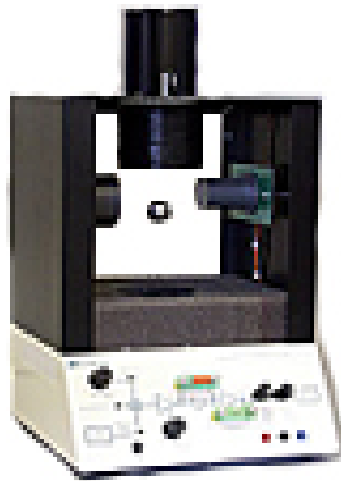


Micro-Projet d'Automatique

# BILLE EN SUSTENTATION MAGNÉTIQUE



LIMA Gustavo

PRUVOST Côme

TREMBLIN Marina

# Bille en sustentation magnétique

## Table des matières

---

1. Présentation du système et du projet.....	4
1.1. Présentation du système étudié.....	4
Équation électrique.....	4
Équation mécanique.....	5
1.2 Déroulement du projet.....	5
2. La Séquence Binaire Pseudo-Aléatoire.....	5
3. Acquisition du signal.....	7
4. Relevé des valeurs de point de fonctionnement.....	8
5. Identification du correcteur.....	9
5.1. La boîte à outils ident.....	9
5.2. Identification des expérimentations.....	11
6. Implémentation du correcteur numérique sur le système réel.....	12
6.1 Implémentation du correcteur à l'aide de simulink.....	12
6.2. Validation du correcteur sur le système réel.....	13
7. Linéarisation du système physique.....	13
8. Correction par PID et essais en réel.....	15

## Index des figures

---

Figure 1: Le système physique bille-aimant.....	4
Figure 2: Séquence Binaire Pseudo-Aléatoire réalisée avec simulink.....	6
Figure 3: La SBPA encapsulée avec son horloge interne.....	7
Figure 4: Schéma simulink utilisé pour l'acquisition de données via xPC Target.....	8
Figure 5: Schéma simulink pour l'acquisition de $z_0$ et $u_0$ .....	9
Figure 6: La boîte à outils ident.....	10
Figure 7: Dialogue de choix du type d'identification dans ident.....	10
Figure 8: Les options du préprocesseur.....	10
Figure 9: Comparaisons des modèles identifiés avec les expérimentations.....	11
Figure 10: Agrandissement d'un point critique de l'identification.....	12
Figure 11: Schéma simulink d'implémentation du correcteur numérique.....	12
Figure 12: Lieu des racines et réponse indicielle lorsque les modes rapides sont actifs.....	16
Figure 13: Lieu des racines et réponse indicielle lorsque l'effet des pôles rapides est atténué.....	16

Ce micro-projet a pour but l'écriture de lois de commande numérique. Nous nous sommes pour cela aidé des outils matlab tel que xPC Target et ident. Le système étudié est une bille en sustentation magnétique. Nous avons essayer de remplacer le correcteur déjà implanté dans le système.

Nous allons dans ce rapport détailler les aspects modélisation, synthèse et simulation des correcteurs, résultats d'expérimentation et les analyses critiques des résultats expérimentaux.

# 1. Présentation du système et du projet

## 1.1. Présentation du système étudié

Dans ce bureau d'étude, nous tentons de corriger un système non-linéaire. Le système est une bille en sustentation magnétique. Il est donc composé d'une bille en acier maintenue en lévitation magnétique grâce au champ magnétique induit par une bobine. Le but de l'asservissement étant de maintenir la bille dans une position verticale donnée.

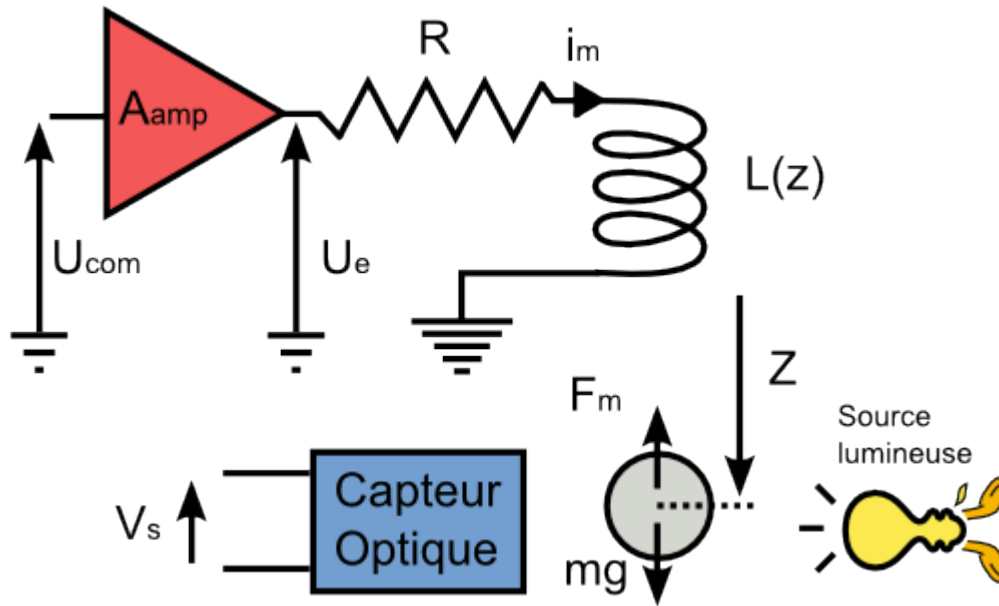


Figure 1: Le système physique bille-aimant

On envoie un courant dans la bobine afin de créer une force magnétique. Cette force est fonction du courant qui la traverse ainsi que de l'inductance de la bobine. Cette inductance dépend, quand à elle, de la position de la bille. L'équilibre mécanique est atteint quand la force exercée par la bobine s'annule avec la force de pesanteur de la bille. Un capteur optique détecte la position de la bille.

### Équation électrique

On applique une tension  $V_e$  aux bornes de la bobine d'inductance  $L(Z)$  et de résistance  $R$ .

$$U_e = A_{amp} U_{com} + B_{amp} = R i_m + \frac{d(L i_m)}{dt}$$

L'inductance de la bobine peut s'exprimer de différentes manières. La forme supposée est la suivante :

$$L(Z) = L_1 + \frac{L_0}{Z}$$

Cela entraîne pour l'équation électrique la forme suivante:

$$U_e = A_{amp} U_{com} + B_{amp} = R i_m + \left( L_1 + \frac{L_0}{Z} \right) \frac{d(i_m)}{dt} - \frac{L_0 i_m}{Z^2} \frac{dZ}{dt}$$

## Équation mécanique

---

La force développée par l'électro-aimant peut s'exprimer comme suit :

$$F_m(Z, i_m) = \frac{1}{2} i_m^2 \frac{\partial L}{\partial Z} = -\frac{1}{2} i_m^2 \frac{L_0}{Z^2}$$

On obtient donc d'après l'équation fondamentale de la dynamique la relation suivante :

$$m \ddot{Z} = mg - \frac{1}{2} i_m^2 \frac{L_0}{Z^2}$$

On pose comme variables d'état :

$$X_1 = Z, \quad X_2 = \dot{Z}, \quad X_3 = i_m$$

On obtient le modèle suivant :

$$\begin{cases} \dot{X}_1 = X_2 \\ \dot{X}_2 = g - \frac{X_3^2 L_0}{2m X_1^2} \\ \dot{X}_3 = \frac{L_0 X_2 X_3}{X_1 (L_0 + L_1 X_1)} - \frac{R X_1 X_3}{L_0 + L_1 X_1} + \frac{X_1}{L_0 + L_1 X_1} (A U_{com} + B) \end{cases}$$

On peut constater que ce modèle est non-linéaire.

### 1.2 Déroulement du projet

---

Un correcteur électronique est déjà implémenté sur le système. On cherche à le modéliser, le discrétiser et le remplacer.

Tout d'abord, on procède à des mesures sur le correcteur. On utilise pour cela un générateur de Signal Binaire Pseudo Aléatoire (SBPA). Cela permet d'étudier le correcteur sur une réponse indicielle mais aussi sur un terme plus long et sur des changements rapides.

On utilise ensuite la boîte à outils `ident` pour identifier le modèle à partir de plusieurs courbes dont l'entrée diffère, notamment pour éviter les problèmes de saturation.

Une fois le modèle identifié, on peut l'implémenter dans un schéma `simulink`. Cependant, il ne faut pas oublier que le correcteur a été modélisé autour d'un point de fonctionnement, il faut donc établir une liste de valeurs à utiliser comme point de fonctionnement. On peut enfin implémenter le correcteur numérique sur le système réel. On utilise pour cela `xPC Target`.

Enfin, on modélise le système bille-aimant pour créer un nouveau correcteur. On utilise un correcteur proportionnel intégral dérivé réglé à l'aide de `rltool`.

## 2. La Séquence Binaire Pseudo-Aléatoire

---

La Séquence Binaire Pseudo-Aléatoire est un signal rectangulaire, ou plutôt une succession d'états haut et d'états bas, dont la longueur varie de manière aléatoire.

La SBPA est générée à l'aide de registres à décalage bouclés auxquels on applique une fonction binaire. On en tire alors une suite de bits plus ou moins prévisible. En effet, sur une grande longueur, le signal reste périodique puisque les mêmes valeurs sont déplacées dans la boucle et

reviennent donc à leur position initiale au bout d'un certain temps. La longueur maximale d'une séquence est  $2^N - 1$  où  $N$  est le nombre de cellules du registre à décalage.

Un tel signal est intéressant pour l'analyse d'un système car il permet d'étudier des cas spécifiques que ne peut pas étudier un signal périodique. D'une manière générale, le signal étant rectangulaire, on observe une réponse indicielle. Mais le fait que la longueur des états varie permet d'obtenir en une seule série de données la réponse du système sur plusieurs fréquences. De plus, la valeur moyenne du signal est nulle ce qui permet de ne pas déplacer le point de fonctionnement du système. Facile à générer par rapport à d'autres signaux présentant les mêmes propriétés, c'est une méthode privilégiée pour l'identification.

On réalise une SBPA sous *simulink* en utilisant, comme on peut le voir dans la Figure 2, 9 blocs mémoire (*flip-flop*) bouclés, et l'on branche des portes ou exclusives en quatre points des registres. Ces portes forment la fonction crible qui permet d'obtenir sous la forme d'un booléen une séquence pseudo-aléatoire. Le bloc compare to zero est important pour transformer cette valeur booléenne en valeur d'état haut et d'état bas en tension. Les blocs gain et constant seront des paramètres du module SBPA et qui permettront de régler les valeurs en tension de l'état haut et de l'état bas. On note la présence du bloc trigger qui coordonne, à l'aide une horloge, le décalage des registres.

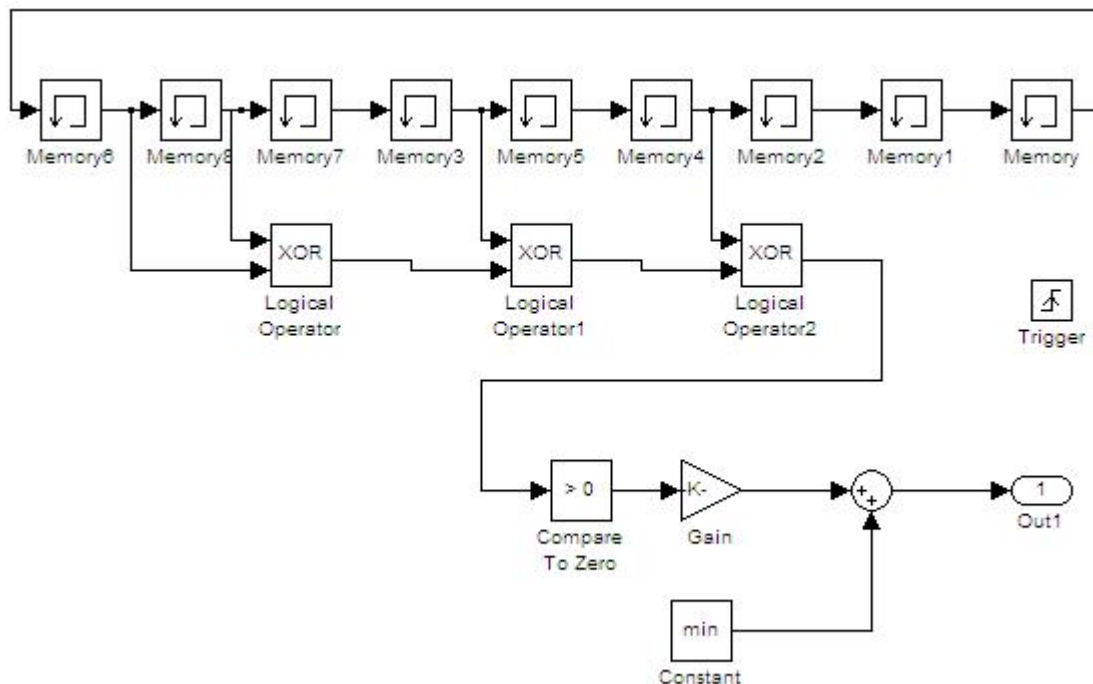


Figure 2: Séquence Binaire Pseudo-Aléatoire réalisée avec *simulink*

On encapsule l'ensemble dans un seul bloc subsystem ce qui permet d'utiliser le système à un niveau supérieur sans avoir besoin de se pencher sur son fonctionnement interne. Au niveau supérieur, on ajoute une horloge pour alimenter le trigger du sous-système et régler sa fréquence de travail (cf Figure 3), puis on crée à partir de l'ensemble un bloc simulink fermé avec son ensemble de paramètres. Celui-ci réagit comme un bloc standard de simulink et permet d'utiliser la SBPA comme un simple composant d'un système plus complexe.

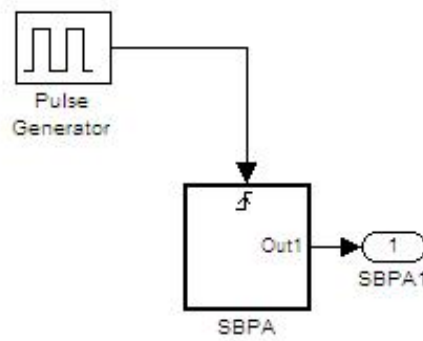


Figure 3: La SBPA encapsulée avec son horloge interne

On obtient au final un bloc produisant une séquence binaire pseudo-aléatoire et capable d'être réglée en fréquence et dont les valeurs de l'état bas et l'amplitude sont paramétrables.

### 3. Acquisition du signal

xPC Target est un outils de prototypage de système temps réel et de simulation temps réel de système physique (*hardware-in-the-loop*). Il utilise un concept hôte-cible pour transformer un ordinateur en plate-forme de simulation ce qui lui permet d'être plus simple et plus accessible qu'un contrôleur dédié. Couplé à simulink et real time workshop, il permet de mettre en place et d'ajuster rapidement et visuellement un système automatique. Simulink gère la représentation par blocs du système et facilite les calculs de l'automaticien puis real time workshop convertit automatiquement le modèle conçu en application pouvant être lancée par xPC Target sur la machine cible et pouvant agir directement, via des cartes d'acquisition sur un système physique.

Avant de procéder à l'acquisition, il faut préparer le système et le calibrer. On commence donc par faire en sorte que le correcteur ne reçoive en entrée que le signal destiné à l'acquisition. Pour cela, on enlève la bille du système de lévitation et on compense le signal de retour qui reste avec le bouton de réglage de la consigne. Ainsi leur somme est nulle et il ne reste que le signal utile.

Pour identifier le correcteur, il faut le soumettre à une entrée connue et mesurer sa sortie. Une séquence binaire pseudo-aléatoire sera utilisée en entrée pour les raisons évoquées au chapitre précédent. On utilise donc xPC Target pour générer ce signal d'entrée et récupérer les valeurs de sortie correspondantes, valeurs que l'on récupérera dans matlab en vue de les analyser.

On réalise pour cela le schéma simulink de la Figure 4 qui présente une SBPA dont la sortie est enregistrée dans un vecteur matlab, affichée dans un graphe sur l'écran de la machine cible et bien entendu, envoyé par un convertisseur numérique/analogique (PCI-6024E DA) vers l'entrée du correcteur. Ce signal traverse le correcteur réel et la sortie du correcteur est récupérée par un convertisseur analogique/numérique (PCI-6024E AD) puis enregistrée dans un vecteur matlab et affichée sur un second graphe de la machine cible. L'ensemble est cadencé par la fréquence d'échantillonnage définie par

*Les règles d'acquisition sont, dans l'univers de Star Trek, une série de 285 aphorismes, proverbes et principes à la base de la philosophie marchande de la culture ferengie. - Wikipedia*  
 16 - Un accord est un accord... qu'un plus profitable remplace.  
 47 - Ne fais jamais confiance à un homme qui porte un meilleur costume que le tien.  
 208 - Parfois la réponse est plus dangereuse que la question.  
 211 - Tes employés sont les échelons sur l'échelle du succès. N'hésite pas à leur marcher dessus.

Pechant = 5e-5;

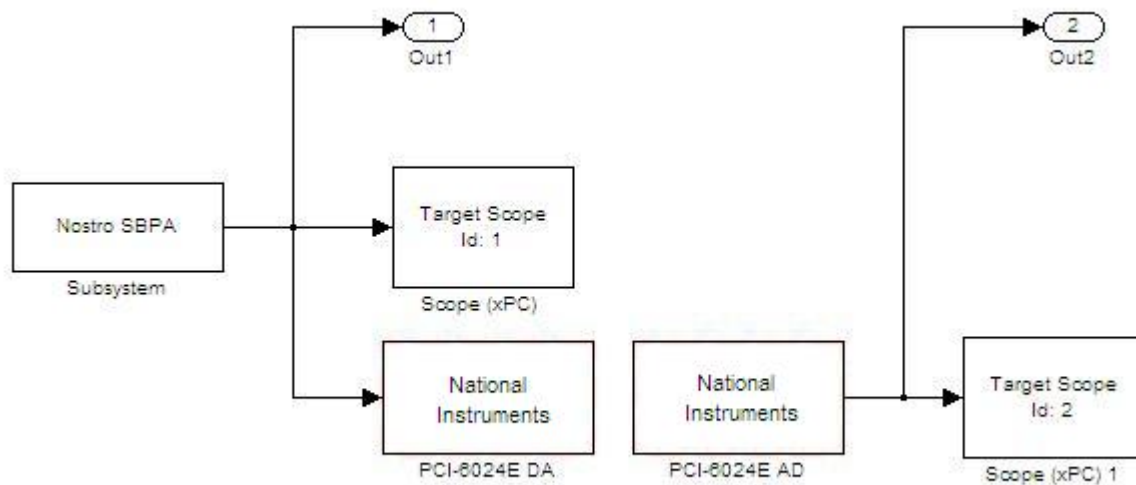


Figure 4: Schéma simulink utilisé pour l'acquisition de données via xPC Target

Une fois le système compilé, il est envoyé sur la machine cible et l'acquisition est lancée. Une fois qu'elle est terminée, l'interface de communication xPC Target permet de télécharger les données enregistrées ainsi que le vecteur temps et de les transformer en vecteurs accessibles dans matlab.

Il est également possible avec xPC Target de modifier certains paramètres du système simulink sans avoir besoin de recompiler tout le programme de simulation. On peut ainsi régler en particulier l'amplitude du signal d'entrée, d'une part pour éviter que le correcteur ne sature en sortie, et d'autre part pour obtenir facilement la réponse du correcteur face à une entrée d'amplitude différente.

Le correcteur implémenté est très sensible à l'amplitude d'entrée et il faut diminuer fortement l'amplitude pour s'affranchir des problèmes de saturation. Cela montre également que le correcteur fonctionne avec des signaux très faibles en entrée.

Les premières mesures effectuées sont donc à la limite de la saturation puis nous avons choisi de diminuer encore le gain en entrée et nous avons récupéré des données qui ne sont pas saturées.

## 4. Relevé des valeurs de point de fonctionnement

Étant donné que le correcteur analogique est non-linéaire, nous avons relevé, pour différents points de fonctionnement du système, la consigne  $z_0$ , la commande  $u_0$ , le courant dans la bobine  $i_0$ , afin de les intégrer dans la boucle :

$z_0$	-1,76	- 0,514	0,637	1,738	2,54
$u_0$	2,75	1,48	0,800	0,110	-0,743
$i_0$	0,92	0,71	0,60	0,49	0,34

En effet, l'identification ne se fait qu'autour d'un point de fonctionnement et le correcteur est prévu pour réagir aux variations de la commande autour de ce point de fonctionnement. Il faut donc fournir au modèle une valeur de commande de manière à ce que seules les variations soient corrigées.

À la sortie du correcteur, on n'obtient que des variations de l'entrée calculées à partir des variations



de commande, et non la valeur complète de l'entrée. Il faut donc relever pour chaque commande la valeur moyenne de l'entrée correspondante et l'ajouter à la variation calculée.

On relève également les valeurs du courant dans la bobine qui nous serviront plus tard lors de la linéarisation du modèle physique du système. Ces valeurs sont relevées manuellement à l'aide d'un voltmètre (oscilloscope), d'un ampèremètre et avec le potentiomètre ajustant la consigne. Les relevés se font avec la bille en lévitation pour que le système soit complet, et on doit donc se limiter au domaine dans lequel le système est stable, c'est à dire que la bille doit rester en lévitation sans se coller à l'aimant ou tomber au sol.

Un autre jeu de résultats est enregistré en utilisant xPC Target avec l'inconvénient qu'il n'est pas possible d'enregistrer les valeurs de  $i_0$  car la carte d'acquisition ne prend en charge que deux entrées.

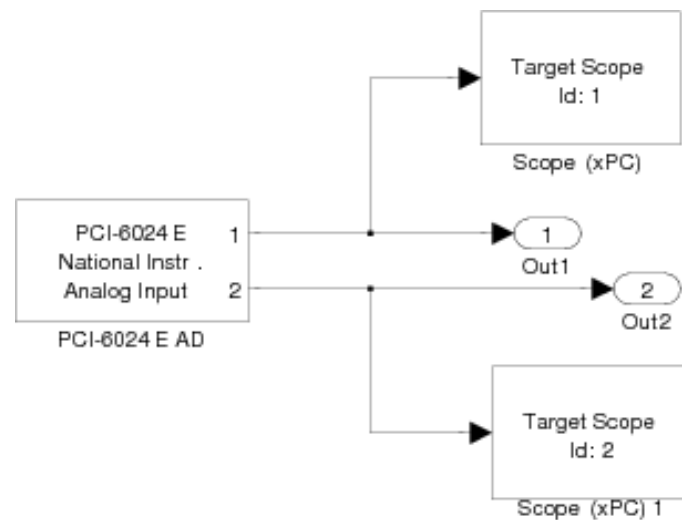


Figure 5: Schéma simulink pour l'acquisition de  $z_0$  et  $u_0$

La Figure 5 montre le schéma simulink utilisé pour l'acquisition de  $z_0$  et  $u_0$  par xPC Target. Il n'y a pas d'entrée fournie au système puisque l'on se sert toujours du potentiomètre de réglage. La valeur fournie par le potentiomètre, soit  $z_0$  est enregistrée dans la sortie Out1 tandis que  $u_0$  est enregistré dans Out2. Les échantillons sont prélevés deux fois par seconde.

## 5. Identification du correcteur

### 5.1. La boîte à outils ident

La boîte à outils ident est l'outil idéal pour identifier une série de mesures expérimentales à un modèle mathématique.

L'outil est relativement simple à prendre en main pour peu que l'on sache interpréter ses résultats. On utilise tout d'abord le menu déroulant Import data pour importer les données expérimentales dans ident. Ces données auront été préalablement transformées en données ident avec la fonction iddata qui combine en un seul objet l'entrée du système, sa sortie et la période d'échantillonnage des mesures. Les données importées peuvent être visionnées et contrôlées grâce au bouton Time plot.

*Dans l'acception psychologique de la notion, la forme d'identité la plus directement accessible est celle qu'a dégagée, en rupture avec la tradition cartésienne, la critique de la substantialité du moi. - L'identification, Encyclopédia Universalis*

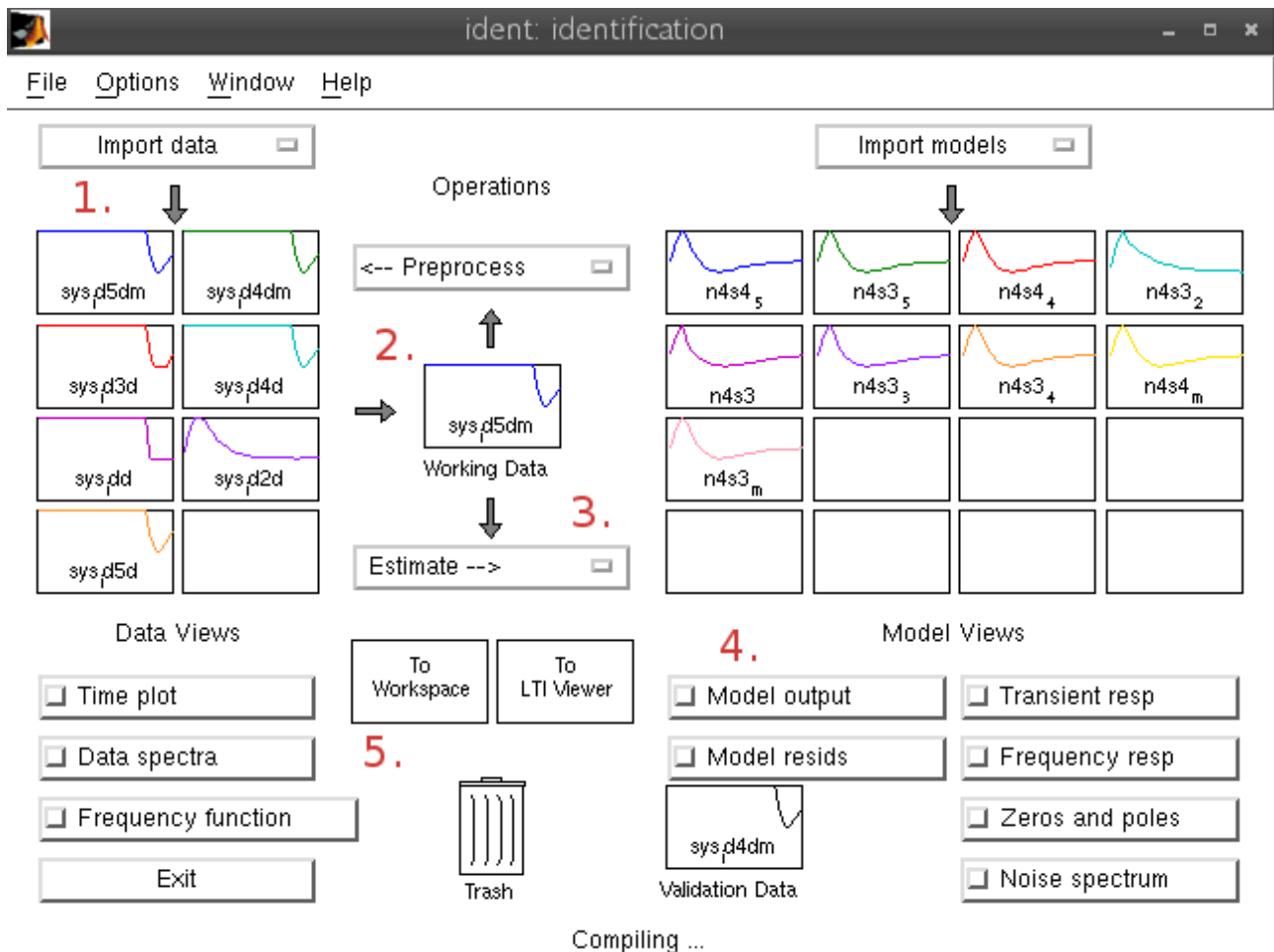


Figure 6: La boîte à outils *ident*

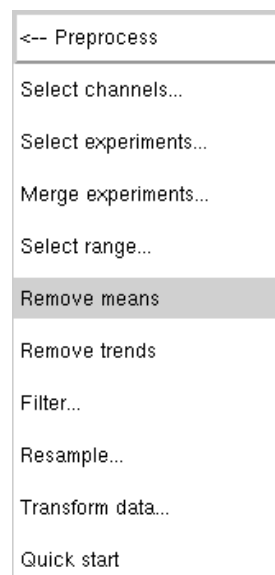


Figure 8: Les options du préprocesseur

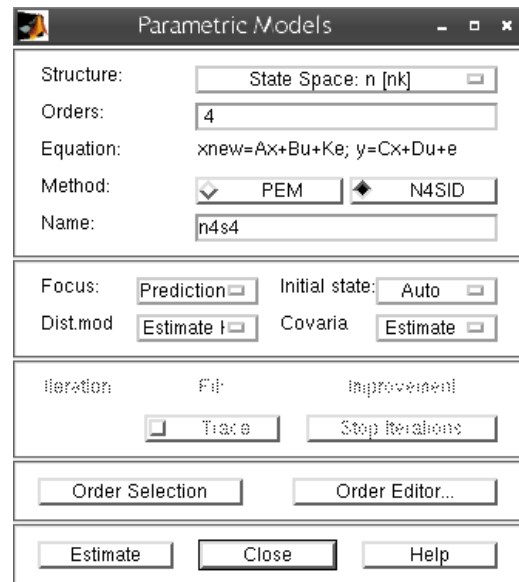


Figure 7: Dialogue de choix du type d'identification dans *ident*

Dans un deuxième temps, les données doivent être préparées et l'on utilise pour cela le Preprocess qui permet entre autre d'éliminer les composantes continues du signal ou encore de

combiner plusieurs sets de résultats pour augmenter la précision de l'identification ou vérifier un modèle plus finement. Les données épurées sont alors identifiées grâce à Estimate et l'on peut choisir le type d'identification à effectuer ainsi que son ordre (cf Figure 7). Enfin, il faut vérifier le modèle identifié en le comparant avec l'expérimentation et effectuer plusieurs identifications pour choisir la meilleure. Encore une fois, ident propose un moyen simple de le faire grâce au bouton Model output et à la case Validation data.

## 5.2. Identification des expérimentations

Nous avons choisi d'identifier les résultats par un système du troisième ou du quatrième ordre sous la forme d'équations d'état. Les premiers sets de mesures étant faussés car saturés, nous avons travaillé sur les derniers résultats. Les sets saturés nous ont en revanche servi plus tard, à valider le modèle choisi.

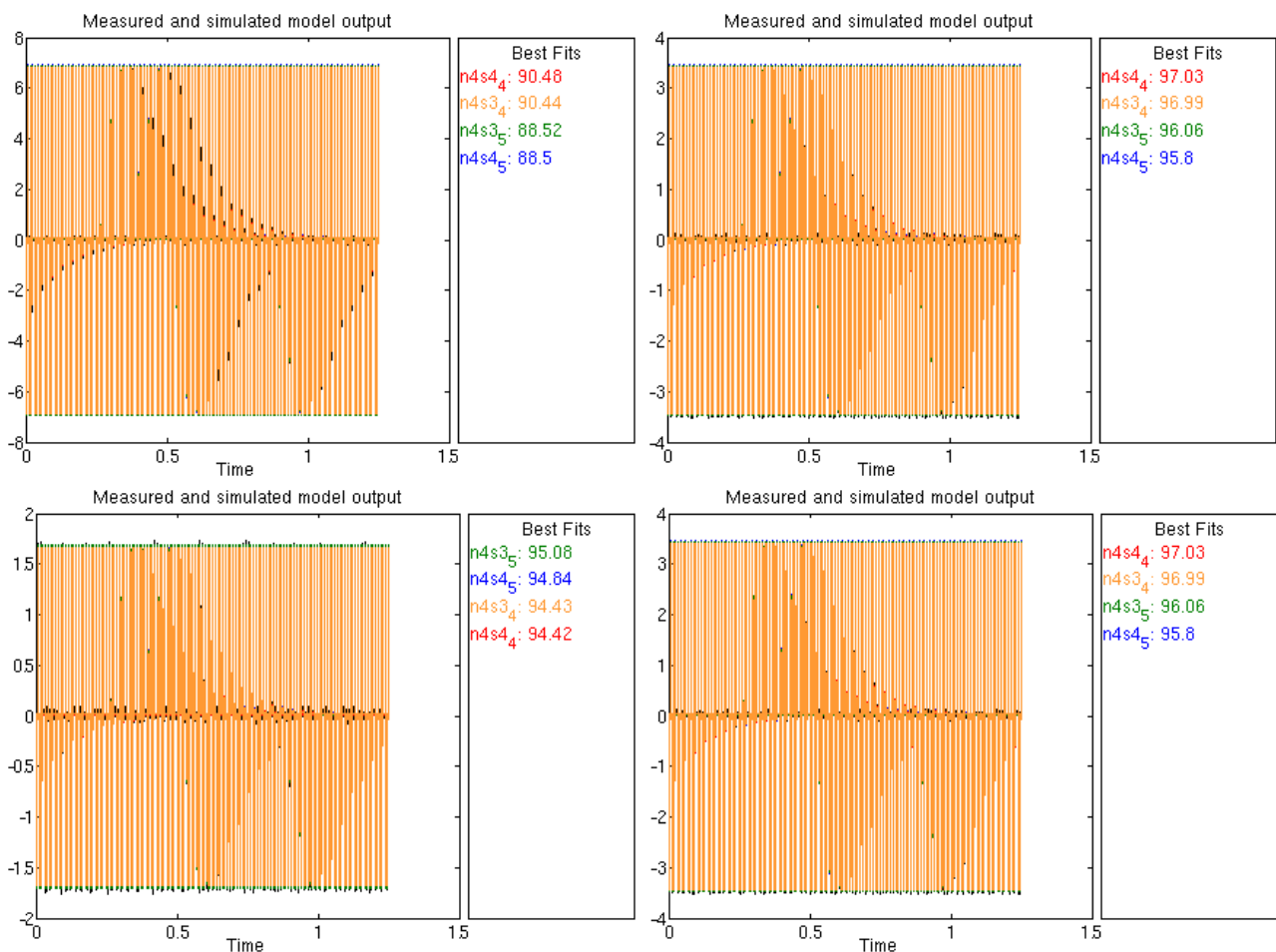


Figure 9: Comparaisons des modèles identifiés avec les expérimentations

Les différentes identifications sont comparées dans la Figure 9. Seules les identifications les plus pertinentes ont été gardées. Le schéma pour les noms est le suivant : l'ordre du système est donné par le chiffre qui suit la lettre 's' et le numéro en indice est celui de la série de données qui a permis l'identification. Les identifications sont comparées de droite à gauche et de bas en haut à la série 3, 4, 5 et aux trois séries combinées. Il s'en dégage que le modèle  $n4s4_4$  semble le plus approprié mais également qu'un modèle d'ordre 3 donne aussi de bons résultats, voire meilleurs lorsqu'il s'agit d'identifier la 5<sup>ème</sup> série de mesures.

On choisit finalement le modèle d'ordre 4 qui donne de meilleurs résultats sur l'ensemble des mesures effectuées, bien que celles-ci soient parfois saturées.

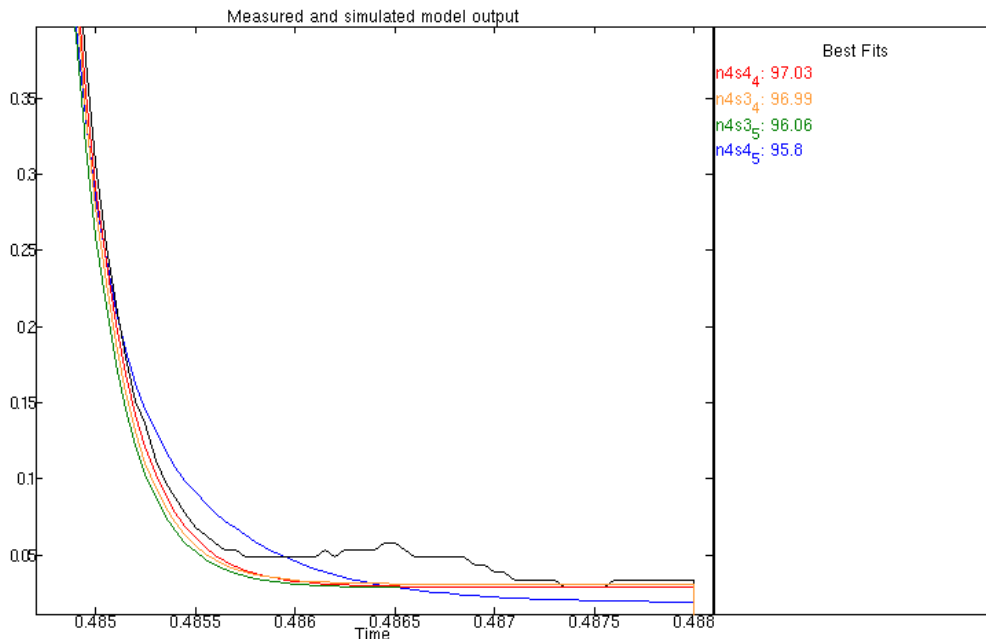


Figure 10: Agrandissement d'un point critique de l'identification

## 6. Implémentation du correcteur numérique sur le système réel

### 6.1 Implémentation du correcteur à l'aide de *simulink*

Une fois que l'on est satisfait d'une identification, on peut envoyer le système vers matlab en le glissant vers To workspace dans ident. Quelques commandes permettent alors de le rendre utilisable directement dans simulink.

```
sys = ss(n4s3.A, n4s3.B, n4s3.C, n4s3.D, n4s3.Ts) ;
sys_tf = tf(sys);
```

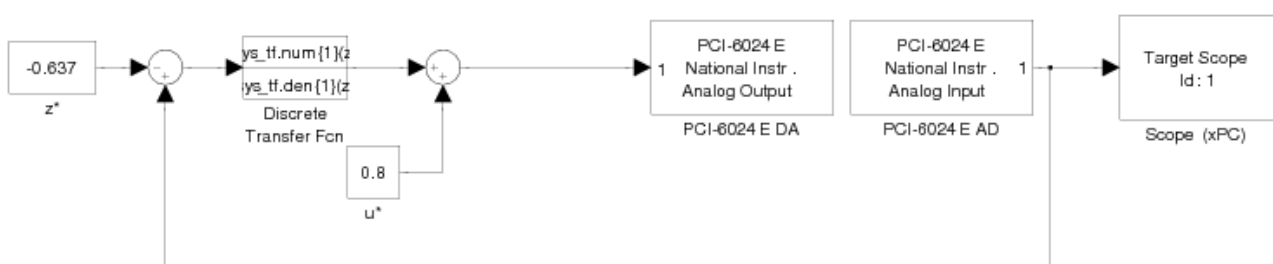


Figure 11: Schéma *simulink* d'implémentation du correcteur numérique

Il ne faut pas oublier que le correcteur analogique est non-linéaire. C'est pour cette raison que l'on intègre dans la boucle la valeur de la consigne au point de fonctionnement et la valeur de la commande correspondante. Le correcteur numérique ne travaille lui, que sur l'écart à ces valeurs, c'est à dire qu'il associe à un écart de la consigne un écart de la commande. Le système réel s'insère entre la sortie PCI-DA et l'entrée PCI-AD.

## 6.2. Validation du correcteur sur le système réel

Une fois le correcteur implémenté sous simulink et la compilation réalisée avec succès, on peut transférer le système sur la cible XPC Target et lancer l'application. On constate immédiatement que la bille est plus difficile à placer pour que le système réagisse. En effet, il faut d'abord l'approcher du point de fonctionnement pour que le correcteur soit capable de la stabiliser. Notre correcteur est assez sensible au placement initial de la bille mais réagit bien aux perturbations une fois le système stabilisé. Plusieurs points de fonctionnements sont essayés ainsi que plusieurs correcteurs. Cependant, la mise en œuvre du système ne permet pas d'obtenir des courbes et d'évaluer quantitativement les stabilités de chacun des correcteurs. Qualitativement, ces correcteurs sont proches.

*Qu'est-ce qui est plus amusant que de faire tourner des enfants sur un tourniquet ?*

*Les arrêter avec une pelle ?*

## 7. Linéarisation du système physique

Pour calculer un nouveau correcteur, il faut étudier plus précisément le système physique et le linéariser autour d'un point de fonctionnement. Pour linéariser le système, on commence par écrire la matrice jacobienne du système dans sa représentation d'état.

Le système s'écrit :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = g - \frac{L_o}{2mx_1^2} x_3^2 \\ \dot{x}_3 = \frac{L_o}{x_1(L_o + L_1 x_1)} x_2 x_3 - \frac{R}{L_o + L_1 x_1} x_1 x_3 + \frac{x_1}{L_o + L_1 x_1} (A_{amp} U_{com} + B) \end{cases}$$

d'où les dérivées partielles pour  $X = [x_1 \ x_2 \ x_3]^T$  :

$$\frac{\partial \dot{x}_1}{\partial x_1} = 0 \quad \frac{\partial \dot{x}_2}{\partial x_1} = \frac{L_o}{m x_1^3} x_3^2 \quad \frac{\partial \dot{x}_3}{\partial x_1} = \frac{-L_o x_2 x_3 (L_o + 2 L_1 x_1)}{x_1 (L_o + L_1 x_1)^2} + \frac{R x_3 (L_o + L_1 x_1) - R L_1 x_1 x_3}{(L_o + L_1 x_1)^2} + \frac{(L_o + L_1 x_1) - x_1 L_1}{(L_o + L_1 x_1)^2} (A_{amp} U_{com} + B)$$

$$\frac{\partial \dot{x}_1}{\partial x_2} = 1 \quad \frac{\partial \dot{x}_2}{\partial x_2} = 0 \quad \frac{\partial \dot{x}_3}{\partial x_2} = \frac{L_o}{x_1 (L_o + L_1 x_1)} x_3$$

$$\frac{\partial \dot{x}_1}{\partial x_3} = 0 \quad \frac{\partial \dot{x}_2}{\partial x_3} = \frac{-L_o}{m x_1^2} x_3 \quad \frac{\partial \dot{x}_3}{\partial x_3} = \frac{L_o}{x_1 (L_o + L_1 x_1)} x_2 - \frac{R}{L_o + L_1 x_1} x_1$$

ainsi que les dérivées partielles en  $u = [U_{com}]$  :

$$\frac{\partial \dot{x}_1}{\partial U_{com}} = 0 \quad , \quad \frac{\partial \dot{x}_2}{\partial U_{com}} = 0 \quad , \quad \frac{\partial \dot{x}_3}{\partial U_{com}} = A_{amp} \frac{x_1}{L_o + L_1 x_1}$$

On en déduit la jacobienne

$$A^* = \begin{pmatrix} 0 & 1 & 0 \\ \frac{L_0}{m x_1^3} x_3^2 & 0 & \frac{-L_0}{m x_1^2} x_3 \\ \frac{\partial \dot{x}_3}{\partial x_1} & \frac{L_0}{x_1(L_0 + L_1 x_1)} x_3 & \frac{L_0}{x_1(L_0 + L_1 x_1)} x_2 - \frac{R}{L_0 + L_1 x_1} x_1 \end{pmatrix}_{x_{10} x_{20} x_{30} U_{com0}}$$

et  $B^* = \begin{pmatrix} 0 \\ 0 \\ A_{amp} \frac{x_1}{L_0 + L_1 x_1} \end{pmatrix}_{x_{10} x_{20} x_{30} U_{com0}}.$

On applique alors les valeurs relevées du point de fonctionnement par un calcul matlab :

```
% Calcul d'un correcteur PID pour le système linéarisé.

% Linéarisation du système

% Point de fonctionnement

z_etoile = -0.637;
u_etoile = 0.8;
i_m = 0.60;

% Constantes

alpha = 358;
beta = 6.77;
L_0 = 2.49e-4;
L_1 = 0.612;
Aamp= 3.407;
Bamp = 9.54;
m = 21.5e-3;
R = 20.4;

% Variables au point de fonctionnement

x_1 = (z_etoile - beta) / alpha;
x_2 = 0;
x_3 = i_m;
u = u_etoile;

% Calcul de la jacobienne

dx1_x1 = 0;
dx1_x2 = 1;
dx1_x3 = 0;

dx2_x1 = (x_3^2 * L_0) / (m * x_1^3);
dx2_x2 = 0;
dx2_x3 = - (x_3 * L_0) / (m * x_1^2);

dx3_x1 = - ((L_0 * x_2 * x_3) * (L_0 + 2 * L_1 * x_1)) / ((x_1 * L_0 + L_1 * x_1^2)^2);
dx3_x1 = dx3_x1 - R * x_3 * ((L_0 + 4 * x_1) - R * x_1 * x_3 * L_1) / ((L_0 + L_1 * x_1)^2);
```

```

dx3_x1 = dx3_x1 + ((L_0 + L_1 * x_1) - (x_1 * L_1)) / ((L_0 + L_1 * x_1)^2) * (Aamp * u + Bamp);

dx3_x2 = (L_0 * x_3) / (x_1 * (L_0 + L_1 * x_1));
dx3_x3 = (L_0 * x_2) / (x_1 * (L_0 + L_1 * x_1)) - (R * x_1) / (L_0 + L_1 * x_1);

dx1_u = 0;
dx2_u = 0;
dx3_u = (x_1 / (L_0 + L_1 * x_1)) * Aamp;

A = [ dx1_x1, dx1_x2, dx1_x3; dx2_x1, dx2_x2, dx2_x3; dx3_x1, dx3_x2, dx3_x3];
B = [ dx1_u; dx2_u; dx3_u ];
C = [ alpha, 0, 0 ];
D = [ 0 ];

% Calcul du système

systeme = ss(A, B, C, D);
sys_tf = tf(systeme);

% Calcul du correcteur

% par rltool

```

qui nous donne

$$A = \begin{pmatrix} 0 & 1 & 0 \\ -470.7 & 0 & -16.23 \\ -5737 & 0.5817 & -34 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ 5,679 \end{pmatrix}, \quad C = \begin{pmatrix} 358 & 0 & 0 \end{pmatrix}$$

et  $D=0$ .

PLUS QU'UN CHAPITRE !!!

## 8. Correction par PID et essais en réel

Grâce à ce modèle linéarisé, on peut simuler la correction du système par un correcteur proportionnel intégral dérivé. On utilise l'utilitaire `rltool` de la boîte `SISO Tool` pour calculer automatiquement un premier correcteur PID que l'on tente ensuite d'améliorer.

- Une mère dit à son garçon :*  
*- tu sais, nous sommes sur terre pour travailler...*  
*- Ben moi, plus tard je serai marin dans ce cas !*
- On obtient un système comportant quatre pôles complexes conjugués, deux lents et deux rapides. On cherche cependant un système qui ne répond pas forcément rapidement mais surtout pas violemment car le système physique a un domaine de stabilité très petit (bille collée à l'aimant ou tombant au sol). Il faut donc de préférence que la réponse soit modérée et que le correcteur réagisse sans violence tout en étant suffisamment rapide pour empêcher la bille de sortir des bornes.
- On constate que ce comportement est difficile à atteindre car lorsque l'on réduit l'influence des pôles rapides, les pôles lents prennent de l'ampleur et le dépassement reste élevé, et lorsque l'on réduit les pôles lents, les pôles rapides reprennent de l'influence voire deviennent positifs.

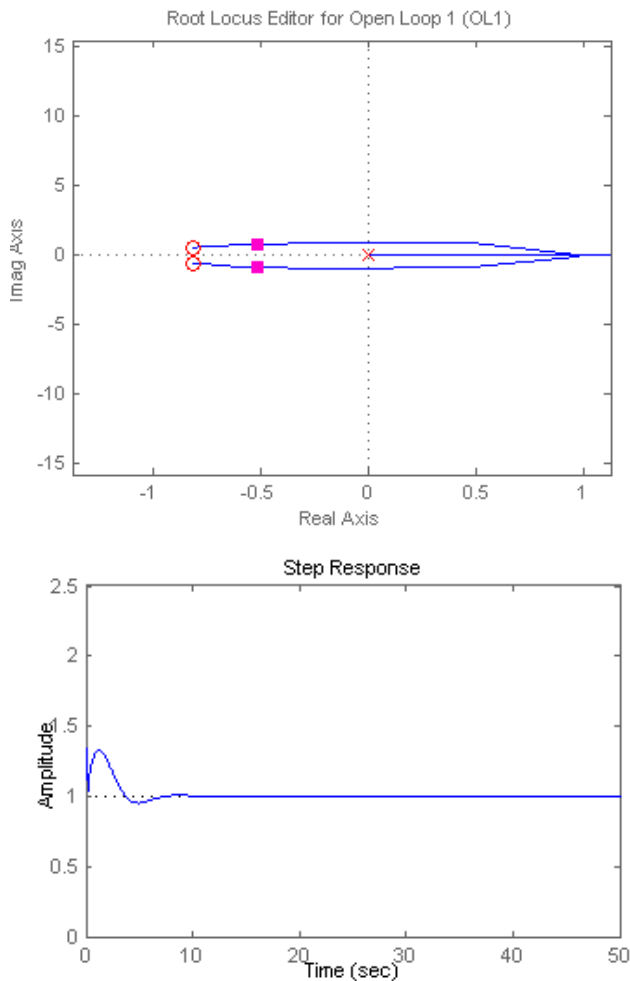


Figure 12: Lieu des racines et réponse indicielle lorsque les modes rapides sont actifs

Note : La réponse présente une forte oscillation durant les premières secondes ce qui n'est pas très visible sur la figure. Le premier dépassement s'élève à 2.5 volts.

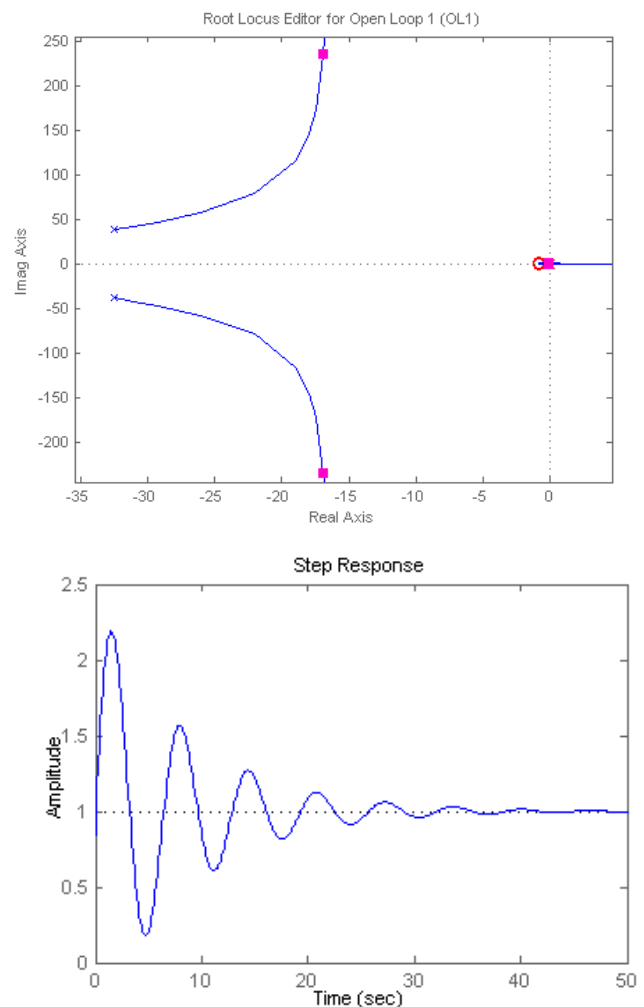


Figure 13: Lieu des racines et réponse indicielle lorsque l'effet des pôles rapides est atténué

Sur la Figure 12, les pôles rapides sont présents et bien que le système se stabilise rapidement (environ 10 secondes), des oscillations brutales se produisent au début de la réponse ce qui amène forcément la bille à sortir des limites de stabilité. Par contre, lorsque les modes rapides sont atténués (Figure 13), il n'y a plus d'oscillations violentes en début de réponse. En revanche, les modes lents entraînent le système dans une oscillation longue et de forte amplitude. Non seulement le système met beaucoup de temps à se stabiliser (plus de 40 secondes) mais le premier dépassement est au-dessus de 2.2 volts ce qui envoie encore une fois la bille en dehors des limites.

La régulation par PID ne semble donc pas très indiquée pour corriger ce système ou du moins pas sous cette forme (on peut envisager des boucles multiples permettant de régler séparément un retour proportionnel, intégral puis dérivé). D'autre part, il semble périlleux de corriger ce système avec en utilisant des moyens numériques. En effet, le besoin de réactivité et de travail sur de très faibles amplitudes est difficilement compatible avec l'échantillonnage et la discrétisation.

*Quelle est la différence entre un roi et un taureau ?  
L'arène.  
entièrement dans  
Le taureau rentre*



Ce bureau d'étude est un bon exemple d'analyse et de correction d'un système non linéaire depuis le début jusqu'à la fin. Il illustre les techniques les plus courantes d'identification, met en oeuvre les outils pour le faire, et permet d'appréhender la conception d'un correcteur pour un système non linéaire. Il permet également de nous familiariser avec les outils que met matlab à disposition de l'automaticien pour concevoir ses correcteurs, comme le couple simulink/xPC Target ou encore ident et rltool.

Le système étudié est intéressant d'une part parce qu'il est amusant (il défie les lois de la physique) et également parce qu'il est suffisamment complexe tout en étant abordable. Il est cependant dommage que la mise en oeuvre d'xPC Target soit complexe et que les ordinateurs soient parfois vétustes car nous avons tout de même perdu le temps d'une séance, ce qui nous aurait été utile pour améliorer le correcteur PID réalisé sous rltool.

*Mais bon, le prof est sympa...*

Ce rapport est également disponible à l'adresse

<http://www.cilyan.org/misc/work/RapportBeAutomatique2008.pdf>

