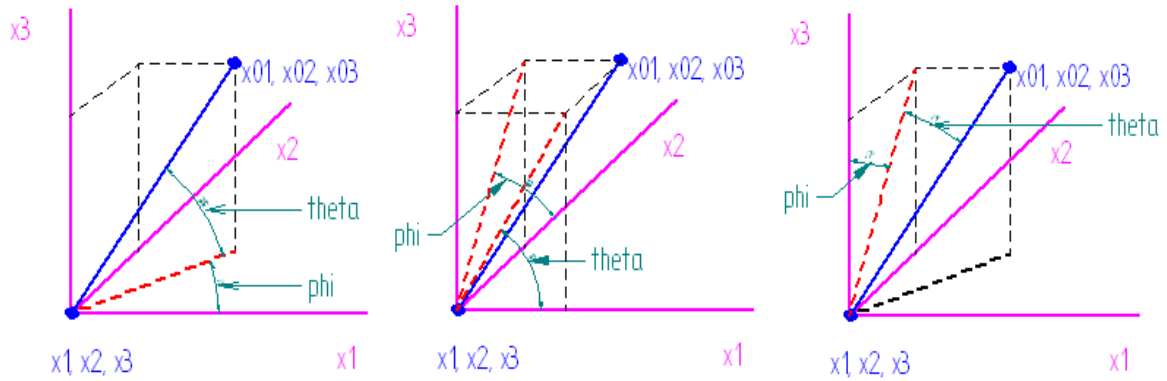


FIG. 2 – Systèmes de coordonnées

interpréter et augmente trop la complexité du système.

Le troisième système de coordonnées considérées utilise aussi des coordonnées sphériques comme le premier système mais on projette maintenant le vecteur sur le plan (x_2, x_3) . (figure 2) Dans ce cas la singularité se trouve aussi θ égale à nonante degrés mais cette singularité correspond avec une direction du vecteur horizontale, qui n'est pas utilisé dans des cas réels. On utilise le troisième système de coordonnées $x_1, x_2, x_3, \theta, \phi$ pour la grue.

3.3 Modèle dynamique : les formules de Lagrange

3.3.1 Formulation générale

La méthode de Lagrange permet d'obtenir les équations dynamiques. La formule générale s'écrit :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \sum F_{i,e} \quad i = 1 \dots n \quad (1)$$

- L est le Lagrangien qui est égale à la différence entre l'énergie cinétique (W_k) et l'énergie potentielle (W_p).

$$L = W_k - W_p \quad (2)$$

- q représente les coordonnées généralisées.

Ces coordonnées sont la nombre de coordonnées minimales qu'on a besoin pour décrire le système. Dans ce cas-ci :

$$q = (q_1, q_2, q_3, q_4, q_5) = (x_1, x_2, x_3, \theta, \phi)$$

- n est égale au nombre de coordonnées généralisées, 5 pour la grue considérée.
- $F_{i,e}$ est la composante des forces externes sur le système dans la direction de la coordonnée q_i .

3.3.2 Lagrangien

L'addition des énergies cinétiques de chaque partie mobile donne l'énergie totale de la grue (W_k).

$$W_k = \frac{1}{2} \sum_{i=1}^3 (m\dot{x}_i^2 + m_0\dot{x}_{0i}^2) \quad (3)$$

L'addition des énergies potentielles de la charge avec celle de l'anneau donne l'énergie totale de la grue (W_p). (g est la constante gravitationnelle)

$$W_p = mgx_3 + m_0gx_{03} \quad (4)$$

3.3.3 Expression des forces

Pour trouver les forces dans les directions des coordonnées on utilise le principe du travail virtuel.

$$T_i \delta q_i = \sum_j \vec{F}_j \delta \vec{r}_{ij} \quad (5)$$

T_i est si déterminé que $T_i \delta q_i$ est égale au travail virtuel des vraies forces, quand le système bouge tellement que seulement q_i change.

Cette principe donne les équations suivantes :

$$\begin{aligned} \delta W_{x1} &= (T_{01} \frac{\delta x_{01}}{\delta x_1} + T_{02} \frac{\delta x_{02}}{\delta x_1} + T_{03} \frac{\delta x_{03}}{\delta x_1}) \delta x_1 \\ &= T_{01} \delta x_1 \\ \delta W_{x2} &= (T_{01} \frac{\delta x_{01}}{\delta x_2} + T_{02} \frac{\delta x_{02}}{\delta x_2} + T_{03} \frac{\delta x_{03}}{\delta x_2}) \delta x_2 \\ &= T_{02} \delta x_2 \\ \delta W_{x3} &= (T_{01} \frac{\delta x_{01}}{\delta x_3} + T_{02} \frac{\delta x_{02}}{\delta x_3} + T_{03} \frac{\delta x_{03}}{\delta x_3}) \delta x_3 \\ &= T_{03} \delta x_3 \\ \delta W_{\theta} &= (T_{01} \frac{\delta x_{01}}{\delta \theta} + T_{02} \frac{\delta x_{02}}{\delta \theta} + T_{03} \frac{\delta x_{03}}{\delta \theta}) \delta \theta \\ &= (T_{01} L_5 \cos(\theta) - T_{02} L_5 \sin(\theta) \sin(\phi) - T_{03} L_5 \sin(\theta) \cos(\phi)) \delta \theta \\ \delta W_{\phi} &= (T_{01} \frac{\delta x_{01}}{\delta \phi} + T_{02} \frac{\delta x_{02}}{\delta \phi} + T_{03} \frac{\delta x_{03}}{\delta \phi}) \delta \phi \\ &= (T_{02} L_5 \cos(\theta) \cos(\phi) - T_{03} L_5 \cos(\theta) \sin(\phi)) \delta \phi \end{aligned}$$

Ces forces T_{01}, T_{02}, T_{03} sont dans les directions des coordonnées x_1, x_2, x_3 , mais les forces T_1, T_2, T_3 qu'on applique au système sont dans la direction des fils. On doit recalculer ces forces vers des forces T_1, T_2, T_3 avec une direction depuis l'anneau vers les poulies.

$$\text{pour } T_1 \Rightarrow (x_{pa1} - x_{01}, x_{pa2} - x_{02}, x_{pa3} - x_{03}) \quad (6)$$

$$\text{pour } T_2 \Rightarrow (x_{pb1} - x_{01}, x_{pb2} - x_{02}, x_{pb3} - x_{03}) \quad (7)$$

$$\text{pour } T_3 \Rightarrow (x_{pc1} - x_{01}, x_{pc2} - x_{02}, x_{pc3} - x_{03}) \quad (8)$$

Expression des forces T_1, T_2, T_3 :

$$(T_{1,1}, T_{1,2}, T_{1,3}) = \frac{(xpa1 - x01, xpa2 - x02, xpa3 - x03) |T_1|}{\sqrt{(xpa1 - x01)^2 + (xpa2 - x02)^2 + (xpa3 - x03)^2}} \quad (9)$$

$$(T_{2,1}, T_{2,2}, T_{2,3}) = \frac{(xpb1 - x01, xpb2 - x02, xpb3 - x03) |T_2|}{\sqrt{(xpb1 - x01)^2 + (xpb2 - x02)^2 + (xpb3 - x03)^2}} \quad (10)$$

$$(T_{3,1}, T_{3,2}, T_{3,3}) = \frac{(xpc1 - x01, xpc2 - x02, xpc3 - x03) |T_3|}{\sqrt{(xpc1 - x01)^2 + (xpc2 - x02)^2 + (xpc3 - x03)^2}} \quad (11)$$

Expression des forces T_{01}, T_{02}, T_{03} :

$$T_{01} = T_{1,1} + T_{2,1} + T_{3,1} \quad (12)$$

$$T_{02} = T_{1,2} + T_{2,2} + T_{3,2} \quad (13)$$

$$T_{03} = T_{1,3} + T_{2,3} + T_{3,3} \quad (14)$$

$$\begin{aligned} T_{01} &= \frac{(xpa1-x01)|T_1|}{\sqrt{(xpa1-x01)^2+(xpa2-x02)^2+(xpa3-x03)^2}} \\ &+ \frac{(xpb1-x01)|T_2|}{\sqrt{(xpb1-x01)^2+(xpb2-x02)^2+(xpb3-x03)^2}} \\ &+ \frac{(xpc1-x01)|T_3|}{\sqrt{(xpc1-x01)^2+(xpc2-x02)^2+(xpc3-x03)^2}} \\ T_{02} &= \frac{(xpa2-x02)|T_1|}{\sqrt{(xpa1-x01)^2+(xpa2-x02)^2+(xpa3-x03)^2}} \\ &+ \frac{(xpb2-x02)|T_2|}{\sqrt{(xpb1-x01)^2+(xpb2-x02)^2+(xpb3-x03)^2}} \\ &+ \frac{(xpc2-x02)|T_3|}{\sqrt{(xpc1-x01)^2+(xpc2-x02)^2+(xpc3-x03)^2}} \\ T_{03} &= \frac{(xpa3-x03)|T_1|}{\sqrt{(xpa1-x01)^2+(xpa2-x02)^2+(xpa3-x03)^2}} \\ &+ \frac{(xpb3-x03)|T_2|}{\sqrt{(xpb1-x01)^2+(xpb2-x02)^2+(xpb3-x03)^2}} \\ &+ \frac{(xpc3-x03)|T_3|}{\sqrt{(xpc1-x01)^2+(xpc2-x02)^2+(xpc3-x03)^2}} \end{aligned}$$

La remplacement des forces T_{01}, T_{02}, T_{03} par les formules 12, 13 et 14 dans les équations de Lagrange, nous donne les équations en fonction des forces qu'on applique, les coordonnées généralisées et ses dérivées et le temps. Les formules qu'on obtient sont mis en annexe pour rester clair. (annexe 1)

3.3.4 Résoudre les équations

Par résoudre le système vers les accélérations on obtient les équations suivantes :

$$\begin{aligned}
\ddot{x}_1(t) &= \frac{\sin(\theta(t))(\cos(\theta(t))(T_{03} \cos(\phi(t)) + T_{02} \sin(\phi(t)))}{m+mo} \\
&+ \frac{\sin(\theta(t))(T_{01} \sin(\theta(t)) + L5mo(\cos^2(\theta(t))\phi'(t)^2 + \theta'(t)^2))}{m+mo} \\
\ddot{x}_2(t) &= \frac{\cos(\theta(t)) \sin(\phi(t))(\cos(\theta(t))(T_{03} \cos(\phi(t)) + T_{02} \sin(\phi(t)))}{m+mo} \\
&+ \frac{\cos(\theta(t)) \sin(\phi(t))(T_{01} \sin(\theta(t)) + L5mo(\cos^2(\theta(t))\phi'(t)^2 + \theta'(t)^2))}{m+mo} \\
\ddot{x}_3(t) &= \frac{-4g(m+mo) + T_{03} + T_{03} \cos(2\phi(t)) + 2 \cos(\phi(t)) \cos(2\theta(t))(T_{03} \cos(\phi(t)) + T_{02} \sin(\phi(t)))}{4(m+mo)} \\
&+ \frac{T_{02} \sin(2\phi(t)) + 2T_{01} \cos(\phi(t)) \sin(2\theta(t)) + 4L5mo \cos(\phi(t)) \cos(\theta(t))(\cos^2(\theta(t))\phi'(t)^2 + \theta'(t)^2)}{4(m+mo)} \\
\ddot{\theta}(t) &= -\frac{L5mo \sin(2\theta(t))\phi'(t)^2 - 2T_{01} \cos(\theta(t))}{2L5mo} \\
&- \frac{2(T_{03} \cos(\phi(t)) + T_{02} \sin(\phi(t))) \sin(\theta(t))}{2L5mo} \\
\ddot{\phi}(t) &= \frac{\sec(\theta(t))(T_{02} \cos(\phi(t)) - T_{03} \sin(\phi(t)))}{L5mo} \\
&+ 2 \tan(\theta(t))\phi'(t)\theta'(t)
\end{aligned}$$

4 Simulation sans rétroaction

Les cinq équations de Lagrange sont du deuxième ordre. Elles sont ramenées à dix équations du premier ordre par introduction des nouvelles variables q_{id} qui expriment :

$$q_{id} = \dot{q}_i$$

$$\dot{q}_{id} = \ddot{q}_i$$

Les dix équations sont placées dans un m-file en Matlab, diffEq. En intégrant ces équations, on obtient les coordonnées généralisées et leurs dérivées. Celles-ci sont les équations d'état du système, $x = (x_1, \dot{x}_1, x_2, \dot{x}_2, x_3, \dot{x}_3, \theta, \dot{\theta}, \phi, \dot{\phi})$.

L'algorithme 'ode45' en Matlab est utilisé pour l'intégration, il va intégrer les équations numériquement par rapport au temps selon un pas variable. Quand l'intégral change beaucoup, l'algorithme va diminuer le pas de temps sinon il va l'augmenter.

Le programme montré en figure 3 est obtenu.

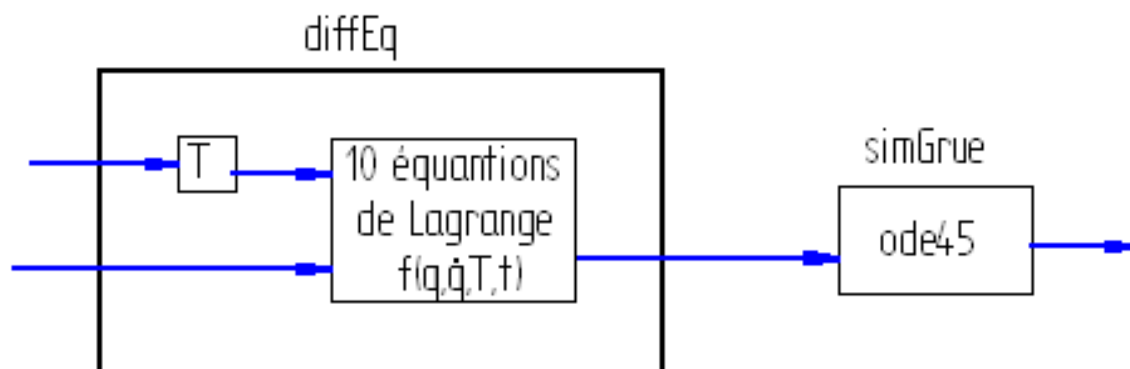


FIG. 3 – Simulation sans rétroaction

Le programme consiste de deux fichiers, diffEq et simGrue. SimGrue comporte tous les valeurs pour les variables ; les coordonnées du position ou on veut aller (q_{ref}), ses dérivées et les conditions initiales. Il lance ode45 et donne les valeurs à diffEq. DiffEq utilise ces valeurs d'abord pour calculer les forces et après pour mettre dans les équations de Lagrange.

5 Linéarisation

Jusqu'à maintenant le système est décrit avec des équations non-linéaires $f(q, \dot{q}, T, t)$. On peut linéariser ces équations autour d'un point afin d'utiliser cette linéarisation pour la construction d'un régulateur linéaire.

L'état correspondant au point autour duquel on linéarise, \bar{q} est nommé \bar{x} . Quand le système se trouve dans un autre point, il y a un autre état x . La différence entre les deux s'appelle $\tilde{x} = x - \bar{x}$. Les équations d'état, $\dot{x} = f(q, \dot{q}, T, t)$ et les équations de sortie $y = g(q, \dot{q}, T, t)$ sont linéarisées par écrire leur développement en séries de Taylor et en

négligeant les termes d'ordre supérieur à un, $p(\tilde{x}, \tilde{u})$ (équation 15). Parce que les sorties y sont égales aux états x , il n'est pas nécessaire de développer y en séries de Taylor.

$$f(x, u) = f(\bar{x}, \bar{u}) + \frac{\partial f(\bar{x}, \bar{u})}{\partial x} \tilde{x} + \frac{\partial f(\bar{x}, \bar{u})}{\partial u} \tilde{u} + p(\tilde{x}, \tilde{u}) \quad (15)$$

Où les formules $\frac{\partial f(\bar{x}, \bar{u})}{\partial x}$ et $\frac{\partial f(\bar{x}, \bar{u})}{\partial u}$ sont des matrices Jacobiennes (équations 16 et 17).

$$\frac{\partial}{\partial x} f(\bar{x}, \bar{u}) = A = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(\bar{x}, \bar{u}) & \frac{\partial}{\partial x_2} f_1(\bar{x}, \bar{u}) & \dots & \frac{\partial}{\partial x_{10}} f_1(\bar{x}, \bar{u}) \\ \frac{\partial}{\partial x_1} f_2(\bar{x}, \bar{u}) & \frac{\partial}{\partial x_2} f_2(\bar{x}, \bar{u}) & \dots & \frac{\partial}{\partial x_{10}} f_2(\bar{x}, \bar{u}) \\ \vdots & \vdots & & \vdots \\ \frac{\partial}{\partial x_1} f_{10}(\bar{x}, \bar{u}) & \frac{\partial}{\partial x_2} f_{10}(\bar{x}, \bar{u}) & \dots & \frac{\partial}{\partial x_{10}} f_{10}(\bar{x}, \bar{u}) \end{bmatrix} \quad (16)$$

$$\frac{\partial}{\partial u} f(\bar{x}, \bar{u}) = B = \begin{bmatrix} \frac{\partial}{\partial u_1} f_1(\bar{x}, \bar{u}) & \frac{\partial}{\partial u_2} f_1(\bar{x}, \bar{u}) & \frac{\partial}{\partial u_3} f_1(\bar{x}, \bar{u}) \\ \frac{\partial}{\partial u_1} f_2(\bar{x}, \bar{u}) & \frac{\partial}{\partial u_2} f_2(\bar{x}, \bar{u}) & \frac{\partial}{\partial u_3} f_2(\bar{x}, \bar{u}) \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial u_1} f_{10}(\bar{x}, \bar{u}) & \frac{\partial}{\partial u_2} f_{10}(\bar{x}, \bar{u}) & \frac{\partial}{\partial u_3} f_{10}(\bar{x}, \bar{u}) \end{bmatrix} \quad (17)$$

En substituant $\dot{x}(t)$ par $\dot{\tilde{x}}(t) + \dot{\tilde{x}}(t)$ (dérivé de la définition de \tilde{x} , on obtient les expressions 18 et 19. Avec C la matrice d'identité et D égale à zéro.

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + B\tilde{u}(t) \quad (18)$$

$$\tilde{y}(t) = C\tilde{x}(t) + D\tilde{u}(t) \quad (19)$$

La dépendance du temps de x et y ne sera pas mentionner explicitement dans les formules suivantes pour augmenter la lisibilité et la transparence des formules.

6 Gouvernabilité

La gouvernabilité exprime que, avec la condition initial pour l'état x_0 et pour un état donné x_d quelconque, il existe un entier fini N et une séquence temporelle de commandes $u(0), u(1), \dots, u(N-1)$ tels que $x(N) = x_d$ après application de cette séquence [2, p. 102]. Cette condition se transforme dans l'expression suivante :

$$C_{controle} = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B] \quad (20)$$

Le système est gouvernable quand le rang de la matrice $C_{controle}$ est égale à $n=10$ =le nombre des états. Pour ce système-ci, ça nous donne une matrice de 10×30 qui est gouvernable. (annexe 2)

7 Commande par placement de valeurs propres

7.1 Régulateur proportionnel-différentiel PD

7.1.1 Théorie

Dans ce chapitre-ci on introduit une contre-réaction. Cette contre-réaction est choisie proportionnelle à l'écart de l'état par rapport à l'état nominal. La commande correspond donc à une somme pondérée des écarts des états par les gains K_i de la commande.

$$\tilde{u} = -K\tilde{x} = -(K_1\tilde{x}_1 + K_2\tilde{x}_2 + \dots + K_{10}\tilde{x}_{10}) \quad (21)$$

Une commande d'état agit en quelque sorte comme un régulateur 'PD' généralisé [2, p. 107].

K est construit par placement des pôles, on peut choisir les pôles du système qui sont les valeurs propres de la matrice $A - BK$. Cette opération est faite par Matlab par la commande $K = \text{place}(A, B, \text{pôles})$.

7.1.2 Implémentation en Matlab

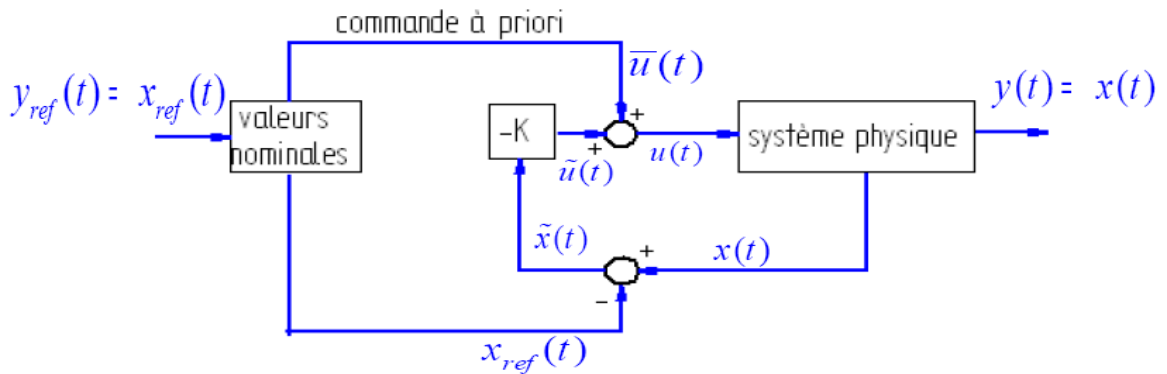


FIG. 4 – Commande d'état

Les fichiers les plus importants de l'implémentation consistent des mêmes fichiers qu'avant, `diffEq.m` et `SimGrue2.m`. Ils sont adaptés au placement des pôles (figure 4). D'abord on doit définir l'état qu'on veut obtenir x_{ref} , l'état autour duquel on linéarise \bar{x} , les conditions initiales x_0 et les pôles.

x_{ref} est utilisé pour calculer \bar{u} . Il s'agit d'une commande à priori parce que nous avons décrit notre système linéarisé en fonction des écarts \tilde{x} et \tilde{u} . Pour obtenir l'entrée u pour le système on doit augmenter \tilde{u} avec \bar{u} . Le système physique est simulé avec le système décrit dans chapitre 4.

7.1.3 Pôles

Pour que le système soit BIBO (Bounded Input Bounded Output) stable, les pôles doivent être négatifs. Les pôles font partie des solutions des équations différentiels $\dot{\tilde{x}} = (A - BK)\tilde{x}$, dans la forme $he^{\lambda t}$ avec λ égale aux pôles et h une constante. Quand les pôles sont négatives, les écarts \tilde{x} vont vers zéro autrefois ils augmente et le système devient instable. Plusieurs groupes de pôles sont essayé.

$P1$	=	$[-1$	-1.1	-1.2	-1.3	-1.4	-1.5	-1.6	-1.7	-1.8	$-1.9]$
$P2$	=	$[-2$	-2.025	-2.05	-2.075	-2.1	-2.125	-2.150	-2.175	-2.2	$-2.225]$
$P3$	=	$[-1$	-2	-3	-4	-5	-6	-7	-8	-9	$-10]$
$P4$	=	$[-2$	-2.05	-2.15	-2.2	-2.25	-2.3	-2.35	-2.4	-2.45	$-2.5]$
$P5$	=	$[-2$	-2.1	-2.2	-2.3	-2.4	-2.5	-2.6	-2.7	-2.8	$-2.9]$
$P6$	=	$[-2.94$	-2.95	-2.96	-2.97	-2.98	-2.99	-3	-3.01	-3.02	$-3.03]$
$P7$	=	$[-1$	-2	-4	-6	-8	-10	-12	-14	-16	$-18]$

7.1.4 Points de simulation

Le système est simulé en plusieurs points dans le volume de travail. Ces points sont caractérisé par l'état x_{ref} , ils sont des points où on veut aller. (figure 7.1.4)

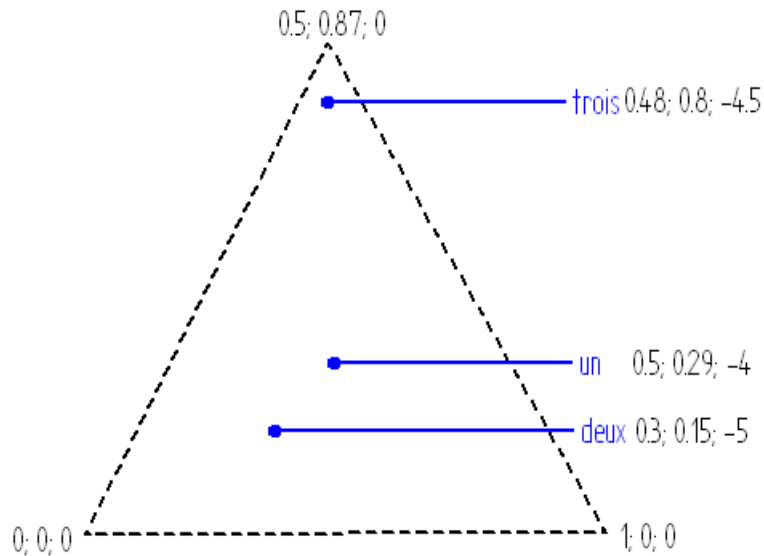


FIG. 5 – Points de simulation

Le premier point est le point autour du quel on a linéarisé, le deuxième point se ne trouve pas si loin de ce point et le troisième se trouve proche d'un pylône.

7.1.5 Code des couleurs

Plusieurs graphiques sont incluses dans les chapitres suivantes. Ces graphiques sont construits en utilisant un code des couleurs, sinon mentionné différemment.

Le vert est toujours utilisé pour ce qu'on veut, le bleu pour la simulation avec le contrôleur linéaire et le magenta pour la simulation avec le contrôleur non-linéaire [1].

Il y a des extra règles pour les graphiques de la grue en 3D :

Les triangles bleus sont les poulies, des x sont des anneaux et des cercles sont des charges.

Les lignes en vert montrent la grue dans la position qu'on veut qu'elle se trouve après stabilisation. Les lignes en rouge montrent la grue dans la position après la simulation.

La ligne en bleu est la trajectoire de la charge depuis sa position initiale vers sa position atteinte après le temps de simulation avec le contrôleur linéaire. La ligne en magenta est la trajectoire de la charge depuis sa position initiale vers sa position atteinte après le temps de simulation avec le contrôleur non-linéaire. La position initiale est là où la ligne bleu et magenta se coupent.

La dimension des forces est Newton, du temps seconde. Ces dimensions sont valables pour tous les graphiques suivants.

7.1.6 Première simulation

La démarche des simulations est la suivante :

- Définir des pôles
- Choisir un des trois points mentionnés ci-dessus
- Définir les perturbations

Dans une première phase, on a étudié le système en utilisant un câble long entre l'anneau et la charge. Sa longueur est trois fois la distance entre les pylônes.

Les résultats sont résumés dans la table ci-dessous (table 1). (Les noms réfèrent vers les fichiers sur le cd-rom [4], les graphiques ne sont pas compris dans le rapport afin de garder la lisibilité.)

pôles	point	nom	perturbation	stabilité	forces	statisme
P1	point1	vingtsept	/	stable	/	/
P1	point2	vingtsix	/	instable	piques et négatives	/
P1	point3	quatre	/	instable	piques et négatives	/
P2	point1	cinq	position	stable	/	/
P2	point2	vingthuit	/	stable	piques et négatives	statisme
P2	point3	six	/	instable	piques et négatives	/
P3	point1	unstandard	/	stable	/	/
P3	point1	unperturbation1	position	stable	piques	/
P3	point2	deuxstandard	/	stable	/	statisme
P3	point2	deuxperturbation1	position	stable	piques et négatives	statisme
P3	point3	troisstandard	position	stable	piques et négatives	statisme
P3	point3	troisperturbation1	position	stable	piques et négatives	statisme
P4	point1	vingt	vitesse	stable	/	/
P4	point2	huit	/	stable	négatives	statisme
P4	point2	vingtdeux	position+vitesse	stable	piques et négatives	statisme
P4	point2	vingtcinq	position+vitesse	stable	piques et négatives	statisme
P4	point3	vingtquatre	position	stable	piques et négatives	statisme
P4	point3	sept	position+vitesse	stable	piques et négatives	statisme
P5	point1	douze	/	stable	/	/
P5	point1	treize	position	stable	piques et négatives	/
P5	point1	dixhuit	position+vitesse	stable	piques et négatives	/
P5	point1	dixneuf	position+vitesse	stable	piques et négatives	/
P5	point2	quatorze	/	stable	/	statisme
P5	point2	quinze	position	stable	/	statisme
P5	point2	vingtetun	position+vitesse	stable	piques et négatives	statisme
P5	point3	seize	/	stable	/	statisme
P5	point3	dixsept	position	stable		statisme
P5	point3	vingttrois	position+vitesse	stable	piques et négatives	statisme

TAB. 1 – premier simulation

Les vitesses mentionné dans la table 1 sont des conditions initiales non-zéro pour les dérivés des coordonnées. Il s'agit des 'coups de vent' en réalité.

On voit directement qu'il y a un statisme dans tous les simulations dans un autre point que le point autour du quel on a linéarisé. On peut expliquer ce statisme par la manque d'un intégrateur et les directions des forces, qui sont juste seulement dans le point autour on a linéarisé. Cette faute existe parce que la commande à priori utilise cette position pour calculer \bar{u} . Cet écart provoque des forces appliqué au système même dans le cas que la grue se trouve déjà dans son but. Figure 6 montre le statisme de x_1 de la simulation 'quinze'. Les lignes rouges marquent la position désiré, la valeur simulé est en blue.

Un autre problème général est la présence des forces négatives. Celles-ci sont impossibles

parce que des forces négatives veulent dire qu'on pousse aux câbles. Le système souffre aussi des piques de forces qui peut provoquer des fractures des câbles où le chauffage des moteurs. Figure 7 montre en bleu les forces appliqués au système, on constate qu'il y a des piques de force au début et que la force T_1 reste négative. Aussi T_2 est négative au début. La ligne en rouge marque une force de zéro, tout au-dessous cette ligne est négative.

La simulation montre qu'on a un contrôleur lente avec les pôles P1 et P2. Le système va lentement vers sa position stable. Quand la perturbation est trop grande, le système devient instable parce que la réaction est trop lente.

Avec la troisième groupe de pôles, P3, on a un système stable mais il applique des forces trop grands aux câbles. Cette contrôleur est trop vite car les pôles sont grands. Les pôles font partie des solutions des équations différentiels $\ddot{x} = (A - BK)\tilde{x}$, dans la forme $e^{\lambda t}$ avec λ égale aux pôles. On voit que des grands pôles provoquent des écarts qui vont vite vers zéro et des petits pôles qui vont lent vers zéro.

En utilisant les pôles P4 et P5, on a des meilleurs résultats mais aussi ses pôles ont les problèmes générales. Figure 8 montre la simulation 'quinze' après 10s. On peut constaté que la grue avec le contrôleur linéaire se stabilise dans un point différament de la position demandé : il y a du statisme.

7.1.7 Influence de la longueur du câble

La longueur du câble l détermine la fréquence de résonance ω du câble, comme dans l'expression 22 avec g la constante gravitationnelle.

$$\omega = \sqrt{\frac{g}{l}} \quad (22)$$

Quand l est plus longue, la fréquence diminue et la câble devient facile à contrôler. En réalité est la longueur du câble est limité. Dans ce cas-ci on choisit une longueur l égale à trente pourcent de la distance entre les pylônes. Cette conclusion exige une nouvelle simulation pour le système.

7.1.8 Seconde simulation

pôles	point	nom	perturbation	stabilité	forces
P1	point1	p1pt1	perturbation	instable	piques et négatives
P3	point1	p3pt1	perturbation	instable	piques et négatives
P6	point1	p8pt1	perturbation	instable	piques et négatives
P7	point1	p1enormept1	perturbation	instable	piques et négatives

TAB. 2 – seconde simulation

Les résultats des simulations sont placé dans la table 2. Tous les simulations sont simulé dans le même point, point 1, redéfinié car la longueur du câble est redéfinié. Le nouveau point 1 se trouve à (0.5;0.29;-0.4). Ils ont aussi la même perturbation :

$$\Delta x_1 = 0.20$$

$$\Delta x_2 = 0.21$$

$$\Delta x_3 = 0.10$$

Le optimum se trouve dans les pôles P3, mais il n'y a pas des cas stables. Une simulation de cet optimum pour dix secondes est montré dans figure 9.

7.2 Régulateur proportionnel-intégral-différentiel PID

Le contrôleur PD étudié dans le chapitre précédent ne donne pas des résultats satisfaisants. L'introduction des intégrateurs sera étudiée ici. Ils peuvent être une solution pour le statisme observé dans la première simulation de ci-dessus et le surplus de commande nécessaire à l'annulation de l'effet de la perturbation.

7.2.1 Théorie

Les intégrateurs vont intégrer les écarts des états \tilde{x} . En introduisant la variable $\dot{s} = \tilde{x}$, on peut exprimer cette intégration comme $s = \int \dot{s}$. L'écart de forces \tilde{u} qu'on va sommer avec la commande à priori \bar{u} , dépend maintenant linéairement de l'écart de l'état \tilde{x} et l'intégral s . En plaçant les deux termes ensemble on obtient $-K^*$. (équation 23)

$$\tilde{u} = -K\tilde{x} - K_i s = K^* \tilde{x}^* \quad (23)$$

On utilise trois intégrateurs pour le système étudié : sur x_1, x_2, x_3 .

$$\dot{s}_1 = \tilde{x}_1$$

$$\dot{s}_2 = \tilde{x}_2$$

$\dot{s}_3 = \tilde{x}_3$ Les deux équations considérées (24 et 25) donnent ensemble l'équation 26 qui est égale à l'équation 27.

$$\dot{s} = x - x_{ref} = \tilde{x} \quad (24)$$

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + B\tilde{u}(t) \quad (25)$$

$$\dot{\tilde{x}}^*(t) = A^*\tilde{x}^*(t) + B^*\tilde{u}(t) \quad (26)$$

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{s}_{x1} \\ \dot{s}_{x2} \\ \dot{s}_{x3} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ s_{x1} \\ s_{x2} \\ s_{x3} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \tilde{u} \quad (27)$$

$-K^*$ est obtenu par placement de pôles avec la même commande en Matlab qu'avant : $K^* = \text{place}(A^*, B^*, \text{pôles})$. Il y a 13 pôles maintenant, trois pôles en plus que dans le chapitre précédent.

7.2.2 Implementation en Matlab

Le système de chapitre 7.1.2 décrit par figure 4 est adapté à la nouvelle situation montrée en figure 10.

7.2.3 Pôles

Pour que le système soit BIBO (Bounded Input Bounded Output) stable, les pôles doivent être négatifs. Les pôles font partie des solutions des équations différentielles $\ddot{x}^* = (A^* - B^*K^*)\ddot{x}^*$, dans la forme $he^{\lambda t}$ avec λ égale aux pôles et h une constante. Quand les pôles sont négatives, les écarts \tilde{x} vont vers zéro autrefois ils augmente et le système devient instable. Plusieurs groupes de pôles sont essayé, ils ne sont pas tous affiché ici afin de garder la lisibilité.(annexe 3) Les noms sont les abbréviations de leurs grandeurs relatives :P=petit, G=grand, E=enorme. Le premier caractère réfère aux premiers dix pôles, ce sont les pôles pour le contrôleur PD et le deuxième caractère aux derniers trois pôles pour les intégrateurs.

$PP =$	[−0.01	−0.02	−0.03	−0.04	−0.05	−0.06	
	−0.07	−0.08	−0.09	−0.1	−0.11	−0.12	−0.13]
$GP =$	[−10	−11	−12	−13	−14	−15	
	−16	−17	−18	−19	−3	−3.1	−3.2]
$PG =$	[−2.94	−2.95	−2.96	−2.97	−2.98	−2.99	
	−3	−3.01	−3.02	−3.03	−10	−20	−30]
$GG =$	[−1	−2	−3	−4	−5	−6	
	−7	−8	−9	−10	−10	−15	−20]
$EG =$	[−10	−11	−12	−13	−14	−15	
	−16	−17	−18	−19	−20	−25	−30]

7.2.4 Points de simulation

Le système est simulé en plusieurs points dans le volume de travail. Ces points sont caractérisé par l'état x_{ref} , ils sont des points où on veut aller. (7.2.4)

Le point numéro un est le point autour on a linéarisé, le deuxième point ne se trouve pas si loin de ce point et le troisième se trouve proche d'un pylône.

7.2.5 Premier simulation

La démarche des simulations est la suivante :

- Définir des pôles
- Simuler dans le point 1 avec une perturbation
- Choisir les meilleurs pôles
- Simuler dans le point 2 avec une perturbation
- Choisir les meilleurs pôles
- Simuler dans le point 2 avec une perturbation et des coups de vent
- Simuler dans le point 3 avec une perturbation et des coups de vent

Les résultats sont sommé dans la table ci-dessous.(table 7.2.5) (Les noms réfèrent vers les fichiers sur le cd-rom [4], toutes les graphiques ne sont pas compris dans le rapport afin de garder la lisibilité.)

pôles	point	nom	perturbation	stabilité	forces	statisme
PP	point1	pool16pt1b	position	instable	piques et négatives	/
GP	point1	pool15pt1b	position	instable	piques et négatives	/
PG	point1	pool8pt1b	position	stable	/	/
GG	point1	pool13pt1b	position	stable	peu négatives	/
EE	point1	pool14pt1b	position	instable	piques et négatives	/

TAB. 3 – premier simulation

La table montre que des pôles relativement grands pour les intégrateurs donnent des situations stables, comme avec PG et GG. Les conclusions pour les pôles du régulateur PD sont moins claires pour le moment. Quand tous les pôles sont augmentés plus comme EE, le système devient instable. Les figures 12 et 13 montrent les cas stables. On peut remarquer que le système trouve sa position stable plus vite avec les pôles PG qu'avec les pôles GG.

7.2.6 Deuxième simulation

pôles	point	nom	perturbation	stabilité	forces	statisme
PG	point2	pool8pt2a	/	instable	/	/
GG	point2	pool13pt2a	/	stable	/	/

TAB. 4 – deuxième simulation

Une deuxième simulation est effectuée, en ce cas on simule autour du deuxième point de la figure 7.2.4. Avec les pôles PG on obtient un système instable, mais avec les pôles GG un système stable. (figures 14 et 15) Les pôles GG sont l'optimum.

7.2.7 Troisième simulation

La troisième simulation donne des résultats tous stables, dans les trois points et avec des perturbations. Mais dans tous les cas, la grue bouge plus pour obtenir sa position d'équilibre que la grue avec le contrôle non-linéaire [1] comme on peut voir en figures 16 et 17. Le contrôle non-linéaire va aussi plus vite vers son but (figure 18).

Les forces sont presque toujours positives, comme il faut. Dans certaines cas elles deviennent négatives, plutôt au début pour une fraction d'une seconde. Les forces les plus négatives sont enregistrées dans la simulation avec le nom 'pool13pt2b', ses forces sont montrées dans la figure 19.

Il n'y a pas de grands pics de forces, seulement quand il y a des forces négatives changent les forces très vite. Les plus hauts pics sont observés dans la simulation avec le nom 'pool13pt1b', ses forces sont montrées dans la figure 20.

Dans ce cas les forces atteignent 10N pour une charge de 0.3kg, on doit contrôler que les moteurs peuvent tirer si fort et changer si vite sans chauffer trop et sans casser le câble. Les

pôles	point	nom	perturbation	stabilité	forces	statisme
GG	point1	pool13pt1sanspert	/	stable	/	/
GG	point1	pool13pt1b	position	stable	/	/
GG	point1	pool13pt1d	position+vitesse	stable	/	/
GG	point2	pool13pt2sanspert	/	stable	/	/
GG	point2	pool13pt2b	position	stable	/	/
GG	point2	pool13pt2d	position+vitesse	stable	/	/
GG	point3	pool13pt3sanspert	/	stable	/	/
GG	point3	pool13pt3b	position	stable	très peu négative	/
GG	point3	pool13pt3d	position+vitesse	stable	très peu négative	/

TAB. 5 – troisieme simulation

forces changent en général aussi plus vite avec le contrôleur lineaire qu'avec le contrôleur non-lineaire.

Tous les considérations ci-dessus montrent que la loi de contrôle non-lineaire a une meilleur performance que la loi de contrôle lineaire.

8 Les diagrammes de Bode

Les diagrammes de Bode caractérisent la fonction de transfert $G(j\omega)$. Par fonction de transfert on a deux diagrammes, une graphique d'atténuation et une graphique de phase. La graphique d'atténuation porte en ordonnée le module de $G(j\omega)$ en décibels, autrement dit $20\log |G(j\omega)|$, avec une échelle linéaire. La graphique de phase, fait apparaître en ordonnée l'argument $\arg G(j\omega)$ en degrés avec une échelle linéaire. L'abscisse pour les deux diagrammes est la pulsation ω avec une échelle logarithmique. [3]

Quand un système est linéaire, les diagrammes de Bode doivent être indépendant du point autour du quel on linéarise. Car la linéarisation d'un système linéaire donne le même système, quelconque le point autour lequel on linéarise.

Dans ce chapitre, on va étudier si la grue est linéaire. Afin de trouver les diagrammes de Bode, on a besoin des fonctions de transfert du système.

8.1 Les fonctions de transfert

Tous les équations décrivaient dans les chapitres précédents sont des fonctions dépendantes du temps. Afin de trouver les fonctions de transfert du système, on doit transformer ces fonctions avec la transformation de Laplace. Dans le domaine de Laplace on peut exprimer le système comme $U(s) = G(s)Y(s)$. Avec $U(s)$ la transformation de Laplace de l'entrée, $u(t)$. $u(t)$ sont les forces $T_1(t), T_2(t), T_3(t)$. $Y(s)$ est la transformation de Laplace de la sortie, $y(t)$. Il s'agit principalement de la position de la charge $x_1(t), x_2(t), x_3(t)$. L'expression dans le domaine temporelle de la sortie devient $y(t) = Cx(t) + Du(t)$ avec C une matrice avec des zéros. Seulement sur les positions (1,1), (2,3) et (3,5) on a 'un'. La matrice C est égale à zéro.

On trouve les fonctions de transfert avec la commande suivante en Matlab : `sys=ss(A,B,C,D)`. Où matrice A et B sont les matrices de la linéarisation étudié dans les chapitres précédentes. On trouve 3x3 paires de diagrammes de Bode, parce qu'on a pour chaque combinaison d'une entrée et une sortie une fonction de transfert. Les résultats sont mis en annexe.

8.2 Diagrammes de Bode

Avec la commande `TfSys1=minreal(tf(sys1),0.8)` en Matlab on peut dessiner les diagrammes de Bode. Quand on utilise cette commande pour deux systèmes linéarisés autour deux points différents on trouve les figures 21. Le paramètre 0.8 spécifie la tolérance utilisé pour l'élimination des états uncontrollables ou inobservables ou de l'annulation pôle-zéro. En bleu on a le système linéarisé autour le point 1 des simulations, (0.5;0.29;-0.4) et en rouge autour point 2 des simulations, (0.3;0.15;-0.5).

Les figures 21 nous montre que le système a des diagrammes de Bode différent dans les différents points autour on linéarise. Les atténuations et les phases ont des différentes largeurs. Aussi les fréquences de résonance et les changements de phase se trouvent sur des différents points. Si on place les pôles pour diminuer l'effet d'une fréquence de résonance

dans un point, ils ne diminueront pas les effets de résonance dans des autre points. Car les fréquences de résonance sont différentes.

Etudions une paire des diagrammes de Bode.

Figure 22 montre les diagrammes de Bode pour le fonction de transfert pour l'entrée T_2 vers la sortie x_3 .

Une première remarque est la présence de deux fréquences de résonance pour le système linéarisé autour point 1 et seulement un pour le système linéarisé autour point 2. Une autre remarque est que le fréquences de résonance en bleu ne correspond pas avec un des deux piques en rouge. Les atténuations sont au débuts les mêmes mais après 1 rad/sec ils diffèrent. Les phases sont seulement dans une petite bande de fréquence les mêmes.

9 Conclusion

- On a construit un modèle dynamique en utilisant les formules de Lagrange, exposé dans chapitre 3.
- On a fait la synthèse d'une loi de controle linéarisée avec rétroaction en utilisant la technique de placement des valeurs propres.
- D'abord un régulateur PD a été testé, ce qui ne donnait pas des résultats satisfaisant pour des cas avec des dimensions réels. Ce régulateur est instable et souffre du statisme, des piques des forces et des forces négatives.
- Puis un régulateur PID est élaboré, il donne des cas stables dans tous les points simulé et avec tous les perturbations.
- La comparaison avec la loi de controle non-linéaire nous montre que la charge bouge plus avec le contrôleur lineaire et que la stabilité avec le contrôleur lineaire est plus sensible. Un optimum pour les pôles pour un point ne donne pas l'optimum quand la grue se bouge et se stabilise dans un autre point.
- Les diagrammes de Bode nous montre que le système est fortement non-lineaire.

Lausanne, le 12 février 2007

Dominick Vanthienen

Références

- [1] Davide BUCCIERI. 'commande et aide au pilotage d'une grue 3d ne comportant aucune pièce mobile'. Master's thesis, Laboratoire d'Automatique LA-IIS-STI-EPFL, Lausanne, 2003.
- [2] Dr Denis GILLET. *Systèmes multivariables I*. photocopié EPFL, Lausanne, 2001.
- [3] Roland LONGCHAMP. *Commande numérique de systèmes dynamiques*, pages 708–709. Presses polytechniques et Universitaires Romandes, Lausanne, 2006.
- [4] Dominick VANTHIENEN. cd-rom 'a linear state feedback controller for a high efficiency crane', 2007.

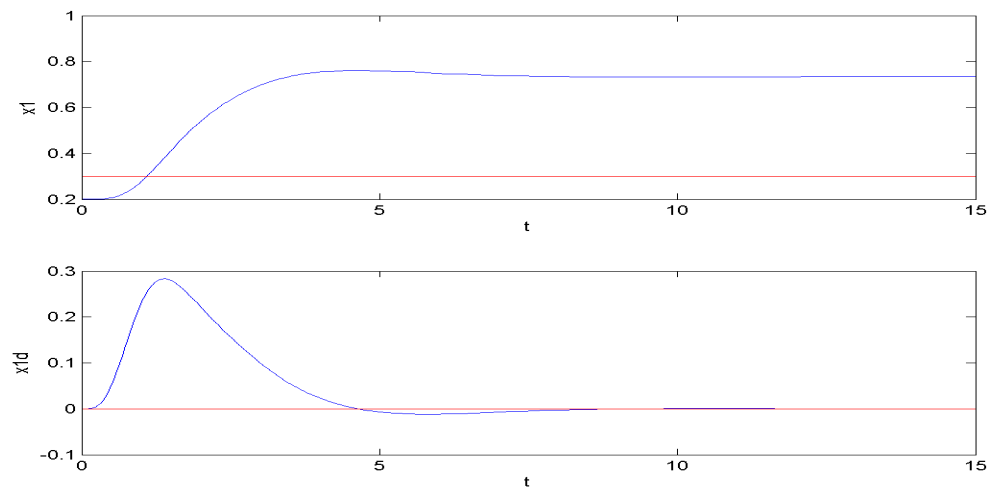


FIG. 6 – Statisme de x_1 de la simulation 'quinze'

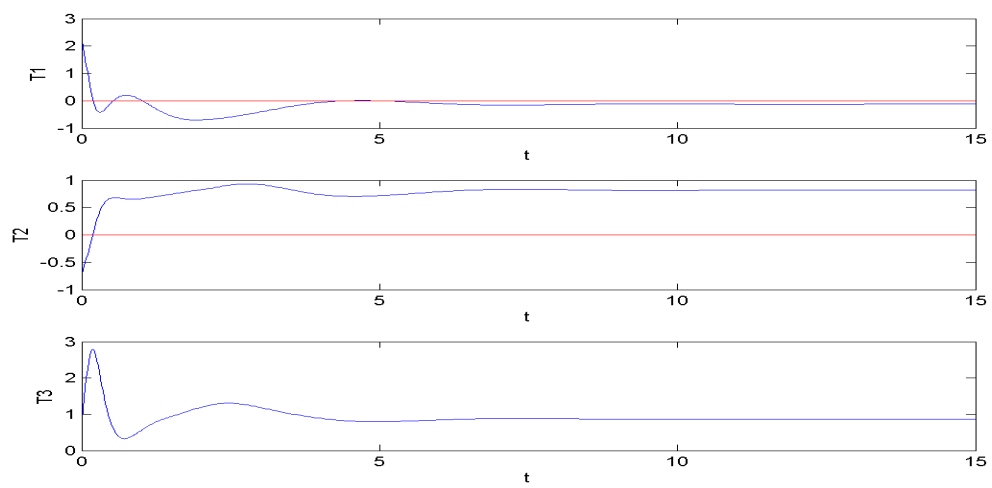


FIG. 7 – Piques de forces et forces négatives de la simulation 'vingtcinq'

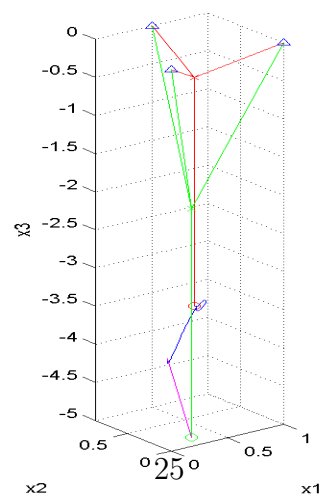


FIG. 8 – Simulation 'quinze' après 10s

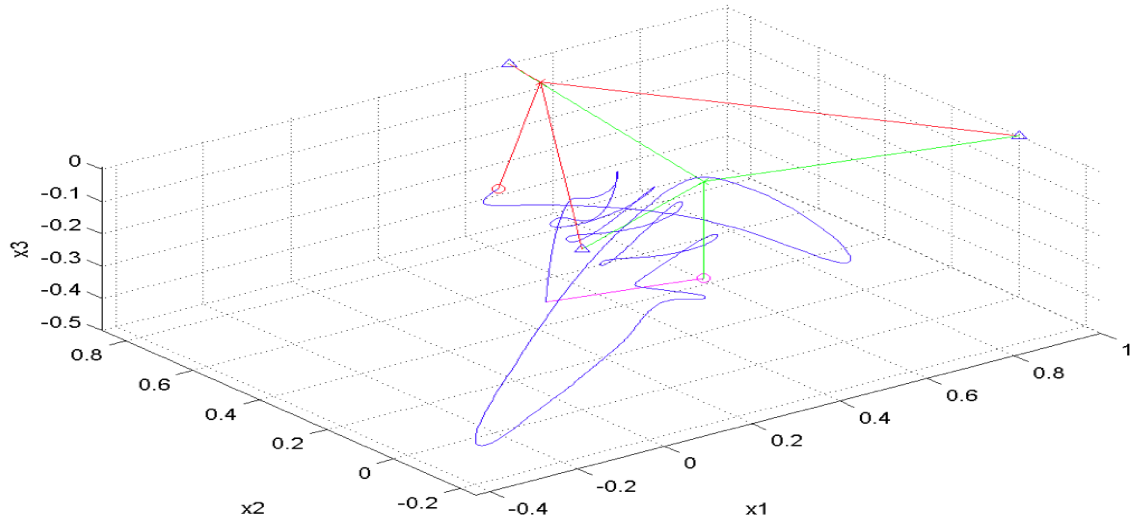


FIG. 9 – Simulation 'p3pt1b' après 10s

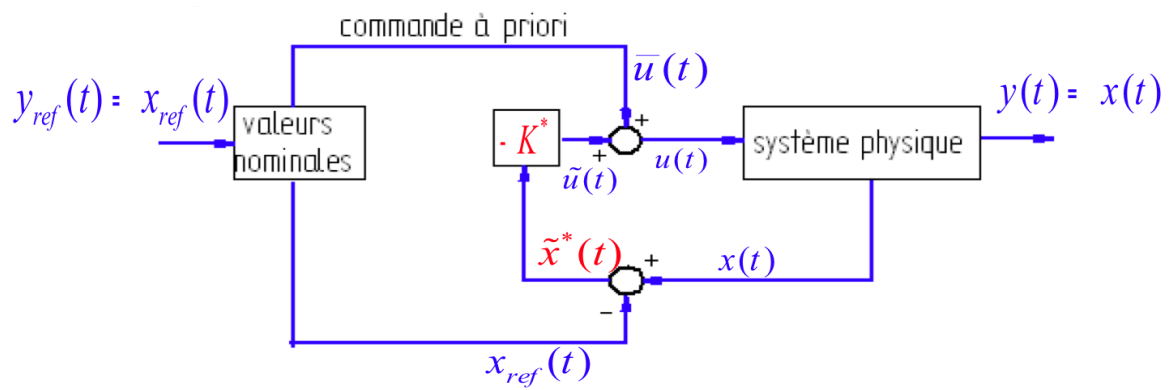


FIG. 10 – Commande d'état

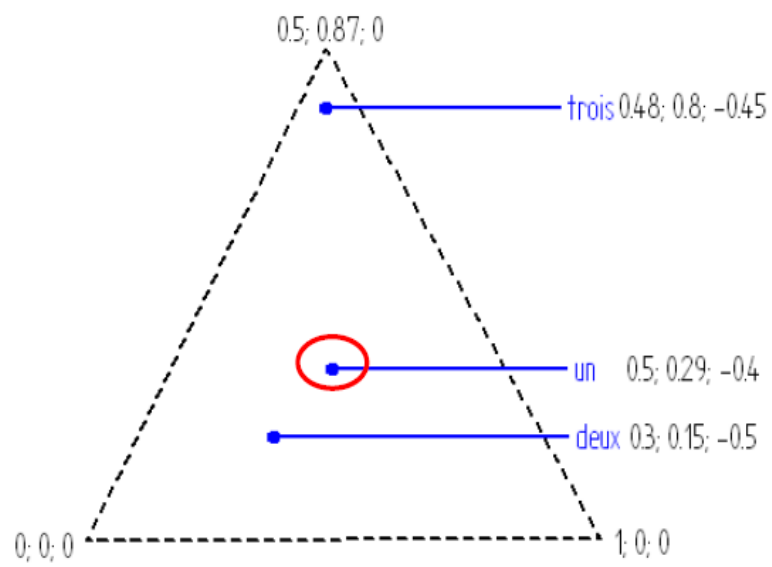


FIG. 11 – Points de simulation

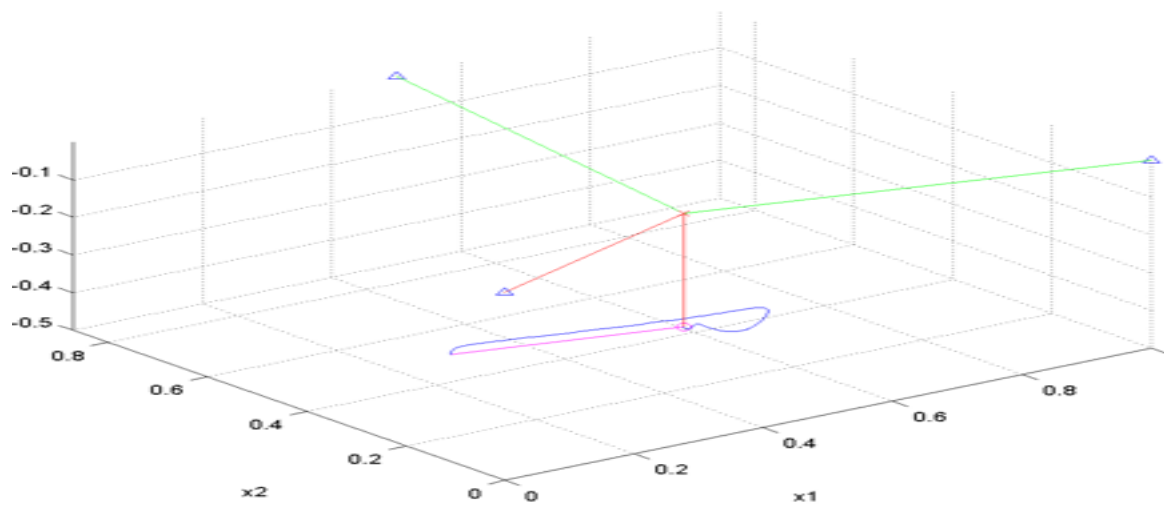


FIG. 12 – Simulation 'pool8pt1b' après 10s

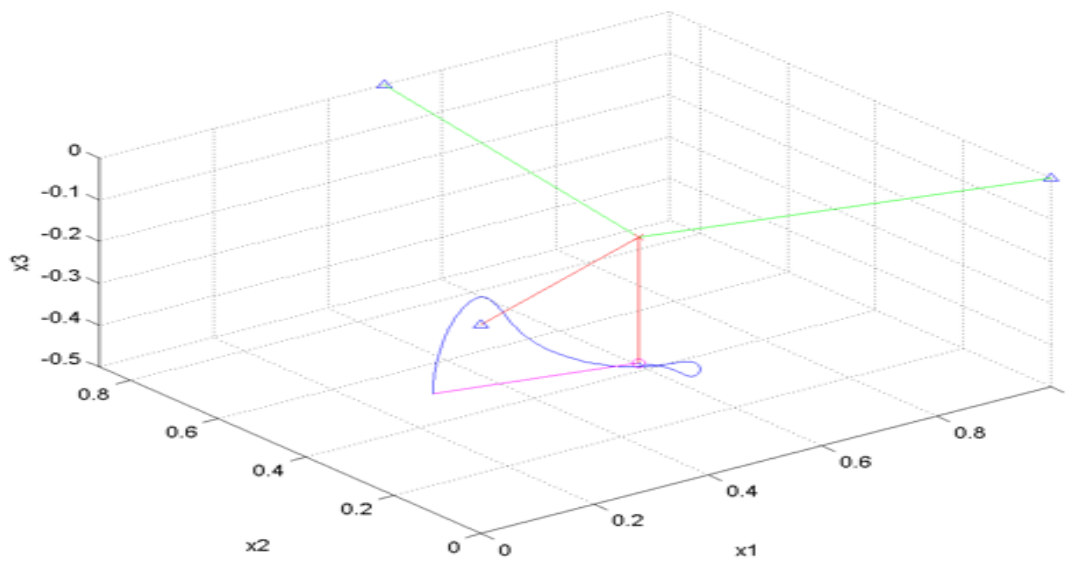


FIG. 13 – Simulation 'pool13pt1b' après 10s

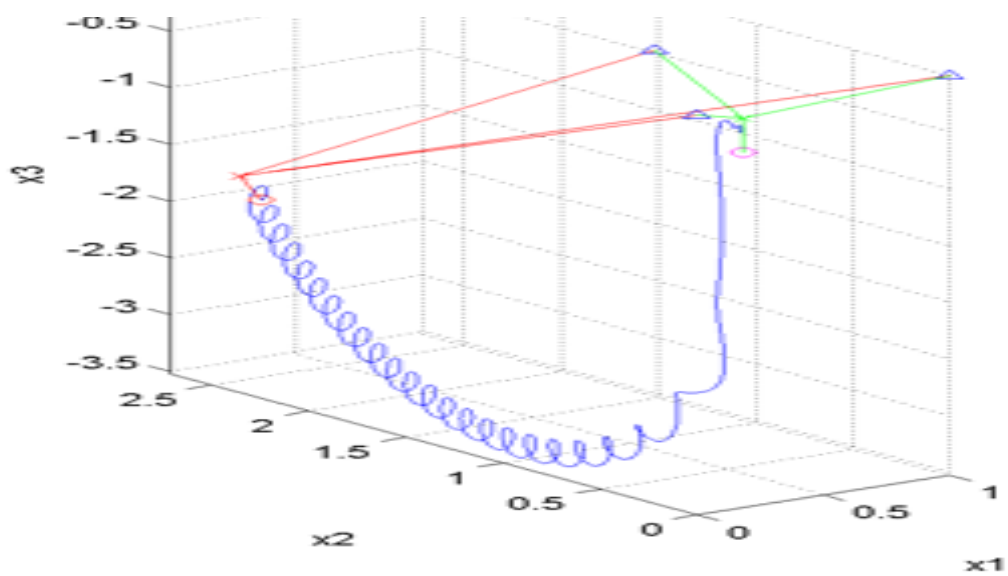


FIG. 14 – Simulation 'pool8pt2a' après 10s

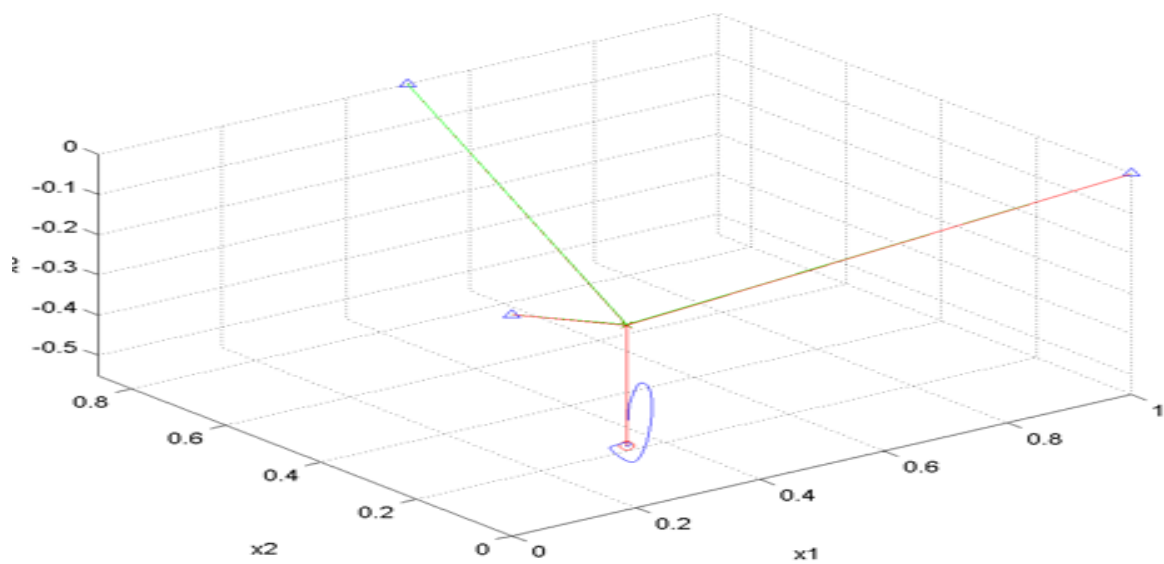


FIG. 15 – Simulation 'pool13pt2a' après 10s

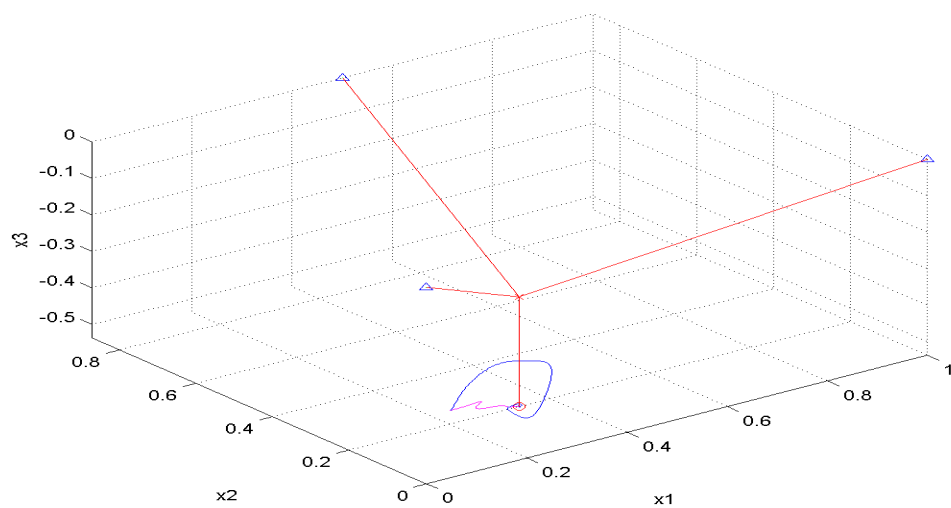


FIG. 16 – Simulation 'pool13pt2d' après 10s

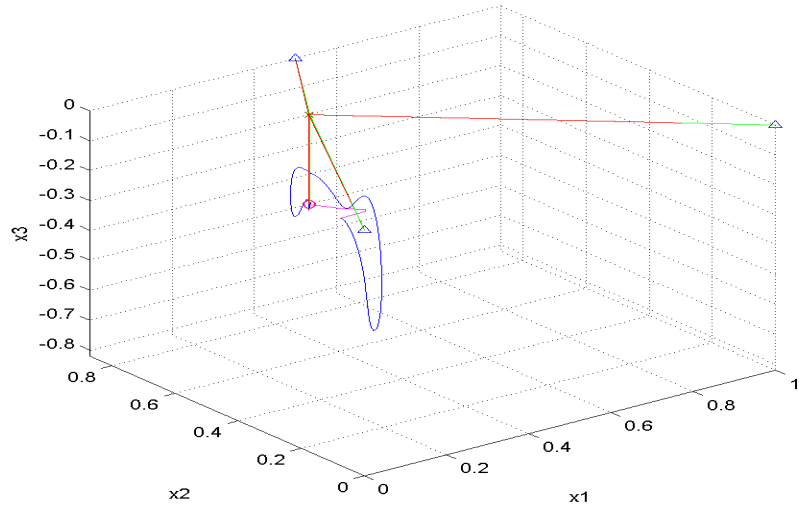


FIG. 17 – Simulation 'pool13pt3d' après 10s

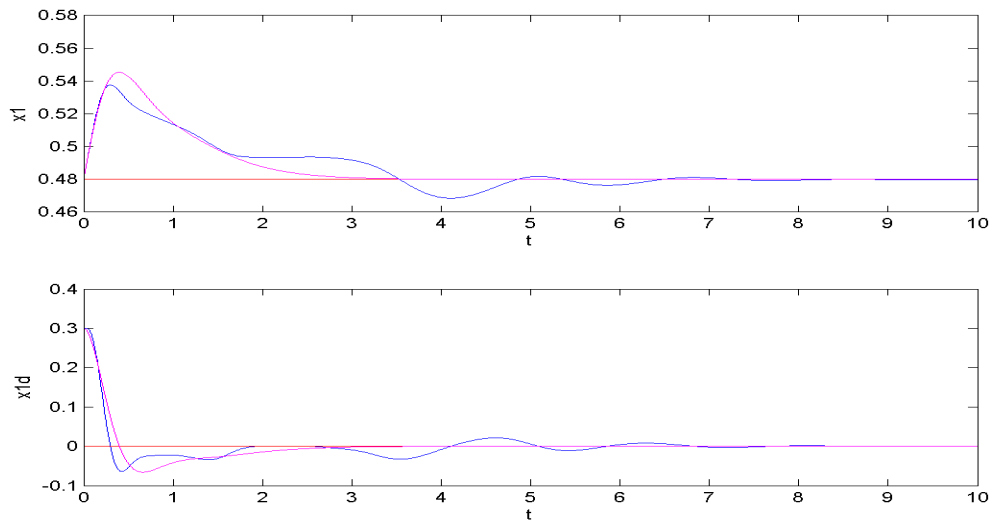


FIG. 18 – Simulation de x_1 de 'p13pt3d' après 10s

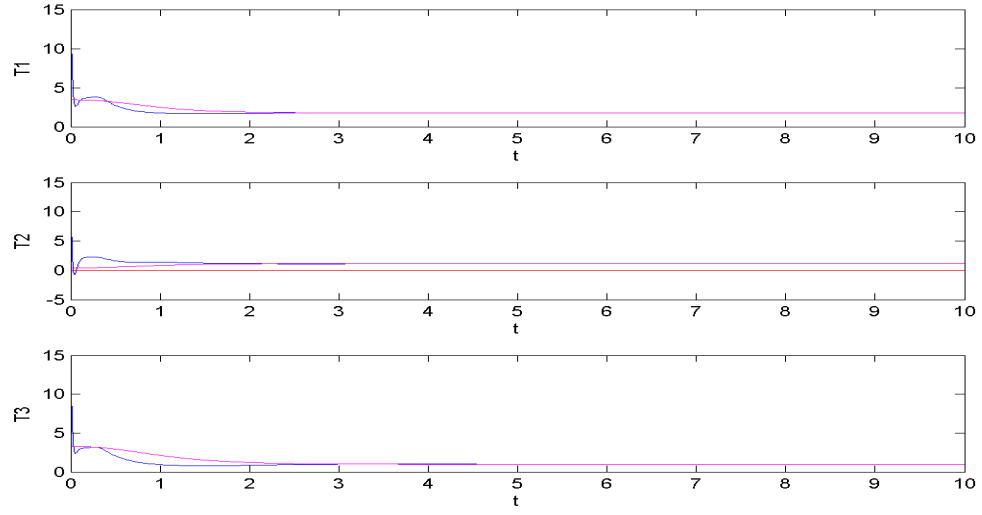


FIG. 19 – Simulation des forces de 'p13pt2b' après 10s

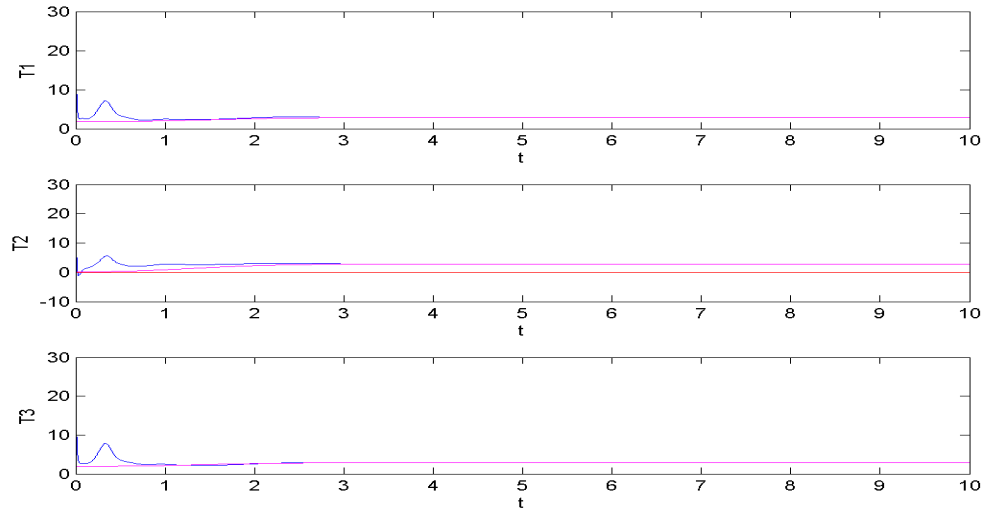


FIG. 20 – Simulation de x_1 de 'p13pt1b' après 10s

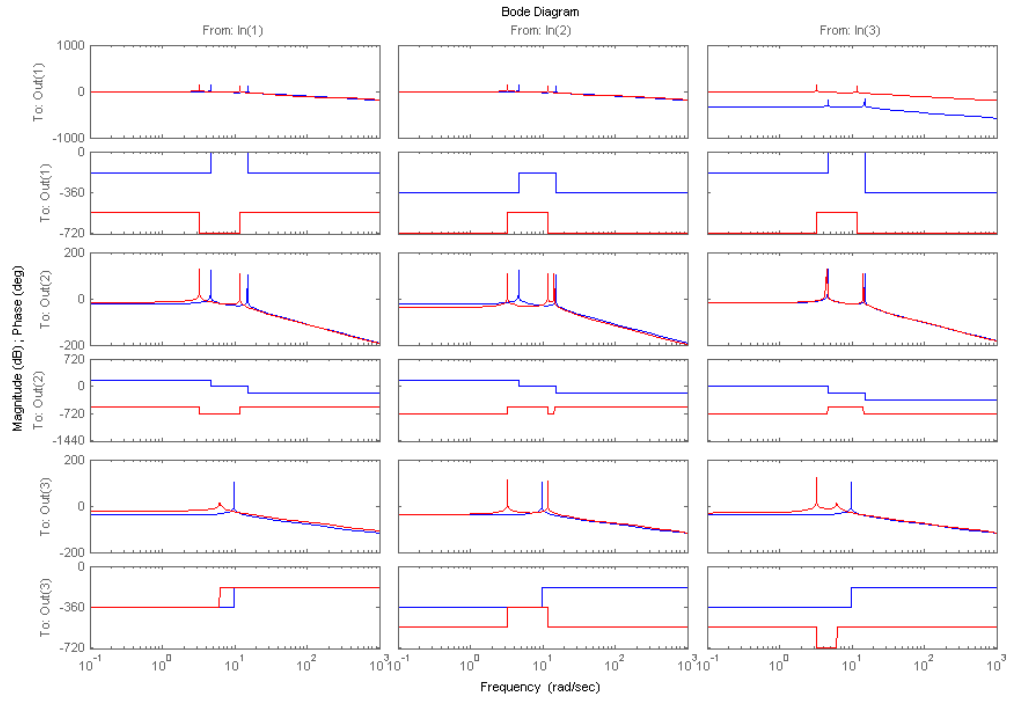


FIG. 21 – Diagrammes de Bode

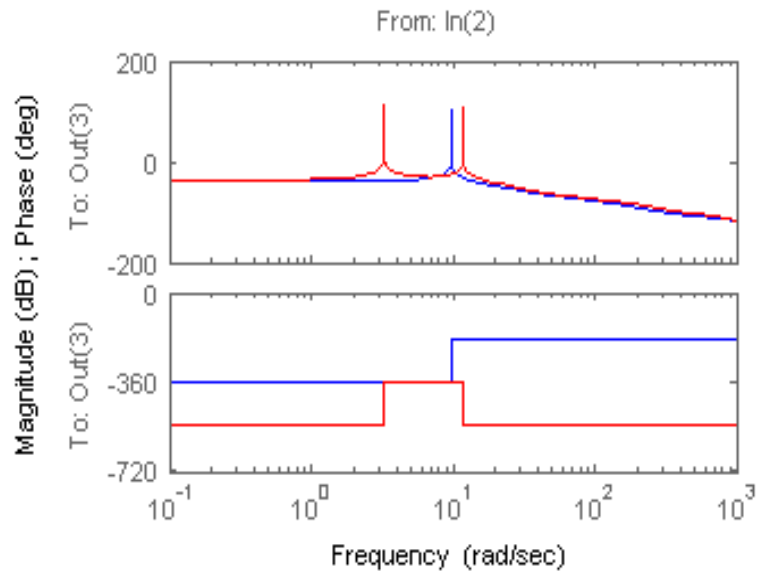


FIG. 22 – Diagramme de Bode $T_2 \rightarrow x_3$