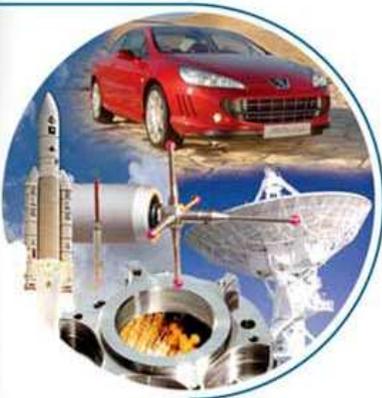


Identification des systèmes

Gonzalo Cabodevila

gonzalo.cabodevila@femto-st.fr

3^{ème} année



**École Nationale Supérieure de
Mécanique et des Microtechniques**
26, chemin de l'Épitaphe
25030 Besançon cedex – FRANCE
<http://intranet-tice.ens2m.fr>

" Je distingue deux moyens de cultiver les sciences : l'un d'augmenter la masse des connaissances par des découvertes ; et c'est ainsi qu'on mérite le nom d'inventeur ; l'autre de rapprocher les découvertes et de les ordonner entre elles, afin que plus d'hommes soient éclairés, et que chacun participe, selon sa portée, à la lumière de son siècle ... "

Diderot

Table des matières

I	Identification des systèmes	9
1	Modèles de connaissance	11
1.1	Modèles simples linéaires	11
1.1.1	Exemple : le moteur à courant continu	11
1.2	Méthodes systématiques	12
1.2.1	Un exemple simple	13
1.3	Détermination des constantes	14
1.4	Linéarisation	14
1.5	Conclusions	14
2	Identification de modèles non paramétriques	15
2.1	Quelle entrée ?	15
2.1.1	Analyse harmonique	15
2.1.2	Réponse impulsionnelle	15
2.1.3	Réponse indicielle	15
2.2	Etude des réponses indicelles	16
2.2.1	Systèmes du premier ordre	16
2.2.2	Systèmes du second ordre résonnant	16
2.2.3	Systèmes du premier ordre retardés ou du second ordre non résonnant	17
2.2.4	Systèmes d'ordre supérieur à 2 non résonnants	18
2.2.5	Système avec intégrateur	20
2.3	Autres modèles ?	20
2.4	Méthode de Ziegler-Nichols	21
3	Algorithme général d'identification	23
3.1	Choix du signal d'excitation	24
3.2	Les séquences binaires pseudo aléatoires (SBPA)	24
3.2.1	Choix des paramètres d'une SBPA	26
3.3	Identification basée sur l'erreur de sortie	27
3.4	Identification basée sur l'erreur de prédiction	28
3.4.1	Méthode des moindres carrés simples	28
3.4.2	Calcul du biais de l'estimateur	29
3.4.3	Méthode des moindres carrés généralisés	31
3.4.4	Méthode de la Matrice Instrumentale	33
3.5	Identification en boucle fermée	33
3.6	Et Matlab [®] dans tout ça ?	34
4	Estimations récursives	37
4.1	Moindres carrés récursifs	37
4.2	Variable instrumentale récursive	39

5	Algorithmes d'optimisation paramétrique.	41
5.1	Présentation générale.	41
5.2	Exemple introductif	41
5.3	Problème posé	42
5.4	Revue bibliographique succincte.	42
5.5	Eléments de comparaison des différentes méthodes.	43
5.5.1	Critères.	43
5.5.2	Comparaisons.	43
5.5.3	Récapitulatif.	44
5.5.4	Synthèse	44
6	Description des algorithmes utilisés.	47
6.1	Gradient et quasi-Newton	47
6.1.1	Méthode du gradient	47
6.1.2	Méthode de Newton	48
6.1.3	Méthode de Newton-Raphson	48
6.1.4	Méthode de Levenberg-Markardt	48
6.1.5	l'algorithme BFGS	49
6.2	Simplex de Nelder & Mead.	50
6.2.1	Extensions - Améliorations.	52
6.3	Algorithmes génétiques.	54
6.3.1	Introduction.	54
6.3.2	Principe des algorithmes génétiques binaires.	54
6.3.3	Extensions - Améliorations.	55
6.3.4	Conclusion.	55
6.4	Algorithmes génétiques codés réels.	55
6.4.1	Principe.	55
6.4.2	Conclusion.	56
6.5	Méthode du "Recuit simulé"	57
6.5.1	Introduction	57
6.5.2	Principe.	57
6.5.3	Extensions - Améliorations.	57
6.5.4	Mise en œuvre du "recuit simulé".	58
	Bibliographie	58
7	Travaux dirigés	61
7.1	Strecj - Broida - Programmation non linéaire	61
7.1.1	Strecj - Broida	61
7.1.2	Programmation non linéaire	61
7.1.3	Limites de la Programmation non linéaire	61
7.2	Identification des systèmes instables ou intégrateurs	61
7.3	Choix de la source d'excitation	62
7.4	Les modèles paramétriques	62
7.5	Cas pratique	63
II	Annexes	65
	Examen final janvier 2004	67
	Correction examen final janvier 2004	70
	Examen final janvier 2005	73
	Examen final janvier 2006	77
	Correction examen final janvier 2006	81

<i>TABLE DES MATIÈRES</i>	7
Examen final janvier 2007	83
Correction examen final janvier 2007	86
Examen final janvier 2008	90

Première partie

Identification des systèmes

Chapitre 1

Modèles de connaissance

1.1 Modèles simples linéaires

Ce sont les modèles issus de la physique. Lorsque le système est peu complexe il est possible d'écrire les relations entre les différentes grandeurs physiques décrivant les différents sous-systèmes.

1.1.1 Exemple : le moteur à courant continu

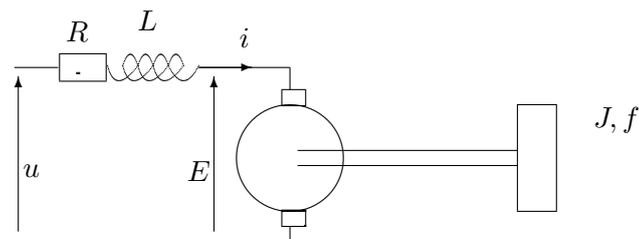


FIGURE 1.1 – Schéma électromécanique d'un moteur à courant continu

$$\text{avec : } \begin{cases} u & = E + Ri + Ldi/dt \\ C & = ki \\ E & = k\omega \\ J \frac{d\omega}{dt} & = C - f\omega \end{cases}$$

De l'ensemble des équations on peut tirer différents graphes :

- Schéma bloc
- Graphe de fluence (voir polycopié de cours de Bernard Lang)
- Graphes à Liens ou *Bond graph* (voir figure 1.2) (Hors du cadre de ce cours : cf Techniques de l'ingénieur et autres bons bouquins sur la question)

De ces graphes, par réduction, on obtient la fonction de transfert, préliminaire à la majorité des méthodes d'analyse et de synthèse de correcteurs que vous connaissez.

La figure 1.3 rappelle les analogies entre les différents domaines de la physique.

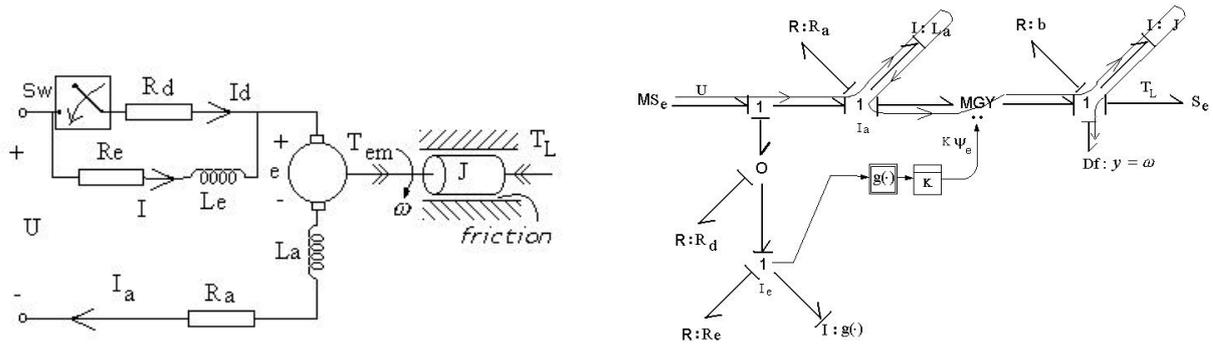


FIGURE 1.2 – Système et son modèle en Bond Graph. (tiré de : <http://www.univ-lille1.fr/l2ep/commande/ee/presentations/p-cs.htm>)

E énergie P puissance	ELECTRICITÉ	MÉCANIQUE		HYDRAULIQUE	THERMIQUE
		translation	rotation		
effet résistif R, f, R_h, R_t (dissipateur d'énergie)	résistance (R) $V_2 \rightarrow V_1$ $i = \frac{1}{R} V_{21}; \varphi = \frac{1}{R} V_{21}^2$ $i = \text{intensité } V = \text{tension}$	amortisseur (f) $F \rightarrow v_1$ $F = f v_{21}; \varphi = f v_{21}^2$ $F = \text{force } v = \text{vitesse}$	amortisseur (f) $T \rightarrow \omega_1$ $T = f \omega_{21}; \varphi = f \omega_{21}^2$ $T = \text{couple } \omega = \text{vitesse}$	résistance (R_h) $P_2 \rightarrow P_1$ $Q = \frac{1}{R_h} P_{21}; \varphi = \frac{1}{R_h} P_{21}^2$ $Q = \text{débit } P = \text{pression}$	résistance (R_t) $T_2 \rightarrow T_1$ $q = \frac{1}{R_t} T_{21}; \varphi = \frac{1}{R_t} T_{21}^2$ $q = \text{flux chal. } T = \text{températ.}$
effet inductif $L, I/K, I$ (stockage)	inductance (L) $V_2 \rightarrow V_1$ $v_{21} = L \frac{di}{dt}; E = \frac{1}{2} L i^2$	ressort (K) $v_2 \rightarrow F$ $v_{21} = \frac{1}{K} \frac{dF}{dt}; E = \frac{1}{2} \frac{F^2}{K}$	torsion (K) $\omega_2 \rightarrow T$ $\omega_{21} = \frac{1}{K} \frac{dT}{dt}; E = \frac{1}{2} \frac{T^2}{K}$	inertie (I_h) $P_2 \rightarrow P_1$ $P_{21} = I_h \frac{dQ}{dt}; E = \frac{1}{2} I_h Q^2$	inertie (I_t) $T_2 \rightarrow T_1$ $T_{21} = I_t \frac{dq}{dt}; E = \frac{1}{2} I_t q^2$
effet capacitif C, M, J, C_h, C_t (stockage)	capacité (C) $V_2 \rightarrow V_1$ $i = C \frac{dV_{21}}{dt}; E = \frac{1}{2} C V_{21}^2$	masse (M) $F \rightarrow v_1$ $F = M \frac{dv_{21}}{dt}; E = \frac{1}{2} M v_{21}^2$	inertie (J) $T \rightarrow \omega_1$ $T = J \frac{d\omega_{21}}{dt}; E = \frac{1}{2} J \omega_{21}^2$	capacité (C_h) hydraulique $Q \rightarrow P_1$ $Q = C_h \frac{dP_{21}}{dt}; E = \frac{1}{2} C_h P_{21}^2$	capacité (C_t) $q \rightarrow T_1$ $q = C_t \frac{dT_{21}}{dt}; E = \frac{1}{2} C_t T_{21}^2$

FIGURE 1.3 – Analogie des équations différentielles linéaires.

1.2 Méthodes systématiques

Lorsque le système s'avère plus complexe, il est souvent utile de mettre en œuvre une méthode *systématique*. Les équations de Lagrange sont l'une de ses méthodes, particulièrement adaptée dans le cadre de la robotique mais presque tous les systèmes sont modélisables par cette méthode. Les équations de Lagrange s'écrivent sous la forme :

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} + \frac{\partial V}{\partial q_i} = Q_i$$

- avec :
- $T = \Sigma$ énergies cinétiques
 - $D = \frac{1}{2} \Sigma$ pertes dépendantes de la vitesse
 - $V = \Sigma$ énergies potentielles
 - $Q =$ Forces extérieures

1.2.1 Un exemple simple

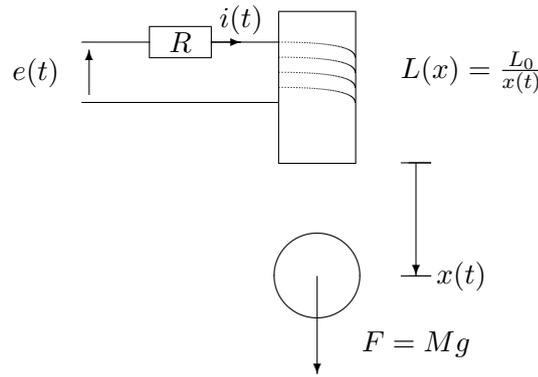


FIGURE 1.4 – Sustentation magnétique.

On suppose $L(x) = \frac{L_0}{x}$

	Mécanique	Electrique
Coordonnées	$q_2 = x(t)$	$q_1 = q(t)$
Vitesse	$\dot{q}_2 = \dot{x}(t)$	$\dot{q}_1 = \dot{q}(t) = i(t)$
Travail des forces extérieures	$\tau_2 = Mgx(t)$	$\tau_1 = e(t) \cdot q(t) = e \cdot q_1$
Forces extérieures	$Q_2 = \frac{\partial \tau_2}{\partial q_2} = Mg$	$Q_1 = \frac{\partial \tau_1}{\partial q_1} = e$
Energie cinétique	$T_2 = \frac{1}{2} M \dot{q}_2^2$	$T_1 = \frac{1}{2} L(x) \dot{q}_1^2 = \frac{1}{2} \frac{L_0}{q_2} \dot{q}_1^2$
Energie potentielle	$V_2 = 0$	$V_1 = 0$ (pas de capacité)
Fonction de dissipation	$D_2 = 0$	$D_1 = \frac{1}{2} R \dot{q}_1^2$
$T = T_2 + T_1 = \frac{1}{2} M \dot{q}_2^2 + \frac{1}{2} \frac{L_0}{q_2} \dot{q}_1^2$		
$V = V_2 + V_1 = 0$		
$D = D_2 + D_1 = \frac{1}{2} R \dot{q}_1^2$		
$\frac{\partial T}{\partial \dot{q}_i}$	$M \dot{q}_2$	$\frac{L_0}{q_2} \dot{q}_1$
$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right)$	$M \ddot{q}_2$	$L_0 \left(\frac{\ddot{q}_1 q_2 - \dot{q}_1 \dot{q}_2}{q_2^2} \right)$
$\frac{\partial D}{\partial \dot{q}_2}$	0	$R \dot{q}_1$
$\frac{\partial V}{\partial q_i}$	0	0
$\frac{\partial T}{\partial q_i}$	$\frac{L_0}{2} \left(-\frac{\dot{q}_1^2}{q_2^2} \right)$	0
Qi	Mg	e

d'où on en déduit les formules suivantes :

$$M \ddot{q}_2 + \left(\frac{L_0}{2} \right) \left(\frac{\dot{q}_1^2}{q_2^2} \right) = Mg$$

$$L_0 \left(\frac{\ddot{q}_1 q_2 - \dot{q}_1 \dot{q}_2}{q_2^2} \right) + R \dot{q}_1 = e$$

$$\ddot{q}_2 = g - \left(\frac{L_0}{2M} \right) \left(\frac{\dot{q}_1^2}{q_2^2} \right)$$

$$\ddot{q}_1 = \frac{q_2}{L_0} \left(e - R \dot{q}_1 + \frac{L_0 \dot{q}_1 \dot{q}_2}{q_2^2} \right)$$

L'utilisation de logiciels de calcul analytique (Maple, Mathematica, ...) permettent une modélisation rapide et sans erreurs.

1.3 Détermination des constantes

Evidemment, avant de pouvoir simuler le système, il faut déterminer la valeur numérique des différents paramètres entrant en jeu. La bonne méthode consiste alors à soumettre le système à une série de mesures permettant d'"isoler" le paramètre que l'on désire mesurer. La littérature regorge de méthodes permettant des mesures fiables.

ATTENTION : l'utilisation de méthodes de type "identification paramétrique" (voir chapitre 3) ne permet en aucun cas de mesurer ces paramètres !

1.4 Linéarisation

Ce système étant non linéaire et notre but étant d'en déterminer un modèle linéaire (fonction de transfert, représentation d'état), la première étape consiste donc à le linéariser. Il existe d'innombrables méthodes de linéarisation, la littérature vous fournira une méthode appropriée à votre problème. Je n'en présenterai ici qu'une :

la linéarisation autour d'un point de fonctionnement.

Cela consiste en un développement limité au premier ordre ($a = a_0 + \hat{a}$) autour d'un point.

En reprenant l'exemple du paragraphe 1.2.1

$$\ddot{q}_1 = F_1(q_1, q_2, \dot{q}_1, \dot{q}_2, e)$$

La linéarisation autour du point E est donnée par :

$$\ddot{\hat{q}}_1 = \left. \frac{\partial F_1}{\partial q_1} \right|_E \hat{q}_1 + \left. \frac{\partial F_1}{\partial q_2} \right|_E \hat{q}_2 + \left. \frac{\partial F_1}{\partial \dot{q}_1} \right|_E \dot{\hat{q}}_1 + \left. \frac{\partial F_1}{\partial \dot{q}_2} \right|_E \dot{\hat{q}}_2 + \left. \frac{\partial F_1}{\partial e} \right|_E \hat{e}$$

Notez bien que cette étape de linéarisation n'est nécessaire que si vous comptez utiliser une méthode de synthèse fondée sur une représentation linéaire !

Concernant la simulation : implantez les deux modèles (linéarisé et non-linéarisé), le premier vous permettra de vérifier votre synthèse de correcteur, le second (un peu plus proche de la réalité) vous permettra de simuler la robustesse de votre correcteur vis-à-vis des non-linéarités du système.

1.5 Conclusions

Il faut toujours avoir un modèle de connaissances, si mauvais soit-il pour pouvoir "sentir" le procédé et interpréter vos résultats. Ce modèle de connaissances est le préalable à toute identification de système quelle que soit la méthode employée.

Chapitre 2

Identification de modèles non paramétriques

Si le modèle connaissance s'avère trop peu précis ou impossible à obtenir, on peut alors se sur un modèle de comportement entrée - sortie. On parle alors d'*identification* du système. L'objectif est d'obtenir un modèle qui "se comporte comme" le système. La plupart du temps, les paramètres de ce modèles n'ont qu'un rapport lointain avec les paramètres d'un modèle de connaissances.

En premier lieu, il faut donc répondre aux deux questions suivantes :

- Quelle entrée ?
- Comment traiter la sortie ?

2.1 Quelle entrée ?

2.1.1 Analyse harmonique

Entrée sinusoïdale de type $u = A\sin(\omega t)$, ω balaye l'espace des pulsations susceptibles de contenir une pulsation de coupure du système. En notant l'amplitude et le déphasage de la sortie vis-à-vis de l'entrée on trace un diagramme de Bode. De l'analyse de ce diagramme on détermine le modèle

Les résultats sont difficiles à exploiter si les constantes de temps sont proches. Bonne excitation sur tout le spectre de fréquences. Ce n'est pas une commande industrielle classique et par conséquent elle est difficile à mettre en oeuvre.

2.1.2 Réponse impulsionnelle

Idéalement la meilleure méthode car le spectre est constant. Mais il est impossible de réaliser un Dirac parfait. Une bonne approximation du Dirac peut être obtenue en électronique (générateur d'impulsions) et en mécanique (percussion souvent utilisée en mécanique vibratoire) voire explosion dans le domaine du génie civil.

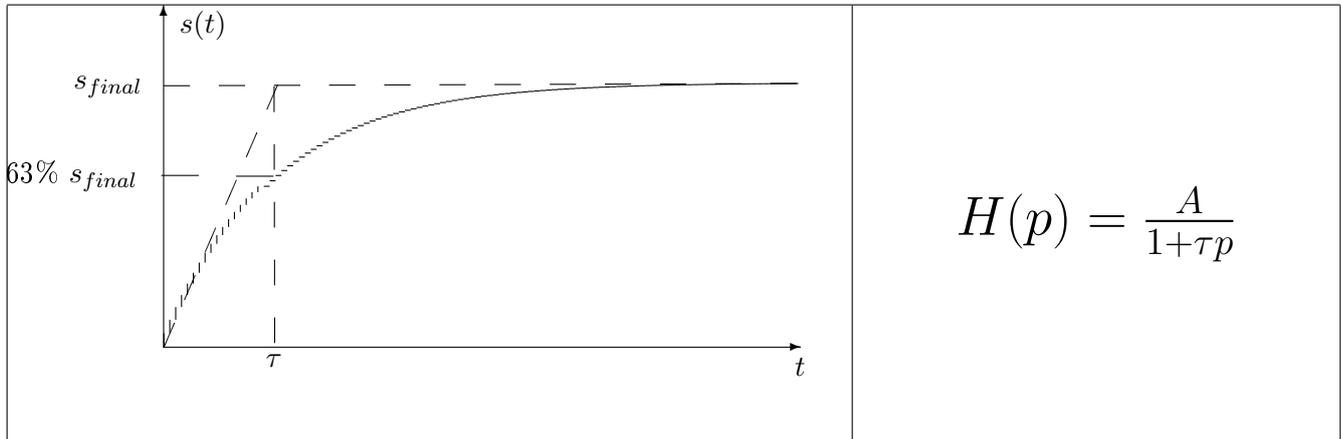
2.1.3 Réponse indicielle

Le spectre est correct, la commande est facile à implanter car c'est une commande classique. C'est la méthode la plus utilisée.

2.2 Etude des réponses indicielles

2.2.1 Systèmes du premier ordre

TABLE 2.1 – Réponse d'un système du premier ordre à un échelon.

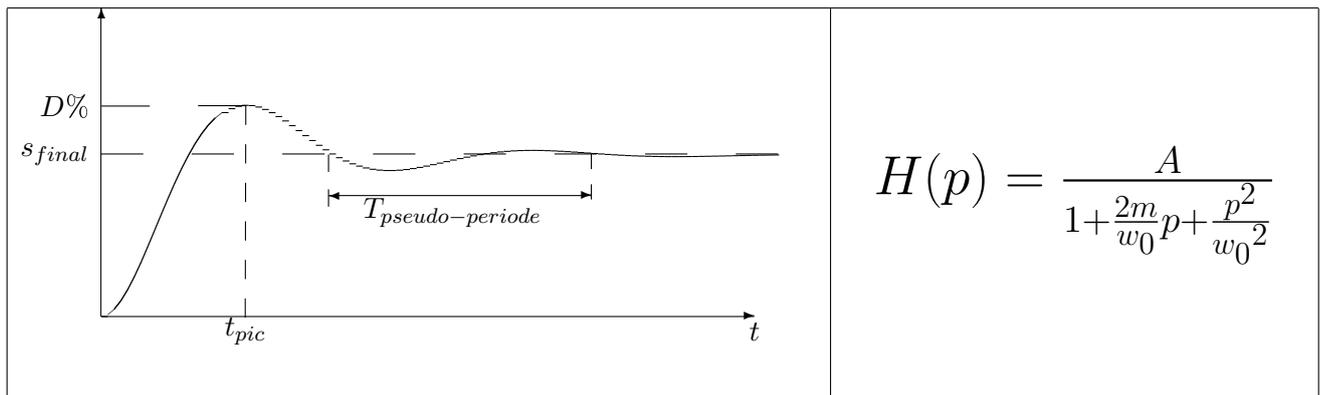


Pas de retard pur, démarrage franc (à comparer aux systèmes d'ordre 2 ou supérieur).

En cas de bruit, la méthode de mesure de τ via l'intersection de la pente de la tangente au démarrage avec la valeur finale est meilleure que le relevé du temps à 63% de la valeur finale.

2.2.2 Systèmes du second ordre résonnant

TABLE 2.2 – Réponse d'un système du deuxième ordre à un échelon.



Le facteur d'amortissement m se détermine à l'aide de la mesure du dépassement :

$$D\% = 100e^{\frac{-\pi m}{\sqrt{1-m^2}}}$$

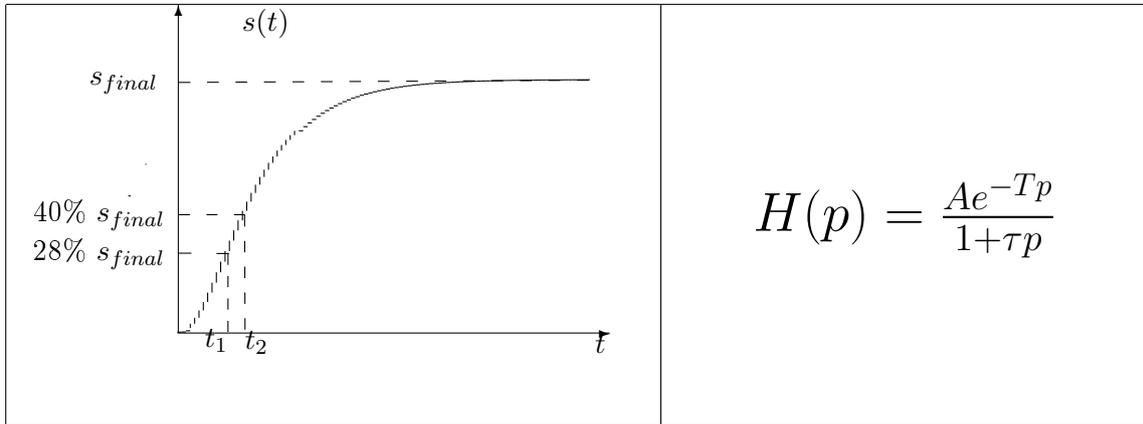
La pulsation propre w_0 s'obtient via l'une des deux formules :

$$t_{pic} = \frac{\pi}{w_0\sqrt{1-m^2}} \quad \text{ou} \quad T_{pseudo-période} = \frac{2\pi}{w_0\sqrt{1-m^2}}$$

2.2.3 Systèmes du premier ordre retardés ou du second ordre non résonnant

La méthode de Broïda consiste à "faire coller" un modèle de la forme $H(p) = \frac{Ae^{-Tp}}{1+\tau p}$ sur la réponse du système.

TABLE 2.3 – Réponse d'un système du deuxième ordre non résonnant à un échelon.



Les valeurs de T et de τ sont calculées à partir des relations suivantes :

$$\tau = 5.5 \times (t_2 - t_1)$$

$$T = 2.8 \times t_1 - 1.8 \times t_2$$

En pratique, la prise en compte du retard pur est difficile dans une synthèse de correcteur. On utilise donc l'approximation suivante :

$$H(p) = \frac{Ae^{-Tp}}{1 + \tau p} \simeq \frac{A(1 - Tp)}{1 + \tau p} \simeq \frac{A}{(1 + \tau p)(1 + Tp)}$$

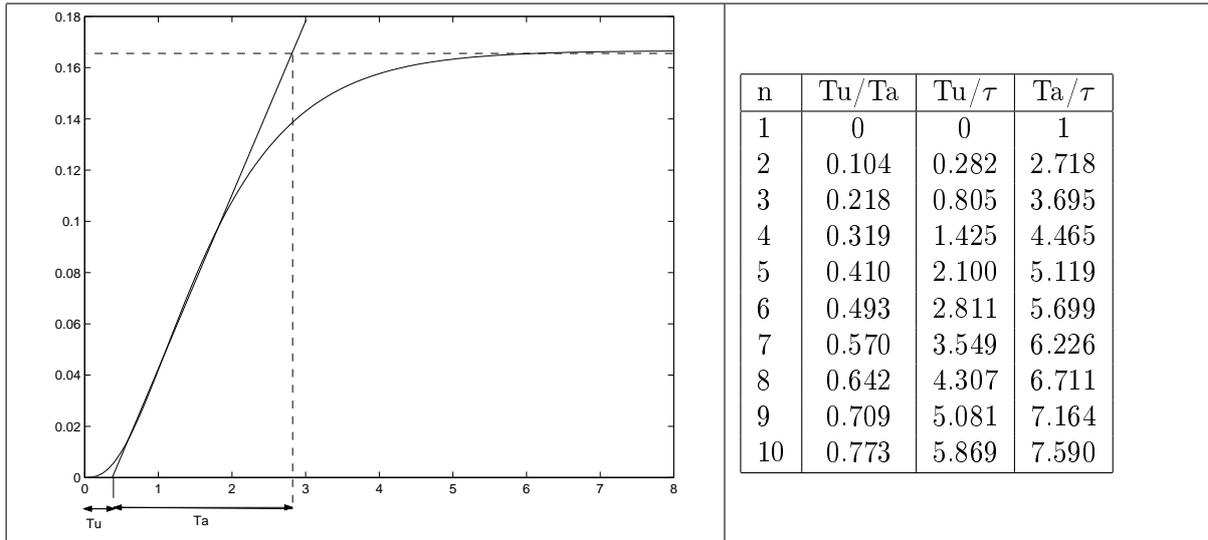
Cette approximation reste valable jusqu'à la pulsation $\omega = 1/T$.

2.2.4 Systèmes d'ordre supérieur à 2 non résonnants

La méthode la plus connue est la méthode de Strejc. Le modèle est :

$$H(p) = \frac{Ae^{-Tp}}{(1 + \tau p)^n}$$

TABLE 2.4 – Méthode de Strejc : Réponse indicielle.



1. Tracer la plus grande pente puis mesurer Tu et Ta .
2. Calculer le ratio $\frac{Tu}{Ta}$.
3. Dans la colonne $\frac{Tu}{Ta}$ du tableau 2.4 trouver la valeur immédiatement inférieure à ce ratio.
4. Sur la ligne de ce ratio déterminer n .
5. Toujours à l'aide des valeurs numériques de cette ligne, calculer τ avec $\frac{Ta}{\tau}$.
6. Calculer la nouvelle valeur de Tu que l'on appellera Tu_{calc} avec $\frac{Tu}{\tau}$.
7. En déduire T avec $T = Tu - Tu_{calc}$.

Le tableau 2.4 vient du calcul de la réponse indicielle des fonctions de transfert de la forme :

$$s(t) = \mathcal{L}^{-1} \left(\frac{Ae^{-Tp}}{p(1 + \tau p)^n} \right)$$

Exemple : pour $n = 4$

$$s(t) = \mathcal{L}^{-1} \left(\frac{1}{p(1 + \tau p)^4} \right) = 1 - e^{-\frac{t}{\tau}} - \frac{te^{-\frac{t}{\tau}}}{\tau} - \frac{1}{2} \frac{t^2 e^{-\frac{t}{\tau}}}{\tau^2} - \frac{1}{6} \frac{t^3 e^{-\frac{t}{\tau}}}{\tau^3}$$

la dérivée seconde s'écrit :

$$\frac{d^2 s(t)}{dt^2} = -\frac{1}{2} \frac{t^2 e^{-\frac{t}{\tau}}}{\tau^4} - \frac{1}{6} \frac{t^3 e^{-\frac{t}{\tau}}}{\tau^5}$$

qui s'annule pour $t = 3\tau$.

la tangente de plus grande pente est donc la tangente à la courbe en $t = 3\tau$. Le calcul de la droite tangente donne :

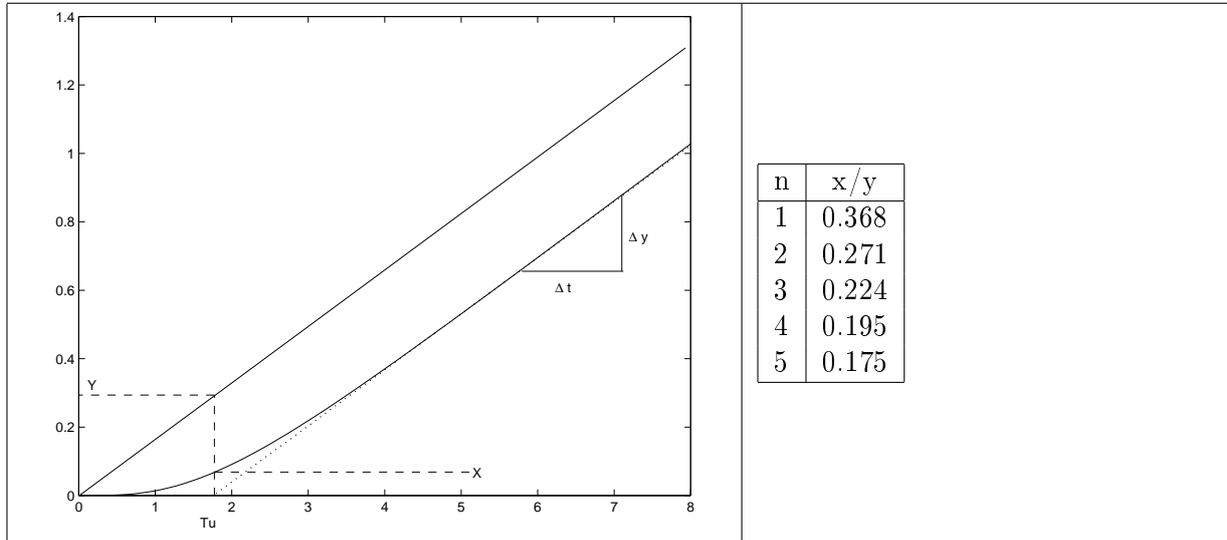
$$\Delta = 9/2 \frac{e^{-3}}{\tau} t - \frac{53}{2} e^{-3} + 1$$

Cette droite Δ intercepte la droite d'équation $s = 1$ en $t = \frac{53}{9} \tau$ et l'axe des abscisses en $t = \frac{1}{9} \frac{\tau(53e^{-3}-2)}{e^{-3}}$ donc $Ta = \frac{53}{9} \tau - \frac{1}{9} \frac{\tau(53e^{-3}-2)}{e^{-3}}$ et $Tu/Ta = 0.3193573120$

Note : si $Tu/Ta < 0.1$, donc $n = 1$, on applique plutôt la méthode de Broïda.

2.2.5 Système avec intégrateur

TABLE 2.5 – Méthode de Strejc (système avec intégrateur)



Identifié sous la forme :

$$H(p) = \frac{k_u}{p} \times \frac{1}{(1 + \tau p)^n}$$

1. Avec x/y : déterminer n
2. Calcul de τ avec $\tau = \frac{T_u}{n}$
3. Calcul de $k_u = \frac{\Delta y}{\Delta u} \times \frac{1}{\Delta t}$

En pratique il est assez rare d'identifier un système intégrateur en boucle ouverte car le système dérive rapidement en dehors de son domaine de linéarité (saturation, ...). En pratique, on procède à une identification en boucle fermée.

2.3 Autres modèles ?

Les méthodes précédentes sont très performantes si toutefois le comportement du système est proche des réponses indicielles attendues.

Qu'en est-il pour les systèmes décrits par les fonctions de transfert suivantes :

$$H_1(p) = \frac{Ae^{-Tp}}{1 + \frac{2z}{w_0}p + \frac{p^2}{w_0^2}} \quad \text{et} \quad H_2(p) = \frac{1 - Tp}{(1 + \tau_1 p)(1 + \tau_2 p)(1 + \tau - 2p)}$$

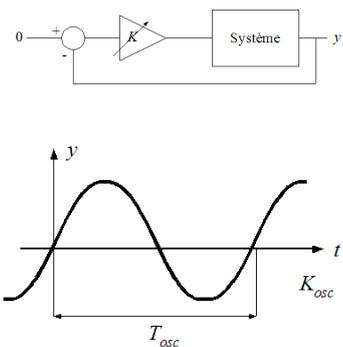
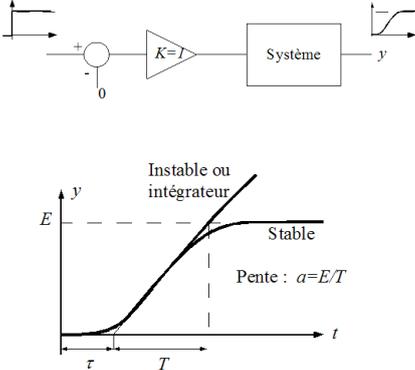
Dans la plupart des cas, l'un des modèles précédents sera suffisant. N'oubliez pas que nous cherchons un modèle de comportement uniquement. Dans les quelques cas résistant aux méthodes précédentes, un peu d'astuce permet de trouver une méthode pour déterminer les constantes du système. Et si rien de ce qui précède ne vous aide, il reste les méthodes de modélisation paramétrique.

2.4 Méthode de Ziegler-Nichols

Tous comptes fait, le but principal de l'identification en automatique est la mise en place d'un correcteur ou d'un régulateur afin d'améliorer les performances du système en boucle fermée. Aussi l'on est en droit de se demander si cette étape d'identification est bien nécessaire...

C'est exactement le principe de la méthode de Ziegler-Nichols, qui n'est pas une méthode d'identification à proprement parler mais une méthode de réglage fondée sur des mesures effectuées directement sur le système. Le tableau 2.6 résume la méthode.

TABLE 2.6 – Coefficients d'un PID réglé par les méthodes de Ziegler-Nichols et Chien-Hrones-Reswick : essai indiciel et méthode du pompage.

	Méthode de pompage Boucle fermée	Méthodes aperiodiques Boucle ouverte		
	Systèmes stables ou instables en boucle ouverte	Systèmes stables, instables ou intégrateurs		
				
	Ziegler-Nichols	Ziegler-Nichols	Chien-Hrones-Reswick	
	Régulation ou Poursuite	Régulation ou Poursuite	Régulation	Poursuite
P	$K = 0.5K_{osc}$	$K = \frac{T}{\tau}$	$K = 0.3\frac{T}{\tau}$	$K = 0.3\frac{T}{\tau}$
P.I	$K = 0.45K_{osc}$ $T_i = 0.83T_{osc}$	$K = 0.9\frac{T}{\tau}$ $T_i = 3.3\tau$	$K = 0.6\frac{T}{\tau}$ $T_i = 4\tau$	$K = 0.35\frac{T}{\tau}$ $T_i = 1.2T$
P.I.D	$K = 0.6K_{osc}$ $T_i = 0.5T_{osc}$ $T_d = 0.125T_{osc}$	$K = 1.2\frac{T}{\tau}$ $T_i = 2\tau$ $T_d = 0.5\tau$	$K = 0.95\frac{T}{\tau}$ $T_i = 2.4\tau$ $T_d = 0.4\tau$	$K = 0.6\frac{T}{\tau}$ $T_i = T$ $T_d = 0.5\tau$

Chapitre 3

Algorithme général d'identification

Il faut répondre aux questions suivantes :

- Quel genre de modèle ?
- Paramétrique ou non paramétrique ?
- Quelles mesures effectuer ?
- Quel signal d'excitation ?
- Quelle structure adopter ?
- Doit-on y inclure des connaissances a priori ?
- Comment choisir le bon modèle parmi tous ceux calculés ?
- Ce modèle est-il adapté à ce que l'on va lui demander ?

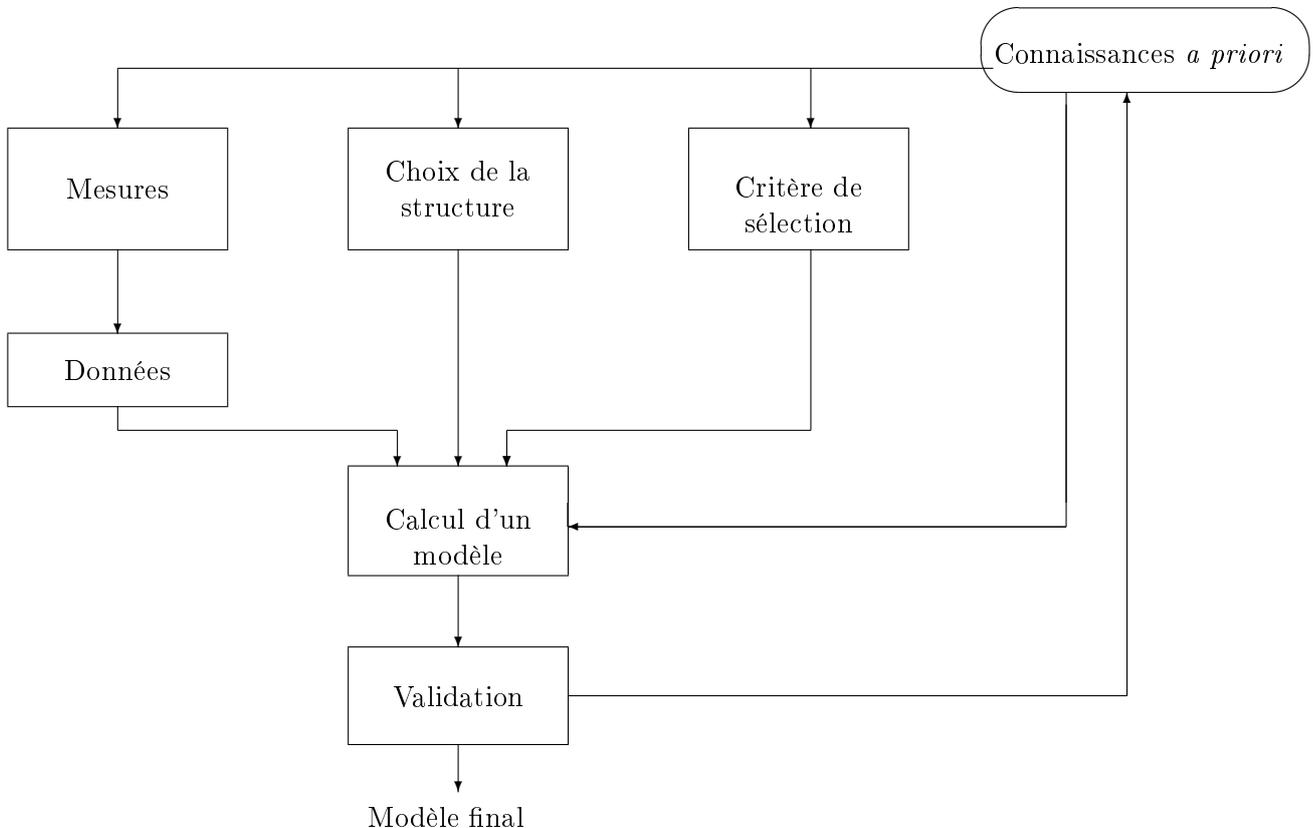


FIGURE 3.1 – Algorithme général d'identification des systèmes. Notez que le processus d'identification est un processus itératif, chaque passage apportant un peu plus de connaissances.

3.1 Choix du signal d'excitation

Pour bien identifier, il faut bien exciter dans tout le spectre de fréquences susceptible de contenir des constantes de temps du système.

- $\sin(\omega t)$: parfait d'un point de vue spectre (balayage en fréquence) mais peu de systèmes acceptent ce genre d'entrées.
- $\delta(t)$: parfait du point de vue théorique, mais difficile de réaliser une bonne approximation de l'impulsion.
- $u(t)$: moins bon d'un point de vue spectral ($u(f) = \frac{1}{2}\delta(f) + \frac{1}{j2\pi f}$), mais facile à implanter.
- $b(t)$: bruit blanc idéal d'un point de vue spectral mais comment le réaliser ?

Il arrive que vous n'ayez aucune possibilité d'exciter le système (ex : machine en production), il faudra alors profiter des commandes "naturelles" du système comme signal d'entrée du système. Dans ce cas, le premier travail consiste à calculer le spectre du signal d'entrée (FFT par exemple). Il faudra vérifier *a posteriori* que les constantes de temps identifiées sont bien dans des domaines de fréquences dans lesquels le système a été excité.

3.2 Les séquences binaires pseudo aléatoires (SBPA)

L'un des moyens de réaliser un signal "aléatoire" est la mise en œuvre de Séquences Binaires Pseudo Aléatoires (en anglais PRBS : Pseudo Random Binary Sequence)

Principe :

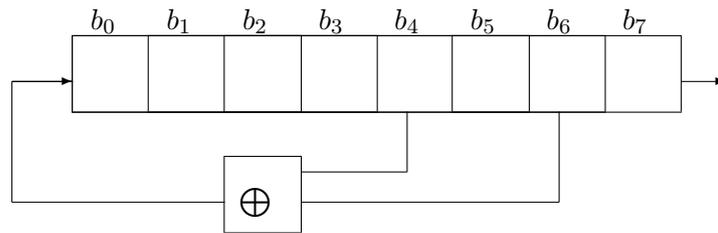


FIGURE 3.2 – Principe de génération de nombre aléatoires fondé sur un registre à décalage.

Pour que le système fonctionne, on doit initialiser le registre à n'importe quelle valeur binaire sauf zéro. le nombre codé en binaire $A = b_7b_6b_5b_4b_3b_2b_1b_0$ est "aléatoire". de la même façon, le bit b_7 semble indépendant de ses valeurs précédentes.¹

TABLE 3.1 – Tableau des bits à utiliser pour obtenir une séquence de longueur maximale

n	bits	n	bits
3	$b_2 \oplus b_1$	12	$b_{11} \oplus b_7 \oplus b_3 \oplus b_0$
4	$b_3 \oplus b_2$	14	$b_{13} \oplus b_9 \oplus b_5 \oplus b_0$
5	$b_4 \oplus b_2$	16	$b_{15} \oplus b_{11} \oplus b_2 \oplus b_0$
6	$b_5 \oplus b_4$	18	$b_{17} \oplus b_1$
7	$b_6 \oplus b_5$	20	$b_{19} \oplus b_2$
8	$b_7 \oplus b_3 \oplus b_2 \oplus b_1$	22	$b_{21} \oplus b_0$
9	$b_8 \oplus b_4$	24	$b_{23} \oplus b_6 \oplus b_1 \oplus b_0$
10	$b_9 \oplus b_6$	26	$b_{25} \oplus b_5 \oplus b_1 \oplus b_0$

1. Cela ne semble pas très sérieux, mais sachez que les générateurs de bruit en électronique utilisent ce principe mais sur 40 bits et que le cryptage de textes utilise un principe tout à fait similaire sur 128 bits!

Une séquence sur N bits à une longueur de $2^N - 1 = L$ dont 2^{N-1} "1" et $2^{N-1} - 1$ "0", sa valeur moyenne est donc non nulle.

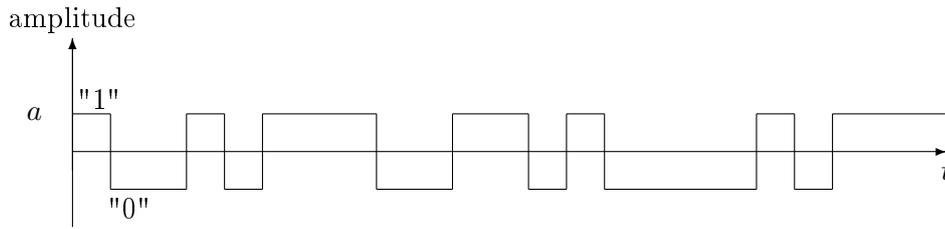


FIGURE 3.3 – diagramme temporel d'une séquence binaire pseudo aléatoire.

$$E[s(t)] = \frac{a}{L}$$

Pour déterminer la forme du spectre il faut en premier lieu calculer sa fonction d'auto-corrélation, le spectre étant la transformée de Fourier de la fonction d'auto-corrélation.

Fonction d'auto-corrélation

$$C_{xx}(\tau) = \frac{1}{LT} \int_0^{LT} x(\tau)x(t - \tau)dt$$

si $|\tau| \leq T$

$$C_{xx}(\tau) = a^2 \left(1 - \frac{L+1}{L} \cdot \frac{|\tau|}{T}\right)$$

si $|\tau| \geq T$

$$C_{xx}(\tau) = -\frac{a^2}{L}$$



FIGURE 3.4 – Fonction d'auto-corrélation d'une SBPA.

On en déduit le spectre qui est la transformée de Fourier de la fonction d'auto-corrélation.

$$P(f) = a^2 \frac{L+1}{L^2} \sum_{-\infty}^{+\infty} \delta(v - \frac{n}{LT}) \left(\frac{\sin \pi \frac{n}{L}}{\pi \frac{n}{L}}\right)^2 - \frac{a^2}{L} \delta(v)$$

Le spectre de la SBPA représenté en figure 3.5 est donc un peigne de Dirac modulé par un sinus cardinal. On considère que le bruit peut être considéré comme un bruit blanc dans le premier tiers du premier lobe ($\pm 1/3T_e$). On remarque donc que :

- plus L grand \Rightarrow plus la valeur moyenne est faible
- plus L grand \Rightarrow plus de raies, plus proche du bruit blanc

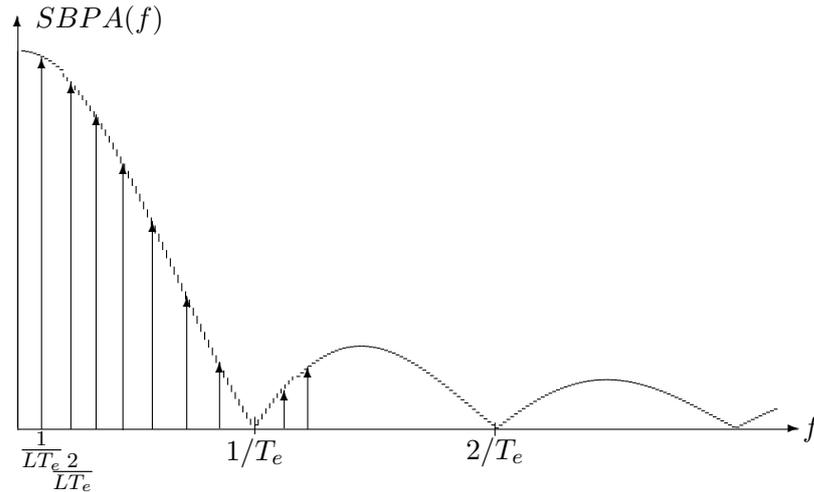


FIGURE 3.5 – Spectre d'une SBPA.

3.2.1 Choix des paramètres d'une SBPA

Il faut donc déterminer a , L et T_e "optimum" vis-à-vis du système à identifier. Le premier compromis concerne L et T_e .

Plus la séquence est longue :

- plus il y a d'informations
- plus le transitoire est négligeable
- plus les organes souffrent
- plus les dérives s'accroissent

Le deuxième compromis est : plus a est petit,

- plus le signal est noyé dans le bruit (le vrai celui là !)
- moins les organes souffrent
- moins les dérives s'accroissent

Le choix de a est donc avant tout lié au bruit présent dans le système. On choisira a juste assez grand pour avoir du signal en sortie du système.

Choix de L et T_e Une SBPA de longueur $L = 2^N - 1$ envoyée à la fréquence $1/T_e$ présente un plateau de longueur maximale de longueur $N \times T_e$. Le plus petit plateau étant de longueur T_e

Le choix de N (donc L) et T_e est donc, encore une fois, un compromis entre :

- une bonne identification du gain statique
- une bonne excitation sur la bande de fréquence du système

Soit τ_{max} la plus grande constante de temps du système et τ_{min} la plus petite constante de temps du système.

Pour une bonne identification du gain statique on choisira :

$$NT_e = 3 \text{ à } 5 \tau_{max}$$

Pour une bonne excitation sur le spectre entre 0 et la plus haute fréquence de coupure on choisira :

$$0,3 \frac{1}{T_e} = \frac{1}{2\pi\tau_{min}}$$

La résolution de ses deux équations donne N et T_e .

Note 1 : si $\tau_{max} \gg \tau_{min}$ N devient vite très grand donc LT_e devient prohibitif !

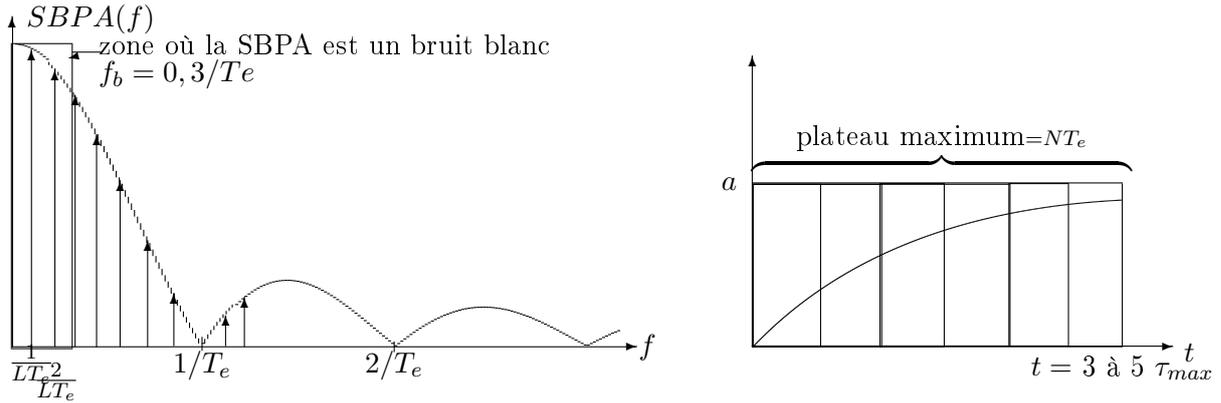


FIGURE 3.6 – Paramètres d'une SBPA.

Note 2 : Si $f_e = 1/T_e$ est bloquée et trop grande (beaucoup de systèmes sont sur-échantillonnés), il est parfaitement possible choisir $T_e = n/f_e$ en répétant chaque bit n fois.

Note 3 : Si f_e est bloquée et trop petite, abandonnez l'idée d'identifier la petite constante de temps : de toutes façons vous ne pourrez pas la contrôler !

3.3 Identification basée sur l'erreur de sortie

Le principe de cette méthode d'identification est illustré en figure 3.7. Le modèle est une fonction de n paramètres θ_i , i variant de 1 à n . Il s'agit alors de déterminer les paramètres θ_i tels que le critère soit minimum. Le critère est en général choisi de la forme $J = \sum \varepsilon^2$.

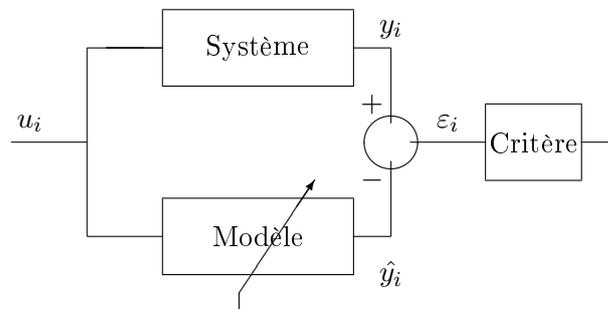


FIGURE 3.7 – Principe d'identification fondé sur l'erreur de sortie.

La recherche des paramètres optimaux $\hat{\theta}_i$ se fait par programmation non linéaire. IL s'agit d'utiliser un algorithme qui à partir de paramètres non optimaux θ_i et un critère J donne les paramètres $\hat{\theta}_i$. Ces algorithmes sont nombreux, à titre de d'exemple voici les plus utilisés.

- gradient
- quasi Newton
- Nelder et Mead
- algorithmes génétiques
- recuit simulé

Avantages de cette méthode :

- Pas d'hypothèse sur la forme du modèle, il peut en particulier être non linéaire.

– Adapté à la recherche de paramètres physiques si modèle de connaissances (modèle continu !).
L'inconvénient majeur est qu'aucun de ces algorithmes n'est capable de garantir que le résultat est réellement l'optimum.

3.4 Identification basée sur l'erreur de prédiction

Dans cette partie, nous supposons que le modèle obtenu est un prédicteur, c'est à dire qu'il permet de calculer la sortie à l'instant i en fonction des entrées et des sorties réelles aux instants précédents u_{i-k} et y_{i-k} .

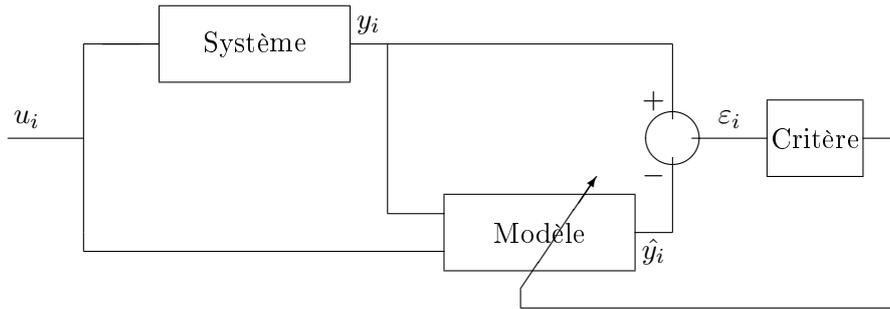


FIGURE 3.8 – Principe d'identification fondée sur l'erreur de prédiction.

3.4.1 Méthode des moindres carrés simples

Les calculs suivants seront fondés sur le modèle présenté en figure 3.9.

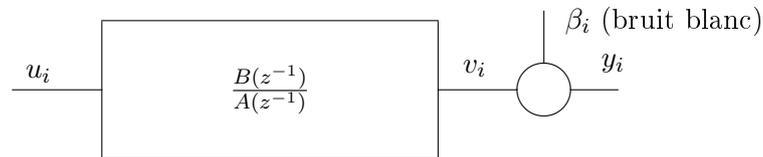


FIGURE 3.9 – Modèle du système étudié.

La mise en équation du système donne :

$$\begin{aligned} Av_i &= Bu_i \\ y_i &= v_i + \beta_i \\ A(y_i - \beta_i) &= Bu_i \\ Ay_i &= Bu_i + A\beta_i \end{aligned}$$

posons :

$$e_i = A\beta_i$$

e_i sont les *résidus* de l'estimation. On obtient finalement :

$$Ay_i = Bu_i + e_i \tag{3.1}$$

Si nous possédons N mesures consécutives, on peut écrire $N - n$ fois² l'équation 3.1.

2. n est l'ordre, supposé connu du polynôme A , on appellera p l'ordre du polynôme B .

Sous forme matricielle, on obtient :

$$\begin{bmatrix} y_N \\ y_{N-1} \\ y_{N-2} \\ y_{N-3} \\ \vdots \\ y_{n+2} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} -y_{N-1} & -y_{N-2} & \cdots & -y_{N-n} & u_N & \cdots & u_{N-p} \\ -y_{N-2} & -y_{N-3} & & -y_{N-n-1} & u_{N-1} & & u_{N-p-1} \\ -y_{N-3} & \vdots & & \vdots & u_{N-2} & & u_{N-p-2} \\ \vdots & -y_{n+1} & & \vdots & \vdots & & u_{N-p-3} \\ -y_{n+1} & -y_n & & \vdots & \vdots & & u_{N-p} \\ -y_n & -y_{n-1} & \cdots & -y_1 & u_{n+1} & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix} + \begin{bmatrix} e_N \\ e_{N-1} \\ e_{N-2} \\ \vdots \\ \vdots \\ e_{n+2} \\ e_{n+1} \end{bmatrix}$$

soit :

$$y = X\theta + e$$

le critère J est :

$$J = \sum e^2 = e^T e$$

donc :

$$J = (y - X\theta)^T (y - X\theta) = y^T y - \theta^T X^T y - y^T X\theta + \theta^T X^T X\theta$$

Nous cherchons la valeur $\hat{\theta}$ de θ qui minimise J . Ainsi

$$\left. \frac{\partial J}{\partial \theta} \right|_{\theta=\hat{\theta}} = 0 = -2X^T y + 2X^T X\theta \Big|_{\theta=\hat{\theta}}$$

D'où on en déduit :

$$\hat{\theta} = (X^T X)^{-1} X^T y$$

Il reste à vérifier que la valeur obtenue est bien un minimum

$$\left. \frac{\partial^2 J}{\partial \theta^2} \right| = 2X^T X$$

$2X^T X$ est bien une matrice définie positive donc c'est bien un minimum !

3.4.2 Calcul du biais de l'estimateur

$$\begin{aligned} E[\hat{\theta}] &= E[(X^T X)^{-1} X^T y] \quad \text{et } y = X\theta + e \\ &= E[\theta + (X^T X)^{-1} X^T e] \\ &= \theta + E[(X^T X)^{-1} X^T e] \end{aligned}$$

L'estimateur est non biaisé si :

$$E[\hat{\theta}] = \theta$$

donc si :

$$E[(X^T X)^{-1} X^T e] = 0$$

il faut donc que :

- X et e soient non corrélés

- e soit centré

Mais ce n'est pas le cas ! Le calcul de ce biais sur l'exemple présenté en figure 3.10 permet de le montrer.

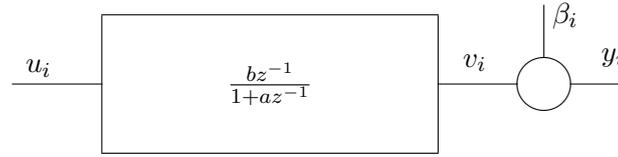


FIGURE 3.10 – Système du premier ordre bruité.

Les équations du système sont :

$$\begin{aligned} v_i + av_{i-1} &= bu_{i-1} \\ y_i &= v_i + \beta_i \end{aligned}$$

où β_i est un bruit répondant aux deux équations suivantes :

$$\begin{aligned} E[\beta_i] &= 0 \\ E[\beta_i \cdot \beta_{i-k}] &= \sigma^2 \delta(k) \end{aligned}$$

soit une moyenne nulle et une fonction d'auto-corrélation de type Dirac.

En réécrivant les équations du système on obtient :

$$\begin{aligned} y_i + ay_{i-1} &= bu_{i-1} + \beta_i + a\beta_{i-1} \\ &= bu_{i-1} + e_i \end{aligned}$$

$$E[e_i] = E[\beta_i + a\beta_{i-1}] = E[\beta_i] + aE[\beta_{i-1}] = 0$$

$$\begin{aligned} E[e_i \cdot e_{i-k}] &= E[(\beta_i + a\beta_{i-1})(\beta_{i-k} + a\beta_{i-k-1})] \\ &= E[\beta_i \cdot \beta_{i-k}] + aE[\beta_i \cdot \beta_{i-k-1}] + aE[\beta_{i-1} \cdot \beta_{i-k}] + a^2E[\beta_{i-1} \cdot \beta_{i-k-1}] \\ &= \sigma^2[(1 + a^2)\delta(k) + a\delta(k-1) + a\delta(k+1)] \end{aligned}$$

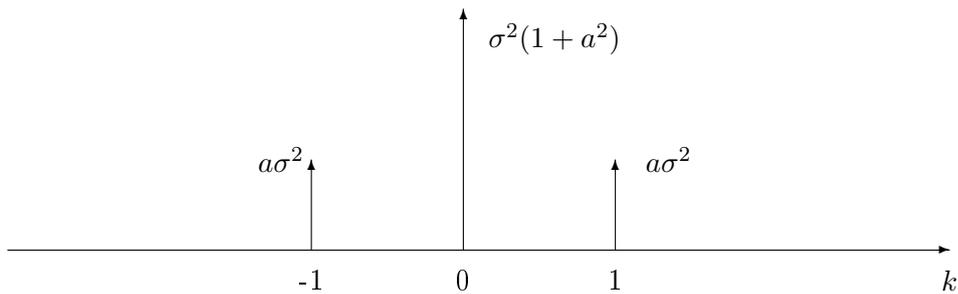


FIGURE 3.11 – Fonction d'auto-corrélation des résidus d'estimation.

La fonction d'auto-corrélation (voir fig. 3.11) montre bien que les résidus e_i ne sont pas un bruit blanc, notre estimateur est bien biaisé ! Il reste donc à trouver une méthode qui permette de rendre ce biais nul. Deux méthodes sont proposées ci-après : La méthode des moindres carrés généralisés et la méthode de la matrice instrumentale.

3.4.3 Méthode des moindres carrés généralisés

Le système étudié est décrit sur la figure 3.12.

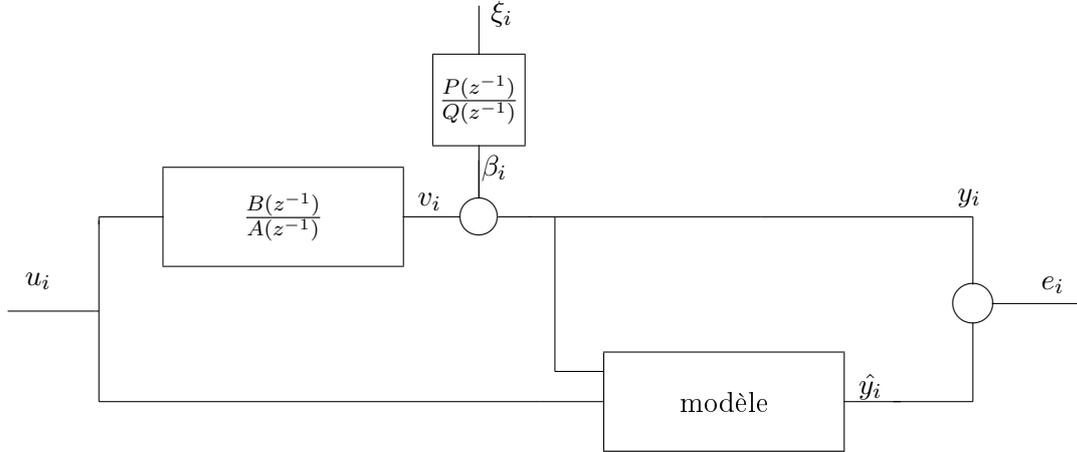


FIGURE 3.12 – Système étudié.

$$\begin{aligned}
 Av_i &= Bu_i \\
 y_i &= v_i + \beta_i \\
 A(y_i - \beta_i) &= Bu_i \\
 Ay_i &= Bu_i + A\beta_i
 \end{aligned} \tag{3.2}$$

les résidus sont donc :

$$e_i = A\beta_i$$

écrits en fonction du bruit blanc ξ_i

$$e_i = A \frac{P}{Q} \xi_i$$

équation que nous écrirons dans un souci de simplification d'écriture :

$$Fe_i = \xi_i$$

En reprenant l'équation 3.2 et en multipliant tous les termes par F on obtient :

$$\begin{aligned}
 AFy_i &= BFu_i + AF\beta_i \\
 &= BFu_i + A \frac{Q}{AP} \beta_i \\
 &= BFu_i + \xi_i
 \end{aligned}$$

En réécrivant cette dernière équation,

$$Ay_i^* = Bu_i^* + \xi_i$$

où y_i^* et u_i^* sont, respectivement les sorties y_i et les commandes u_i filtrées par le filtre F .

En appliquant la méthode des moindres carrés simples sur ces entrées-sorties filtrées le biais de l'estimateur sera nul car le bruit ξ_i est bien un bruit blanc. L'estimation des paramètres sera donc parfaite. Toutefois il reste à déterminer le filtre F .

Détermination de F Reprenons 3.4.3 sous forme matricielle (avec F de la forme : $F = 1 + f_1 z^{-1} + f_2 z^{-2} + \dots + f_n z^{-n}$)

$$\begin{bmatrix} e_N \\ e_{N-1} \\ e_{N-2} \\ e_{N-3} \\ \vdots \\ e_{n+2} \\ e_{n+1} \end{bmatrix} = \begin{bmatrix} -e_{N-1} & -e_{N-2} & \dots & -e_{N-n} \\ -e_{N-2} & -e_{N-3} & & -e_{N-n-1} \\ -e_{N-3} & \vdots & & \vdots \\ \vdots & -e_{n+1} & & \vdots \\ -e_{n+1} & -e_n & & \vdots \\ -e_n & -e_{n-1} & \dots & -e_1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_n \end{bmatrix} + \begin{bmatrix} \xi_N \\ \xi_{N-1} \\ \xi_{N-2} \\ \vdots \\ \vdots \\ \xi_n \end{bmatrix}$$

soit

$$e = X_e f + \xi$$

qui est une forme connue et dont on sais que la solution est :

$$\hat{f} = (X_e^T X_e)^{-1} X_e^T e$$

Les paramètres f de F sont donc déterminés à condition de disposer des résidus e . Mais c'est justement pour les avoir que l'on déroule toutes ces équations...

La solution proposée consiste alors à procéder par itérations en utilisant une première estimation des résidus e obtenue par un premier modèle utilisant ${}^1\theta$ une première approximation des paramètres du système obtenue par la méthode des moindres carrés simples.

L'algorithme est alors, à partir des entrées-sorties ${}^1u_i = u_i$ et ${}^1y_i = y_i$

1. Estimation des paramètres
 ${}^k\theta = ({}^k X^T X)^{-1} {}^k X^T y$
2. Estimation des résidus
 ${}^k e = {}^k y - {}^k X {}^k \theta$
3. Construction du filtre F_1
 ${}^k \hat{f} = ({}^k X_e^T X_e)^{-1} {}^k X_e^T e$
4. Filtrage les données
 ${}^k y_i^* = {}^k y_i$ et ${}^k u_i^* = {}^k F^k u_i$
5. Rebouclage en utilisant ce nouvel ensemble d'entrées sorties

Quelques remarques :

Si les filtres ${}^k F$ sont du premier ordre ($n = 1$), la matrice X_e se réduit à un réel ce qui simplifie la formule ${}^k \hat{f} = ({}^k X_e^T X_e)^{-1} {}^k X_e^T e = \frac{{}^k e}{{}^k X_e}$.

Critère d'arrêt de l'algorithme : Le premier réflexe consiste à calculer la "blancheur" des résidus, mais ce critère est assez difficile à mettre en œuvre. Une méthode plus simple consiste à calculer un critère d'arrêt de la forme :

$$C_a = \frac{\Delta J}{J} = \frac{{}^k J - {}^{k-1} J}{{}^k J} \quad \text{avec } J = e^T e$$

En pratique on utilisera $C_a \leq 5\%$.

A chaque itération le bruit "devient plus blanc" et pas conséquent le critère décroît.

3.4.4 Méthode de la Matrice Instrumentale

Puisque le biais de l'estimateur des moindres carrés est biaisé à cause de la corrélation entre X et e , on se propose de déterminer une autre matrice qui permette le calcul de θ tout en évitant cette corrélation et donc le biais de l'estimateur.

Posons :

$$\hat{\theta} = (Z^T X)^{-1} Z^T y$$

où Z est la matrice instrumentale.

Quelles sont les conditions sur Z pour que l'équation précédente ait un sens ?

En poursuivant le calcul :

$$\hat{\theta} = (Z^T X)^{-1} Z^T (X\theta + e)$$

$$\hat{\theta} = \theta + (Z^T X)^{-1} Z^T e$$

Pour que

$$E[\hat{\theta}] = \theta$$

il faut :

$$E[Z^T X] \neq 0 \quad \text{et} \quad E[Z^T e] = 0$$

Plusieurs choix de Z sont possibles, en voici deux :

le premier choix consiste à faire deux campagnes de mesures avec la même entrée u_i , le plus souvent consécutives (décalées dans le temps). Vous avez alors la possibilité de déterminer deux matrices X soit X_1 et X_2 , puis déterminer $\hat{\theta}$ en posant :

$$\hat{\theta} = (X_2^T X_1)^{-1} X_2^T y_1$$

Il n'y a plus de corrélation entre X_2 et y_1 (le bruit est stochastique et ergodique) donc le biais est bien nul.

La deuxième proposition consiste à créer les données nécessaires à la création de la matrice X_2 à l'aide d'un modèle de type moindre carrés simples issu de la première campagne de mesure.

3.5 Identification en boucle fermée

Ce domaine de l'identification est actuellement très actif. En effet ces modèles sont nécessaires dans le cadre de la commande prédictive. Plus pragmatiquement, rappelons que le plus souvent l'identification en milieu industriel s'effectue sur une machine en production. Il est par conséquent difficile voire impossible d'ouvrir la boucle pour procéder à une identification. C'est tout particulièrement le cas des systèmes instables.

Il reste néanmoins possible d'obtenir un modèle. Le système de la figure 3.13 représente le système à identifier en boucle fermée. le correcteur du système est représenté par la fraction rationnelle $\frac{C(z^{-1})}{D(z^{-1})}$. Trois approches sont alors possibles.

La méthode directe : Dans ce cas on utilise les entrées v_i et les sorties y_i . L'identification se fait alors comme en boucle ouverte. L'avantage de cette approche est de ne pas nécessiter la connaissance du régulateur. Les méthodes ARX, ARMAX et les modèles d'état donnent de bons résultats. Les meilleurs résultats seront obtenus si :

- Le modèle de bruit est bon
- la boucle de retour n'affecte pas ou peu le spectre du signal d'entrée (v_i)
- le rapport signal à bruit est important (peu de bruit)

Dans le cas contraire, l'estimateur est biaisé.

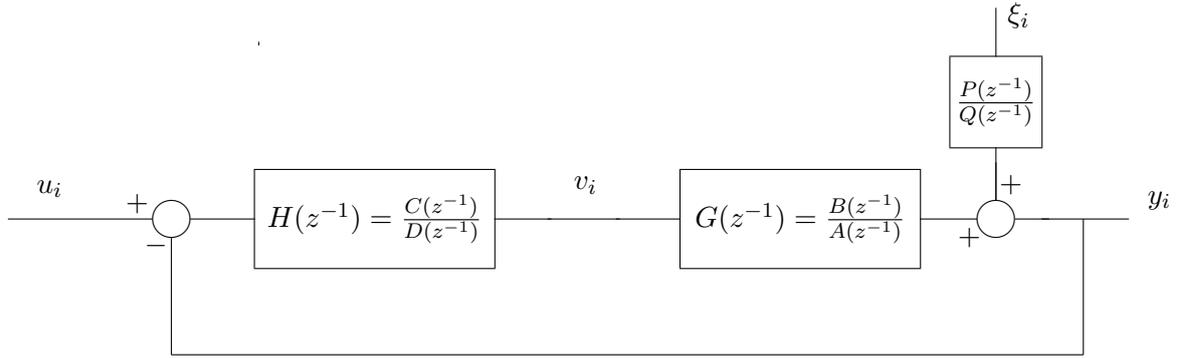


FIGURE 3.13 – Identification des systèmes en boucle fermée.

La méthode indirecte : Dans ce cas on utilise les entrées u_i et les sorties y_i . Bien entendu on identifie alors le système en boucle fermée soit :

$$M(z^{-1}) = \frac{B'(z^{-1})}{A'(z^{-1})} = \frac{H(z^{-1})G(z^{-1})}{1 + H(z^{-1})G(z^{-1})} = \frac{C(z^{-1})B(z^{-1})}{C(z^{-1})B(z^{-1}) + D(z^{-1})A(z^{-1})}$$

La connaissance du régulateur nous permet de déterminer le modèle du système :

$$G(z^{-1}) = \frac{M(z^{-1})}{H(z^{-1}) - M(z^{-1})H(z^{-1})}$$

L'avantage de cette méthode est que n'importe quelle méthode d'identification est applicable et donnera un modèle du système en boucle fermée. Par contre, la moindre erreur sur le régulateur (pas forcément bien connu, paramètres, saturations ...) se retrouve dans le modèle.

La méthode "Joint Input-Output" : Avec cette méthode on se propose de déterminer simultanément les deux modèles :

- le modèle du système
- le modèle du correcteur.

Ceci s'effectue facilement en utilisant les modèles paramétriques d'état sous Matlab.

$$\begin{bmatrix} y \\ v \end{bmatrix} = Au + B\xi$$

La matrice A contient alors les dynamiques du correcteur G_{uv} et celle du système en boucle fermée G_{uy} . Comme pour la méthode précédente on remonte au modèle du système par calcul.

3.6 Et Matlab[®] dans tout ça ?

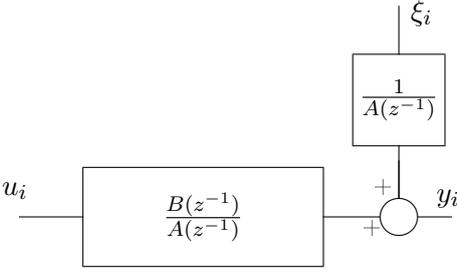
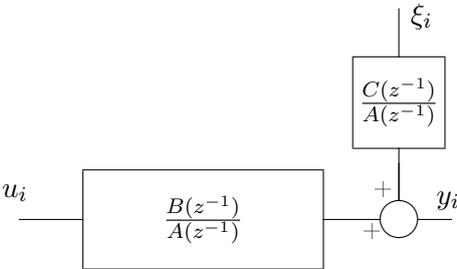
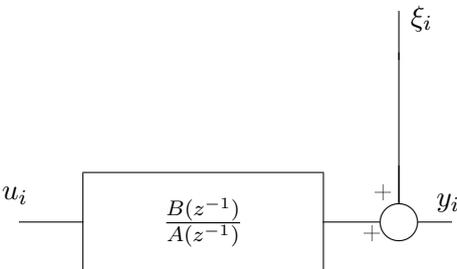
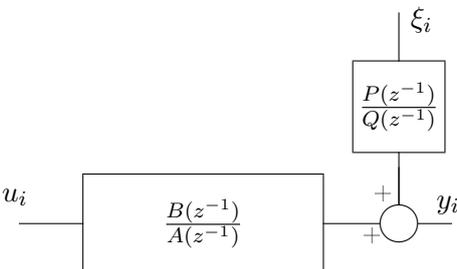
Matlab est avant tout un outil de calcul ! Scilab, gratuit, présente les mêmes fonctionnalités, il est vrai, sans les boîtes à outils du moins pas aussi complètes.

Matlab propose donc toute une série d'outils d'aide à l'identification. En particulier, quatre modèles linéaires, les plus couramment utilisés sont des fonctions de Matlab.

Dans le tableau 3.2 ces modèles sont présentés avec leurs utilisations "classique" et la méthode de détermination des paramètres. Notez qu'une lecture de la documentation concernant la boîte à outils "identification" est une excellente activité !

Face à la facilité de calcul de modèles, le problème n'est plus vraiment de choisir une structure adaptée mais faire le tri parmi les quantités de modèles calculés.

TABLE 3.2 – Modèles paramétriques implantés sous Matlab.

	<p>Modèle ARX Le modèle le plus simple, donne souvent de bons résultats. le seul hic est le traitement du bruit qui est soumis à la même dynamique que l'entrée. A utiliser en première approximation ou lorsque le bruit est surtout à l'entrée.</p> <p>Méthode de détermination des paramètres : Moindres carrés ou Matrice instrumentale</p>
	<p>Modèle ARMAX Proche du modèle ARX, il s'utilise dans les mêmes cas. Il permet en outre de créer un modèle de bruit un peu plus réaliste. C'est le modèle le plus utilisé.</p> <p>Méthode de détermination des paramètres : Maximum de vraisemblance</p>
	<p>Modèle OE Bien que semblant plus simple que les précédents, le calcul des paramètres s'avère plus difficile. Parfait lorsque le bruit est surtout un bruit de capteur, donc proche de la sortie.</p> <p>Méthode de détermination des paramètres : Maximum de vraisemblance</p>
	<p>Modèle de Box-Jenkins Le modèle complet par excellence, la dynamique différente pour l'entrée et le bruit en font un bon modèle.</p> <p>Méthode de détermination des paramètres : Maximum de vraisemblance</p>

Chapitre 4

Estimations récursives

4.1 Moindres carrés récursifs

L'estimation de paramètres par la méthode des moindres carrés simples présente un inconvénient majeur, la nécessité de calculer l'inverse d'une matrice, ce qui est long et parfois impossible sur un microcontrôleur. On se propose à travers cet exercice de déterminer une forme récursive de cette estimation.

Les avantages d'une formulation récursive tiennent essentiellement en deux points.

- La possibilité de traiter un plus grand nombre de données que dans le cas de la formulation directe (pas de pseudo-inverse à calculer), notamment dans le cas de l'implantation sur un microcontrôleur.
- Dans le cas des systèmes variants dans le temps, la forme récursive permet de "suivre" les paramètres du système. Dans ce cas, l'identification en ligne est généralement suivie d'une commande qui elle aussi "s'adapte" aux paramètres courants, on entre alors dans le domaine des commandes auto-adaptatives qui dépassent le cadre de ce cours.

Comme dans tout problème récursif on s'intéresse d'abord à la boucle, ensuite comment on en sort et enfin comment on y entre. Supposons que nous possédions une estimation des paramètres à l'instant N : θ_N

$$\theta_N = (X_N^T X_N)^{-1} X_N^T Y_N$$

à l'instant suivant la nouvelle estimation est :

$$\theta_{N+1} = (X_{N+1}^T X_{N+1})^{-1} X_{N+1}^T Y_{N+1}$$

avec :

$$X_{N+1} = \begin{bmatrix} x_{N+1} \\ X_N \end{bmatrix}, \quad Y_{N+1} = \begin{bmatrix} y_{N+1} \\ Y_N \end{bmatrix} \quad \text{et} \quad E_{N+1} = \begin{bmatrix} e_{N+1} \\ E_N \end{bmatrix}$$

où $x_{N+1} = [-y_N - y_{N-1} \dots - y_{N-n+1} u_{N+1} \dots u_{N-p+1}]$
 θ_{N+1} s'écrit alors :

$$\begin{aligned} \hat{\theta}_{N+1} &= (X_{N+1}^T X_{N+1})^{-1} X_{N+1}^T Y_{N+1} \\ &= (X_N^T X_N + x_{N+1}^T x_{N+1})^{-1} (X_N^T Y_N + x_{N+1}^T y_{N+1}) \end{aligned}$$

En utilisant le lemme d'inversion matricielle :

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

on obtient :

$$\widehat{\theta}_{N+1} = \frac{\left((X_N^T X_N)^{-1} - (X_N^T X_N)^{-1} x_{N+1}^T [1 + x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T]^{-1} x_{N+1} (X_N^T X_N)^{-1} \right)}{(X_N^T Y_N + x_{N+1}^T y_{N+1})}$$

En posant $a = 1 + x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T$

$$\begin{aligned} \widehat{\theta}_{N+1} &= \frac{\left((X_N^T X_N)^{-1} - (X_N^T X_N)^{-1} x_{N+1}^T [1 + x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T]^{-1} x_{N+1} (X_N^T X_N)^{-1} \right)}{(X_N^T Y_N + x_{N+1}^T y_{N+1})} \\ \widehat{\theta}_{N+1} &= \underbrace{\left((X_N^T X_N)^{-1} - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} x_{N+1} (X_N^T X_N)^{-1} \right)}_{\widehat{\theta}_N} (X_N^T Y_N + x_{N+1}^T y_{N+1}) \\ &= \underbrace{(X_N^T X_N)^{-1} X_N^T Y_N}_{\widehat{\theta}_N} + (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} - \underbrace{(X_N^T X_N)^{-1} x_{N+1}^T a^{-1} x_{N+1} (X_N^T X_N)^{-1} X_N^T Y_N}_{\widehat{\theta}_N} \\ &\quad - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} \\ &= \widehat{\theta}_N + (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} x_{N+1} \widehat{\theta}_N \\ &\quad - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} \\ &= \widehat{\theta}_N - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(x_{N+1} \widehat{\theta}_N \right) \\ &\quad + (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(-x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} \right) + (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} \\ &= \widehat{\theta}_N - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(x_{N+1} \widehat{\theta}_N \right) \\ &\quad + (X_N^T X_N)^{-1} x_{N+1}^T \left(-a^{-1} x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} + y_{N+1} \right) \\ &= \widehat{\theta}_N - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(x_{N+1} \widehat{\theta}_N \right) \\ &\quad + (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(-x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T + a \right) y_{N+1} \\ &= \widehat{\theta}_N - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(x_{N+1} \widehat{\theta}_N \right) \\ &\quad + (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} y_{N+1} \\ &= \widehat{\theta}_N + (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(y_{N+1} - x_{N+1} \widehat{\theta}_N \right) \end{aligned}$$

On obtient finalement une formule de récurrence :

$$\widehat{\theta}_{N+1} = \widehat{\theta}_N + (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(y_{N+1} - x_{N+1} \widehat{\theta}_N \right)$$

Le terme $(y_{N+1} - x_{N+1} \widehat{\theta}_N)$ représente l'erreur d'estimation à l'aide des paramètres précédents. L'équation est alors : la nouvelle estimation est l'ancienne estimation corrigée par un terme proportionnel à l'erreur d'estimation précédente, que l'on peut récrire sous la forme.

$$\widehat{\theta}_{N+1} = \widehat{\theta}_N + K_{N+1} \left(y_{N+1} - x_{N+1} \widehat{\theta}_N \right)$$

L'algorithme général est donc :

$$\widehat{\theta}_{N+1} = \widehat{\theta}_N + K_{N+1} \left(y_{N+1} - x_{N+1} \widehat{\theta}_N \right) \quad (4.1)$$

$$K_{N+1} = P_N x_{N+1}^T (I + x_{N+1} P_N x_{N+1}^T)^{-1} \quad (4.2)$$

$$P_{N+1} = P_N - K_{N+1} x_{N+1} P_N \quad (4.3)$$

On remarquera une analogie avec le filtre de Kalman, en fait c'est bien l'équivalent d'un filtre de Kalman appliqué sur le système suivant :

$$\begin{aligned}\theta_{N+1} &= \theta_N \\ y_{N+1} &= x_{N+1}\theta_N + e_{N+1}\end{aligned}$$

La condition de convergence est la même : e_N doit être un bruit blanc.

L'initialisation de l'algorithme se fait de deux façons :

- Si on connaît une première estimation de θ (en provenance d'une méthode non récursive par exemple), on l'utilise. Dans ce cas on prend $P_0 = \lambda I$ avec λ petit (petite variance du bruit)
- Si on ne connaît pas de première approximation, on prend θ_0 quelconque et $P_0 = \lambda I$ avec λ grand (grande variance du bruit).

Cet algorithme ne peut être utilisé que si les paramètres du système sont constants, en effet lorsque N tend vers l'infini, P_N tend vers 0, alors une variation même importante des paramètres n'influe plus sur l'estimation de θ . Notons encore que sur le plan numérique la forme $P_{N+1} = P_N - K_{N+1}x_{N+1}P_N$ est très mal conditionnée, on lui préfère la forme :

$$P_{N+1} = (I - K_{N+1}x_{N+1})P_N(I - K_{N+1}x_{N+1})^T + K_{N+1}K_{N+1}^T$$

Si les paramètres évoluent brusquement, une solution consiste à réinitialiser $P_N = \lambda I$ avec λ grand. Si les paramètres évoluent lentement on peut utiliser :

$$\hat{\theta}_{N+1} = \hat{\theta}_N + K_{N+1} \left(y_{N+1} - x_{N+1}\hat{\theta}_N \right) \quad (4.4)$$

$$K_{N+1} = P_N x_{N+1}^T (\lambda I + x_{N+1} P_N x_{N+1}^T)^{-1} \quad (4.5)$$

$$P_{N+1} = (P_N - K_{N+1}x_{N+1}P_N) \frac{1}{\lambda} \quad (4.6)$$

Cependant cet algorithme présente l'inconvénient de faire croître P_N de façon exponentielle s'il n'y a plus d'excitation.

Voici d'autres choix pertinents :

- à gain constant : on force alors $P_{N+1} = P_N$; cela évite la diminution du gain en cours de recherche des paramètres, et on donne ainsi plus de poids aux acquisitions les plus récentes; cette option convient bien à un système dont les paramètres varient.
- à gain décroissant : $P_N = cst$, cas classique, qui convient à un système à paramètres constants.
- à trace constante : on garde $\text{tr}(P_N) = cst$ (multiplication par un facteur correctif à chaque itération); on a les mêmes avantages que dans la recherche à gain constant mais on module en cours de recherche le poids relatif de chaque paramètre.

Cette méthode présente les mêmes avantages que la méthode des moindres carrés non récursive, mais aussi le même inconvénient : l'estimateur est biaisé! Aussi on lui préfère généralement d'autres méthodes telle que la méthode de la variable instrumentale récursive.

4.2 Variable instrumentale récursive

L'algorithme, très proche de celui énoncé en (4.4) à (4.6).

$$\hat{\theta}_{N+1} = \hat{\theta}_N + K_{N+1} \left(y_{N+1} - x_{N+1}\hat{\theta}_N \right) \quad (4.7)$$

$$K_{N+1} = P_N Z_{N+1}^T (\sigma^2 + x_{N+1} P_N Z_{N+1}^T)^{-1} \quad (4.8)$$

$$P_{N+1} = P_N - K_{N+1}x_{N+1}P_N \quad (4.9)$$

Dans ce cas Z_{N+1} représente un vecteur ligne instrumental composé, un peu comme la matrice instrumentale, soit d'observations retardées de la sortie soit de la sortie d'un modèle auxiliaire.

Beaucoup d'autres algorithmes existent, le lecteur intéressé en trouvera pléthore dans la littérature. Néanmoins ils sont tous à peu près fondés sur le même principe. En fait, ils se résument à l'expression d'un asservissement à gain réglable : on asservit les paramètres pour annuler une erreur, d'équation ou de sortie selon la méthode. La valeur du gain influe, comme sur n'importe quel système, sur la stabilité, soit la convergence de la méthode, et sur la rapidité.

Moduler le gain revient donc à moduler la rapidité de convergence : si l'on a affaire à un procédé dont il faut poursuivre les paramètres, qui varient effectivement dans le temps, on aura intérêt à avoir un gain fort ; si par contre le système est fortement bruité, on a intérêt à avoir un gain faible, sinon la sortie du modèle va poursuivre le bruit en faisant varier les paramètres à chaque période d'échantillonnage, ce qui n'a pas de sens physique. Un bon réglage du gain demande plusieurs essais et pas mal de bon sens !

Chapitre 5

Algorithmes d'optimisation paramétrique.

5.1 Présentation générale.

Ce chapitre est consacré au choix d'une méthode d'optimisation des paramètres. Une étude bibliographique succincte montre que le domaine de l'optimisation paramétrique est en pleine évolution notamment dans le cadre des méthodes stochastiques.

Nous présenterons les méthodes les plus connues afin d'en extraire les principes d'optimisation les plus aptes à résoudre notre problème.

Les méthodes présentées sont en fait celles qui reviennent le plus souvent dans la littérature. Parfois anciennes, ces méthodes, en raison de leur capacité à optimiser des fonctions très différentes, sont encore souvent utilisées aujourd'hui. Par ailleurs, ces méthodes sont souvent à la base de nouvelles méthodes d'optimisation.

Le domaine d'utilisation de chaque méthode est un point délicat à aborder. Une façon simple de déterminer ce domaine serait de le limiter au domaine de convergence garantie, en général les fonctions convexes. Mais cette approche est quelque peu réductrice et ne tient pas compte de la robustesse de la méthode, c'est pourquoi les spécialistes de l'optimisation paramétrique préfèrent se référer à des tests sur des fonctions connues et présentant des caractéristiques des plus diverses.

5.2 Exemple introductif

Imaginons que la fonction à minimiser soit l'altitude et que l'espace sur lequel nous cherchons à optimiser l'altitude soit la surface de notre planète. Il existe plusieurs façons de procéder.

La première façon consiste à parcourir l'ensemble des points de la surface et de mesurer l'altitude en chaque point. En supposant un maillage de la planète de 10m sur 10m et à raison d'une mesure par seconde : cela représente environs 160 000 ans de travail (énumération).

La deuxième méthode consiste à partir d'un point et de descendre dans le sens de la plus grande pente. Nous évitons ainsi tous les sommets locaux mais un observateur qui partirait d'un point des Alpes a peu de chances de trouver le fond de la mer (gradient). Nous pouvons améliorer cette méthode en prenant trois observateurs qui se communiquent leur altitude. Celui qui se situe au point le plus haut se déplace en direction d'un point situé entre les deux autres observateurs et parcourt deux fois la distance qui le sépare de ce point. Ces observateurs s'épargnent le calcul de la pente mais ont toujours peu de chances de trouver la mer (Nelder et Mead).

Une troisième méthode consiste à donner à notre observateur initial une soucoupe volante (et amphibie) qui lui permet de se déplacer très vite. Pendant un temps il se pose au hasard sur la planète et note l'altitude et la position de chaque point. puis au cours du temps il concentre sa recherche autour des points les plus bas en effectuant des déplacements aléatoires mais de plus en plus courts en moyenne. Cet observateur trouvera très rapidement un océan et déterminera son point le plus bas. Par contre si il se concentre trop vite sur l'un des océans il risque de rater le point le plus bas de la planète (recuit simulé).

La quatrième méthode consiste naturellement à prendre plusieurs observateurs qui feraient le même travail que le précédent. Rapidement il se constitueraient en groupes qui déterminent le relief de chaque mers et océans. Lorsqu'un groupe atteint le minimum de son étendue d'eau et remarque qu'un autre groupe obtient un minimum plus faible, le premier groupe se déplace pour aider un autre groupe n'ayant toujours pas atteint le fond (algorithmes génétiques).

5.3 Problème posé

L'idée générale de l'optimisation paramétrique consiste à déterminer un ensemble de paramètres qui optimise un critère soumis à des contraintes. Dans la suite de ce mémoire, nous parlerons exclusivement de minimisation, la maximisation étant la minimisation de l'opposé de la fonction.

$$\begin{array}{ll}
 \text{minimiser} & f(x), \\
 \text{avec} & x = \{x_1, x_2, x_3, \dots, x_n\}, \\
 \text{soumis à} & C_k(x) \geq 0 \quad k = 1 \text{ à } m, \\
 \text{et} & C_k(x) = 0 \quad k = m \text{ à } p.
 \end{array} \tag{5.1}$$

Dans la première partie de ce chapitre, nous nous consacrons à la détermination de trajectoires optimales *sans* contrainte. Ce cas, bien qu'inapplicable dans le cas des robots marcheurs, peut très bien l'être pour des robots manipulateurs.

5.4 Revue bibliographique succincte.

Les principales méthodes d'optimisation peuvent être classées en trois catégories principales :

1. Les méthodes analytiques, qui déterminent le point de calcul suivant en fonction des caractéristiques de la fonction au point considéré.
2. Les méthodes stochastiques, qui intègrent une part de hasard dans le choix du point de calcul suivant.
3. L'énumération, qui consiste à discrétiser l'espace engendré par le vecteur à optimiser et à calculer l'ensemble des possibilités.

La figure 5.1 présente les différents groupes d'algorithmes d'optimisation paramétrique et leur classification.

L'énumération : Elle consiste à prendre un nombre fini de valeurs pour chaque paramètre et à calculer le critère associé pour l'ensemble des possibilités. Dans notre cas elle est inutilisable compte tenu du nombre de paramètres. En utilisant 50 valeurs par paramètre et 13 paramètres par axe, il faut examiner $50^{13 \times 5}$ possibilités!

Les algorithmes d'ordre n : Fondés sur une décomposition à l'ordre n en séries de Taylor de la fonction à minimiser, ces algorithmes utilisent la dérivée d'ordre n pour la détermination du point de calcul suivant. Ils sont très rapides lorsque la fonction est continue, dérivable et convexe. Par contre, ils sont peu robustes si la fonction présente des minima locaux.

Les algorithmes d'ordre 0 : N'utilisant que la valeur de la fonction en certains points, ces algorithmes sont lents mais robustes en cas de discontinuités de la fonction.

Les algorithmes génétiques : Transposition à l'informatique de la sélection naturelle selon Charles Darwin, ils utilisent plusieurs vecteurs de paramètres (une population) qui évoluent vers le minimum par la mise en œuvre de trois opérations stochastiques, la sélection, le croisement et la mutation.

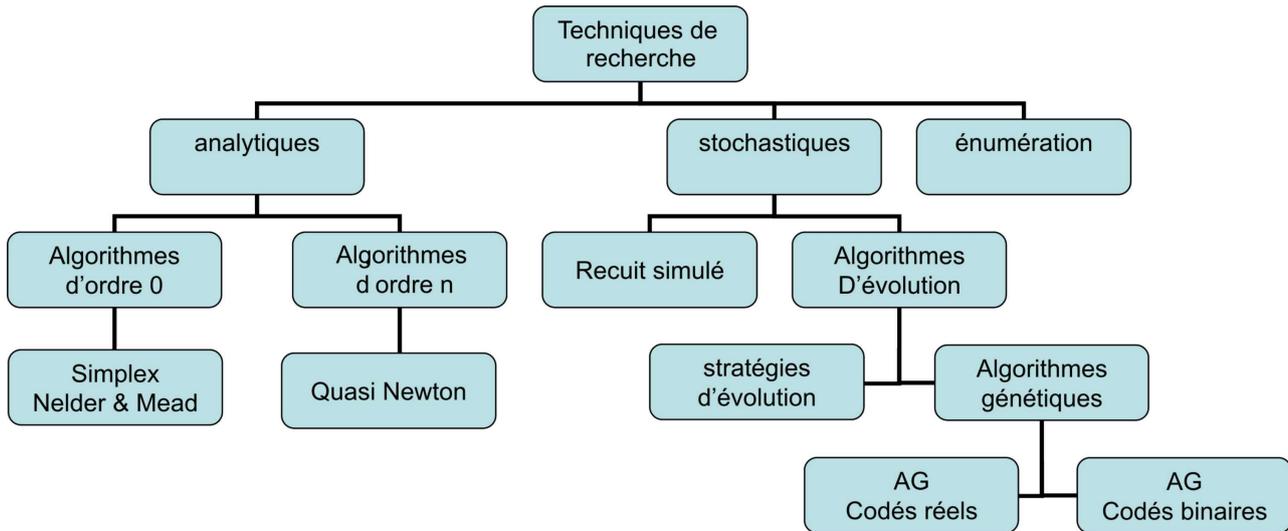


FIGURE 5.1 – Classes d’algorithmes d’optimisation paramétrique.

Les algorithmes de "recuit simulé" : Algorithmes stochastiques par excellence, ils sont statistiquement capables de déterminer le minimum global d’une fonction quelles que soient ses caractéristiques. Par contre, le temps de calcul nécessaire à l’obtention du minimum est long.

Les stratégies d’évolution : Fondées sur une évaluation itérative de la fonction, le déplacement est aléatoire mais il s’adapte en fonction de la topologie locale de la fonction à minimiser. La différence fondamentale entre stratégies d’évolution et le "recuit simulé" est que la première méthode explore la fonction de façon locale alors que la deuxième explore l’ensemble de l’espace.

5.5 Éléments de comparaison des différentes méthodes.

5.5.1 Critères.

La robustesse est l’aptitude de la méthode à converger vers un minimum global sans être perturbée par des minima locaux. Cette notion couvre les notions de robustesse à la rugosité de la fonction et la notion d’aptitude à trouver le minimum global d’une fonction.

La vitesse est mesurée en nombre d’itérations sur la fonction de coût, le temps de calcul nécessaire pour l’algorithme de la méthode elle-même est considéré comme négligeable.

La globalité du minimum est donnée pour une minimisation d’une fonction multimodale sans aucune connaissance a priori de la position du minimum global, donc d’une initialisation aléatoire.

5.5.2 Comparaisons.

Nous reprendrons ici quelques comparaisons trouvées dans la littérature afin d’établir un "classement" des différentes méthodes par rapport aux trois critères définis précédemment, à savoir : la robustesse, la vitesse et la globalité du minimum.

Les algorithmes d’ordre n sont très rapides, d’autant plus rapides que les fonctions optimisées répondent aux hypothèses formulées, par contre ils sont peu robustes en cas de discontinuités. Dans le cas de discontinuités, parmi les méthodes analytiques, ce sont les algorithmes d’ordre 0 qui sont les meilleurs.

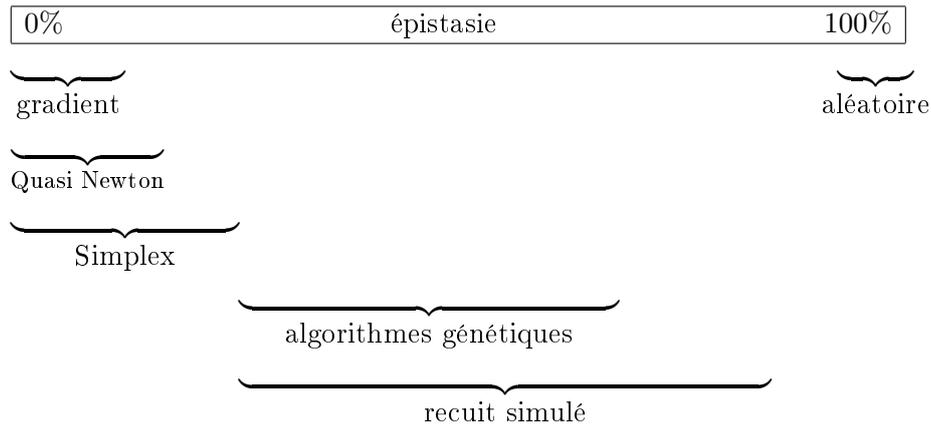


FIGURE 5.2 – Récapitulatif des différentes méthodes en fonction de l'épistasie.

5.5.3 Récapitulatif.

Le tableau 5.1 est une synthèse des données recueillies dans la littérature. Les notations sous forme de + et – permettent seulement une comparaison qualitative entre les méthodes sans augurer de la qualité intrinsèque d'une méthode.

Principe	robustesse	vitesse	minimum
ordre 1 : Gradient	- - -	++	local
ordre 2 : Quasi Newton	- -	+++	local
ordre 0 : Simplex	-	+	local
Algorithmes d'évolution	+	- -	global -
Algorithmes génétiques	+	- - -	global ++
Recuit simulé	++	- - -	global ++

TABLE 5.1 – Comparaison des méthodes d'optimisation.

La figure 5.2 donne le domaine d'utilisation de chaque méthode en fonction de l'épistasie. Cette notion directement tirée du vocabulaire de la génétique indique la propension d'un gène à inhiber ou augmenter l'effet d'un autre gène sur la valeur de la fonction. Ainsi une fonction non linéaire, multimodale présente une forte épistasie, par contre près d'un minimum local de cette fonction, l'épistasie est minimale voire nulle.

Afin de fixer les idées, voici trois exemples de fonctions :

$$f_1(x) = x_1^2 + x_2^2 + x_3^2, \quad \text{épistasie nulle} \quad (5.2)$$

$$f_2(x) = 100(x_1^2 - x_2)^2 + (1 - x_3)^2, \quad \text{épistasie moyenne} \quad (5.3)$$

$$\begin{aligned} f_3(x) &= (x_2^2 - x_3)^2 \text{ si } \sin(x_1) > 0 \\ &= (x_1^2 - x_2)^2 \text{ si } \sin(x_1) < 0, \quad \text{épistasie forte} \end{aligned} \quad (5.4)$$

Ainsi $f_1(x)$ présente une épistasie nulle puisque la valeur d'un gène ne change pas l'influence d'un autre gène sur le critère. Dans $f_2(x)$ l'influence existe entre x_1 et x_2 . Dans $f_3(x)$ l'influence est forte puisque la valeur de x_1 décide de quels gènes interviennent dans le critère.

5.5.4 Synthèse

La meilleure méthode serait une méthode qui évolue en fonction de la connaissance de la fonction à optimiser. Ainsi la recherche dans ce domaine s'oriente vers le couplage de plusieurs méthodes

d'optimisation. Le problème d'optimisation revient alors à effectuer successivement :

1. Définition du problème et en particulier des limites de l'espace à étudier notamment par la prise en compte de limites sur la valeur des différents paramètres à optimiser.
2. Génération aléatoire d'une population respectant les limites précédentes et évaluation de chaque individu vis-à-vis du critère.
3. Optimisation "lente" permettant :
 - une exploration de l'espace total,
 - la détermination de sous-espaces convexes présentant un minimum local.
4. Optimisation rapide dans les sous-espaces convexes.

Le passage d'une méthode à l'autre peut être fondé sur la détermination de l'épistasie.

Chapitre 6

Description des algorithmes utilisés.

6.1 Gradient et quasi-Newton

Ces méthodes consistent à écrire un développement limité de la fonction autour du point courant, puis d'en déduire par la minimisation de ce développement limité, le point suivant. Le lecteur intéressé par ce chapitre lira avec bonheur *Numerical recipes in C* [?] où les "détails" des algorithmes (initialisation, choix de décroissance, ...) seront parfaitement explicités.

6.1.1 Méthode du gradient

La dérivée de $f : \mathbb{R}^n \rightarrow R$ s'écrit :

$$Df(x)_i = \frac{\partial f}{\partial x_i}$$

$Df(x)$ est un vecteur ligne.

Le gradient de la fonction est :

$$\nabla f(x) = Df(x)^T = \left(\frac{\partial f}{\partial x} \right)^T$$

$\nabla f(x)$ est un vecteur colonne.

La fonction f à minimiser peut être approchée autour d'un point x_0 par :

$$f(x_0 + \Delta x) \simeq f(x_0) + \nabla f(x_0)^T \Delta x$$

le point de calcul suivant sera calculé par :

$$x_{i+1} = x_i + \Delta x$$

le pas étant déterminé par :

$$\Delta x = -k \nabla f(x)$$

k étant un réel positif. Le choix de k donne lieu à de nombreuses variantes de l'algorithme.

- Si k est trop petit, l'algorithme converge lentement.
- Si k est trop grand, l'algorithme oscille autour du minimum.

En pratique on choisit k variable au cours de l'optimisation, la plus connue est le *line search*. Une méthode simple consiste à faire varier k suivant :

- Si $f(x_{i+1}) < f(x_i)$ alors augmenter k (on est dans la bonne direction, on va voir plus loin)
- Si $f(x_{i+1}) = f(x_i)$ alors augmenter k (on est sur un plateau, on va voir plus loin)
- Si $f(x_{i+1}) > f(x_i)$ alors diminuer k et abandonner $f(x_{i+1})$ (on remonte, le minimum est plus près de x_i)

6.1.2 Méthode de Newton

C'est en fait, l'extension à l'ordre 2 de la méthode du gradient. Ainsi la fonction f à minimiser est approchée par :

$$f(x_0 + \Delta x) \simeq f(x_0) + \nabla f(x_0)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x_0) \Delta x$$

ou $\nabla^2 f(x_0)$ est le Hessien de la fonction (la dérivée seconde)

$$D\nabla f(x) = \nabla^2 f(x)_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \frac{\partial^2 f}{\partial x_i \partial x_j} \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

Le point où la fonction approchée est minimum est tel que la dérivée de la fonction est nulle soit l'équation :

$$\frac{\partial}{\partial \Delta x} \left(f(x_0) + \nabla f(x_0)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x_0) \Delta x \right) = 0$$

$$\nabla f(x_0)^T + \nabla^2 f(x_0) \Delta x = 0$$

le pas est alors :

$$\Delta x = -\nabla^2 f(x)^{-1} \nabla f(x)$$

On voit que par rapport à la méthode du gradient, la direction de recherche est pondérée par l'inverse du hessien. En pratique cette méthode est très peu utilisée car la détermination du hessien est longue et coûteuse en temps de calcul, son inverse aussi. On lui préférera des méthodes fondées sur l'approximation du hessien.

6.1.3 Méthode de Newton-Raphson

Comme pour la méthode de Newton le pas est calculé par :

$$\Delta x = -\nabla^2 f(x)^{-1} \nabla f(x)$$

Mais au lieu de calculer le hessien puis son inverse¹ on se contente de calculer Δx tel que :

$$\nabla^2 f(x) \Delta x = -\nabla f(x)$$

Notez bien que l'on a transformé le calcul de l'inverse de $\nabla^2 f(x)^{-1}$ en une résolution d'un système linéaire à chaque itération.

6.1.4 Méthode de Levenberg-Markardt

Les deux méthodes précédentes présentent l'inconvénient (que quelques variantes contournent) de ne pas converger lorsque le point initial est loin du minimum recherché.

L'algorithme Levenberg-Markardt lève cet inconvénient en proposant un compromis entre la méthode du gradient (robuste mais lente à l'approche du minimum) et celle de Newton (peu robuste loin du minimum mais très efficace près). Le pas Δx_i s'obtient en résolvant le système d'équations linéaires :

$$(\nabla f(x_i) \nabla f(x_i)^T + \lambda_i I) \Delta x_i = -\nabla f(x_i)^T f(x_i)$$

ainsi, lorsque λ_i est grand devant le hessien, l'algorithme tend vers l'algorithme du gradient, lorsque λ_i est petit, il converge aussi vite que celui de Newton.

1. Pour une fonction de 100 paramètres le hessien est une matrice 100×100 !

6.1.5 l'algorithme BFGS

BFGS sont les initiales de Broyden-Fletcher-Goldfarb-Shanno, c'est sans doute l'algorithme le plus utilisé.

A partir d'un point initial x_0 et une première approximation du hessien H_0

1. Calculer le pas par :

$$H_i \Delta x_i = -\nabla f(x_i)$$

2. Déterminer α_i optimal par une méthode de type "line search" et calculer :

$$x_{i+1} = x_i + \alpha_i \Delta x_i$$

3. Calculer la variation du gradient :

$$y_i = \frac{\nabla f(x_{i+1}) - \nabla f(x_i)}{\alpha_i}$$

4. Puis la nouvelle estimation du hessien est :

$$H_{i+1} = H_i + \frac{y_i y_i^T}{y_i^T \Delta x_i} + \frac{H_i \Delta x_i \Delta x_i^T H_i^T}{\Delta x_i^T H_i \Delta x_i}$$

L'ensemble de ses méthodes convergent bien vers un minimum à condition que la fonction soit convexe ! Le principal inconvénient de ses méthodes est de devoir calculer le gradient de la fonction à chaque itération aussi les méthodes suivantes se passent de cette étape de calcul et sont nettement plus robustes si la fonction présente des minima locaux.

6.2 Simplex de Nelder & Mead.

Parmi les algorithmes d'ordre 0, la méthode d'optimisation "Simplex" de Nelder & Mead [?], est l'une des plus anciennes connues et est encore aujourd'hui souvent utilisée (*fminsearch* de matlab). La démonstration de la convergence de cet algorithme dans le cas de fonctions convexes, n'a été réalisée qu'en 1995.

Un Simplex est un tableau de $n + 1$ vecteurs (x_i) de n paramètres (α_i). La méthode procède à une série de :

- Réflexions (le volume du Simplex reste constant)
pour déplacer le barycentre du Simplex en direction du point de critère le plus faible.
- Expansions (le volume augmente)
qui étendent le Simplex en direction du point de critère le plus faible.
- Contractions (le volume diminue)
qui lui permettent de passer dans des "goulets".

Algorithme

L'algorithme est présenté par l'ordinogramme de la figure 6.5 Le vecteur donnant le critère le plus grand à la $k^{\text{ème}}$ itération ($x_{k_{max}}$) est projeté à travers le barycentre des autres vecteurs du Simplex (\bar{x}_k).

$$x_r = (1 + \alpha)\bar{x}_k - \alpha x_{k_{max}}, \quad \text{avec } \alpha = 1, \quad \text{réflexion}, \quad (6.1)$$

La figure 6.1 représente la position des vecteurs du Simplex dans l'espace des paramètres (ici représenté dans un espace de dimension 3).

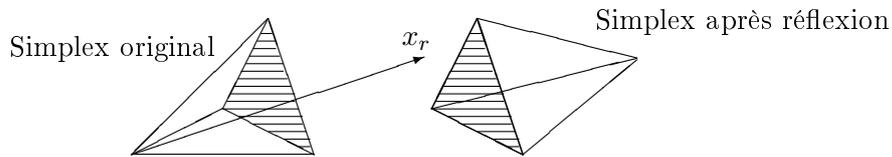


FIGURE 6.1 – Image d'un Simplex de 4 vecteurs avant et après réflexion.

Si la valeur du critère $f(x_r)$, associée au vecteur x_r est la plus faible obtenue jusqu'à présent, on tente une expansion du Simplex dans la direction de la projection. Sinon on conserve x_r et on recommence une nouvelle itération. La procédure d'expansion (voir figure 6.2 est donnée par :

$$x_e = \gamma x_r + (1 - \gamma)\bar{x}_k, \quad \text{avec } \gamma = 2, \quad \text{expansion}, \quad (6.2)$$

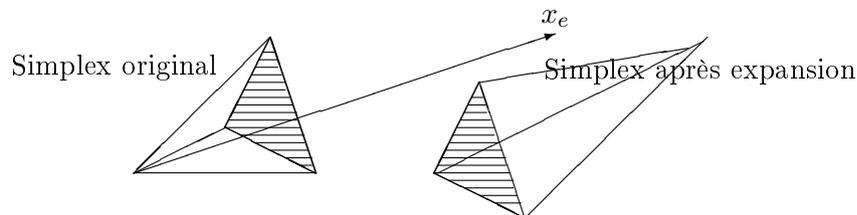


FIGURE 6.2 – Image d'un Simplex de 4 vecteurs avant et après expansion.

Si la valeur du critère $f(x_e)$ est plus faible que celle associée aux autres vecteurs du Simplex on la conserve, sinon on conserve x_r et l'on passe à l'itération suivante.

Si $f(x_r)$ est plus grande que la valeur maximale des vecteurs du Simplex, on tente une compression (voir figure 6.3) suivant la direction de la projection.

$$x_c = \beta x_{max} + (1 + \beta)\bar{x}_k, \quad \text{avec } \beta = 1/2, \quad \text{compression,} \quad (6.3)$$

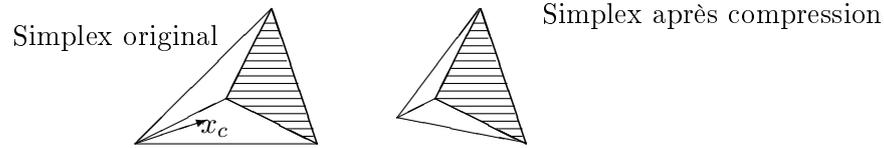


FIGURE 6.3 – Image d'un Simplex de 4 vecteurs avant et après compression.

Si $f(x_c)$ est plus grande que la plus grande valeur associée des autres vecteurs du Simplex, on procède alors à une contraction (voir figure 6.4) de tous les vecteurs du Simplex en direction du vecteur donnant le critère le plus faible ($x_{k_{min}}$).

$${}^i x_{k+1} = \frac{{}^i x_k + x_{k_{min}}}{2}, \quad \text{contraction.} \quad (6.4)$$

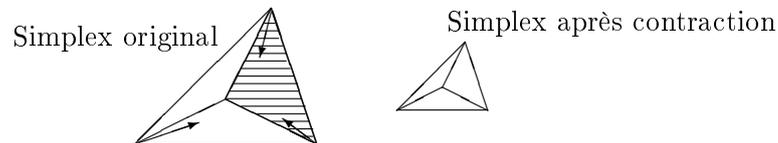


FIGURE 6.4 – Image d'un Simplex de 4 vecteurs avant et après contraction.

Initialisation. L'initialisation du Simplex peut être quelconque à condition que les vecteurs soient linéairement indépendants. L'initialisation classique est de la forme :

$$x_i = x_0 + \delta u_i, \quad i = 1..n, \quad (6.5)$$

où :

$$\begin{cases} x_0 = & : \text{vecteur initial quelconque,} \\ x_i = & : \text{vecteurs du Simplex,} \\ \delta = & : \text{paramètre réel,} \\ u_i = & : \text{vecteurs linéairement indépendants (en général normés).} \end{cases}$$

Les vecteurs u_i forment donc une base de l'espace des paramètres.

Critère d'arrêt . Le critère d'arrêt de l'optimisation est :

$$s = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}} < \epsilon \quad (6.6)$$

ϵ étant une valeur réelle positive choisie au départ.

Le choix de ϵ est limité par :

- La borne supérieure est liée à la valeur approximative du critère optimal. L'application de 6.6 pour une précision relative voulue de 1% sur un critère optimal environ égal à 10 conduit au choix de $\epsilon = 5 \times 10^{-3}$.
- La borne inférieure est liée à la précision du calculateur. On montre que ϵ doit être choisi comme égal à la racine carrée du plus petit nombre qui peut être représenté, ce qui donne, pour des calculs effectués en virgule flottante et en double précision, une limite inférieure de $\epsilon = \sqrt{1,7 \times 10^{-308}} = 1,3 \times 10^{-154}$.

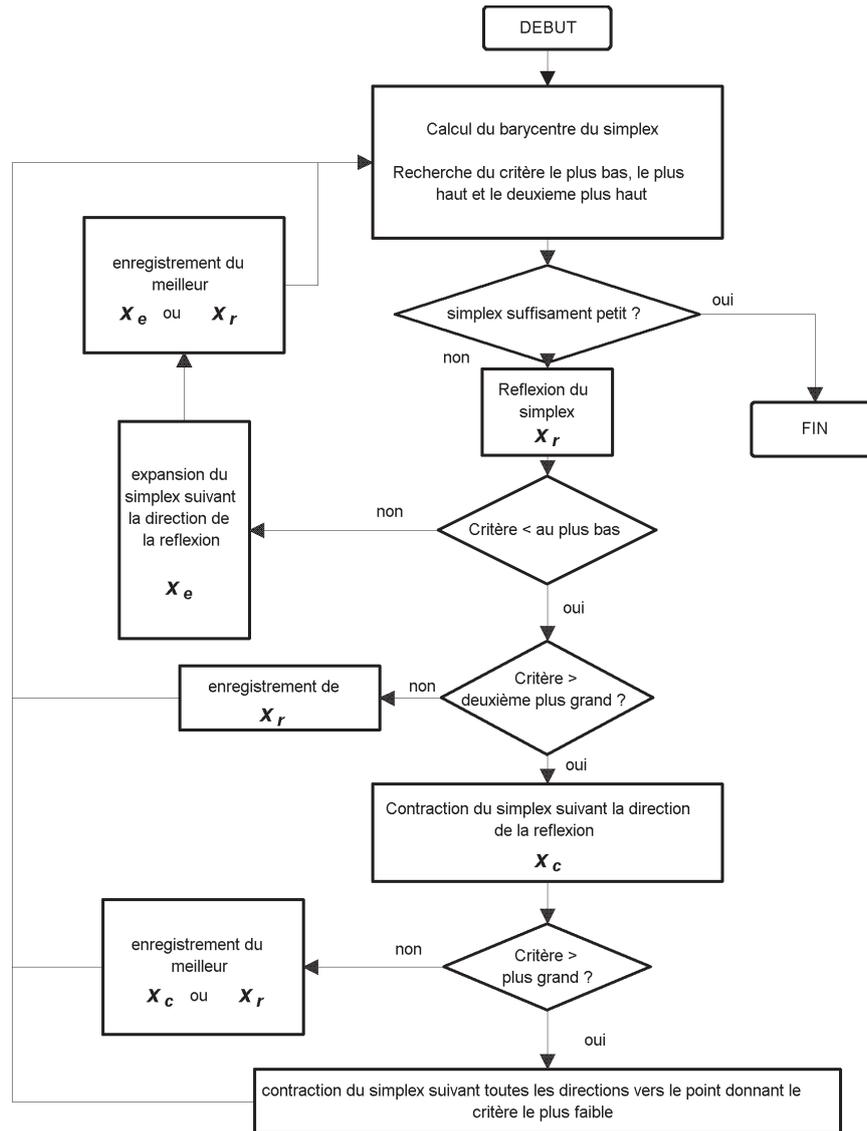


FIGURE 6.5 – Ordinogramme de l’algorithme du Simplex de Nelder et Mead.

6.2.1 Extensions - Améliorations.

Les améliorations sont souvent liées à la vitesse de convergence de l’algorithme. Notons parmi les plus connues :

- La méthode du barycentre pondéré (Weighted Centroid) :
Le point réfléchi est calculé par réflexion non pas par rapport à l’isobarycentre mais par rapport au barycentre pondéré par l’inverse de la valeur du critère. Ainsi le point réfléchi tend à se rapprocher des points ayant le plus faible critère.
- La méthode du Simplex super modifiée (Super Modified Simplex) : : Après calcul du critère au point réfléchi et du critère du barycentre, nous faisons passer une équation du second degré par les trois points (original, le barycentre et le point réfléchi) dont on détermine l’optimum. C’est ce dernier point qui sera choisi comme point réfléchi.

L’introduction d’une variable aléatoire dans la recherche de nouveaux points rend l’optimisation plus globale. Cette amélioration sort déjà du cadre des méthodes d’ordre 0 pour entrer dans le cadre des Algorithmes Génétiques (voir §6.3) ou des méthodes du ”recuit simulé” (voir §6.5).

Conclusions.

Cette méthode est la plus robuste des méthodes de type analytique. Les capacités d'adaptation du volume du Simplex ainsi que l'absence de calcul du gradient en font une méthode rapide, peu sensible aux discontinuités de la fonction de coût et facile à mettre en œuvre.

6.3 Algorithmes génétiques.

6.3.1 Introduction.

Développés par Holland en 1970, ces algorithmes sont la transposition algorithmique de la théorie de Ch. Darwin sur la "survivance du plus fort" ou la "sélection naturelle des espèces". Selon Ch. Darwin, les individus les plus faibles disparaissent, seuls les individus les plus forts se reproduisent et parfois ceux-ci mutent.

6.3.2 Principe des algorithmes génétiques binaires.

Les algorithmes génétiques binaires travaillent sur une entité fondamentale appelée chromosome qui est en fait une chaîne de bits, l'unité d'information la plus simple. L'utilisation de cette structure fondamentale liée au ordinateur et non pas au processus à optimiser permet une grande adaptabilité aux différents problèmes (combinatoires, optimisation de critères fonction de variables réelles). Ainsi le vecteur de paramètres initial est transformé en une chaîne de bits.

$$x = (\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n) \longrightarrow x = (b_1, b_2, b_3, b_4, b_5, \dots, b_m) \quad (6.7)$$

où :

$$x \quad : \quad \text{est un individu ou chromosome,} \quad (6.8)$$

$$b_i \quad : \quad \text{sont les gènes.} \quad (6.9)$$

Le passage de l'un à l'autre fera l'objet du codage.

L'algorithme travaille sur une population de taille constante (30 à 300 individus typiquement) qui évolue à chaque itération, pour créer une nouvelle génération.

Initialisation. La population initiale est créée aléatoirement.

Sélection. La sélection des individus qui vont se reproduire est aléatoire mais fonction de la valeur de leur critère associé. L'individu j est sélectionné suivant une probabilité p_{s_j} .

$$p_{s_j} = \frac{f(jx)}{\sum_{i=1}^n f(ix)}. \quad (6.10)$$

Il existe une autre possibilité : la sélection des parents à croiser suivant leur rang (après classement de l'ensemble de la population suivant la valeur de leur critère associé) et non la valeur de leur critère associé. Cette deuxième méthode présente l'avantage d'empêcher une convergence trop rapide vers un individu nettement meilleur que les autres.

Croisement. Le croisement est l'étape principale de l'optimisation. Cette opération consiste à engendrer 2 enfants ayant un chromosome composé de parties des chromosomes des 2 parents.

Exemple avec 2 points de croisement (||) :

$$\left. \begin{array}{l} \text{parent1} = 001||00110010||1011 \\ \qquad \qquad \qquad \uparrow \downarrow \\ \text{parent2} = 101||10011100||1101 \end{array} \right\} \longrightarrow \begin{array}{l} \text{enfant1} = 001||10011100||1011 \\ \text{enfant2} = 101||00110010||1101 \end{array} \quad (6.11)$$

Si la position des points de croisement était constante, combiné à la sélection, le croisement aurait pour effet d'optimiser chaque partie de la chaîne. Mais la position des points de croisement est aléatoire, ce qui participe à l'exploration de l'espace total. La probabilité de croisement, c'est-à-dire la probabilité que deux individus sélectionnés pour le croisement subissent effectivement le croisement est en général de $p_c = 0,5$ à $0,7$.

Mutation. La mutation d'un individu permet de réaliser l'aristogénèse et sert ainsi à empêcher la perte de diversité. En effet si au cours de l'optimisation un gène de rang donné avait la même valeur pour toute une génération, tous les descendants conserveraient ce gène. La probabilité de mutation est en général de $p_m = 0,01$ à $0,05$.

$$\text{individu} = 001100\|1\|10001011 \longrightarrow \text{individu} = 001100\|0\|10001011 \quad (6.12)$$

Codage. Les fonctions que l'on veut minimiser étant dans la grande majorité des cas des fonctions de variables réelles, un codage des chaînes binaires est nécessaire. Le codage le plus utilisé est le codage binaire à virgule fixe, par exemple :

$$1001,1010 \longrightarrow 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} = 9,625 \quad (6.13)$$

La longueur de la chaîne dépend donc de la précision souhaitée sur les valeurs réelles. Certains auteurs préfèrent le code de Gray qui présente l'avantage d'avoir des codes consécutifs toujours distants de 1, Mais ce codage diminue l'aptitude à l'exploration de l'espace total par le croisement. Ces méthodes de codage donnent une grande facilité pour imposer des contraintes sur les valeurs d'un paramètre.

6.3.3 Extensions - Améliorations.

Les algorithmes génétiques sont actuellement en pleine expansion. Le principal problème actuel des algorithmes génétiques étant la faible propension à l'exploration de l'espace complet, les recherches portent en particulier sur ce point.

- Longueur de la chaîne de gènes dynamique.
- Notion de niche écologique.
- Sur les croisements :
 - croisement multipoint (3,4,..) au lieu de 2 comme décrit précédemment,
 - croisement uniforme qui est un croisement bit à bit avec une probabilité d'échanger les bits qui dépend du nombre d'échanges déjà effectués. Le croisement uniforme augmente la capacité d'exploration de l'espace total, il est surtout utilisé dans le cas de populations faibles (dans le cas de fortes populations, le nombre élevé d'individus assure déjà un bonne exploration de l'espace total).
- Conservation d'une génération sur l'autre du ou des meilleurs individus.

6.3.4 Conclusion.

Cette méthode est bien adaptée à la minimisation dans le cas de fonctions fortement non linéaires. La mutation est surtout utile dans le cas d'un critère à forte épistasie, le croisement étant plutôt utile dans la phase de détermination des minima locaux. Les algorithmes génétiques sont une bonne méthode intermédiaire entre les méthodes analytiques et les méthodes stochastiques.

6.4 Algorithmes génétiques codés réels.

Afin de résoudre le problème du codage et de la précision des paramètres dans les algorithmes génétiques, certains auteurs ont transposé les idées des algorithmes génétiques à des gènes de type réel.

6.4.1 Principe.

La séquence des opérations reste identique à celle des algorithmes génétiques binaires, seules les opérations et le codage diffèrent.

Un chromosome est composé d'un ensemble de gènes réels de la forme :

$$\text{chromosome} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \dots, \alpha_n), \text{gènes} = \alpha_i, \quad i = 1..n. \quad (6.14)$$

Croisement. Le croisement, qui consiste à engendrer des descendants par échange de gènes, devient :

$$\left. \begin{array}{l} \text{parent1} = (\alpha_1, \|\alpha_2, \alpha_3, \|\alpha_4, \alpha_5) \\ \qquad \qquad \qquad \uparrow \downarrow \\ \text{parent2} = (\beta_1, \|\beta_2, \beta_3, \|\beta_4, \beta_5) \end{array} \right\} \longrightarrow \begin{array}{l} \text{enfant1} = (\alpha_1, \|\beta_2, \beta_3, \|\alpha_4, \alpha_5) \\ \text{enfant2} = (\beta_1, \|\alpha_2, \alpha_3, \|\beta_4, \beta_5) \end{array} \quad (6.15)$$

ou

$$\left. \begin{array}{l} \text{parent1} = (\alpha_1, \|\alpha_2, \alpha_3, \|\alpha_4, \alpha_5) \\ \qquad \qquad \qquad \uparrow \downarrow \\ \text{parent2} = (\beta_1, \|\beta_2, \beta_3, \|\beta_4, \beta_5) \end{array} \right\} \longrightarrow \begin{array}{l} \text{enfant1} = (\alpha_1, \|\frac{\alpha_2+\beta_2}{2}, \frac{\alpha_3+\beta_3}{2}, \|\alpha_4, \alpha_5) \\ \text{enfant2} = (\beta_1, \|\frac{\alpha_2+\beta_2}{2}, \frac{\alpha_3+\beta_3}{2}, \|\beta_4, \beta_5) \end{array} \quad (6.16)$$

Mutation. Là encore, la mutation consiste à empêcher la perte de matériel génétique. Transposé au codage réel, cela revient à ajouter un bruit sur un ou plusieurs chromosomes choisis aléatoirement.

$$\alpha'_i = \alpha_i \pm p \times \alpha_i. \quad (6.17)$$

où p est un nombre aléatoire de moyenne nulle.

Renders propose une mutation dont l'amplitude décroît avec le temps, de la forme suivante :

$$\begin{aligned} \alpha'_i &= \alpha_i + (\alpha_{i_{max}} - \alpha_i) \left(1 - r \left(1 - \frac{k}{k_0}\right)^5\right) \text{ si } b = 1, \\ \alpha'_i &= \alpha_i + (\alpha_{i_{min}} - \alpha_i) \left(1 - r \left(1 - \frac{k}{k_0}\right)^5\right) \text{ si } b = 0, \end{aligned} \quad (6.18)$$

où :

- $\alpha_{i_{max}}$: valeur maximum du paramètre α_i ,
- $\alpha_{i_{min}}$: valeur minimum du paramètre α_i ,
- r : nombre réel aléatoire donnant l'amplitude de la mutation,
- k : indice de la génération courante,
- k_0 : indice de la dernière génération subissant une mutation,
- b : nombre binaire aléatoire donnant le sens de la mutation.

Par analogie avec les algorithmes de type "recuit simulé" (voir §6.5), l'amplitude des mutations décroît au fur et à mesure que l'algorithme converge.

Améliorations. Afin d'améliorer la vitesse de convergence, il est courant de mettre en œuvre une combinaison de plusieurs méthodes d'optimisation : algorithmes génétiques et méthodes analytiques. L'algorithme génétique recherche les minima locaux et la méthode analytique détermine rapidement la valeur du minimum de la fonction dans chaque minimum local.

6.4.2 Conclusion.

Les algorithmes génétiques réels présentent l'avantage de ne pas être limités par le codage des variables réelles et la forme générale de l'algorithme permet un passage aisé des algorithmes génétiques vers des méthodes de type Simplex.

6.5 Méthode du "Recuit simulé"

6.5.1 Introduction

La méthode du "recuit simulé" (*Simulated Annealing*) repose sur un principe de la thermodynamique : la loi de refroidissement des métaux. Un atome donné est animé par un mouvement brownien dont l'amplitude dépend de la température. Lorsque la température décroît ce mouvement s'estompe et l'atome tend à se positionner dans un état d'énergie minimale.

6.5.2 Principe.

Le problème posé est la détermination d'un vecteur x qui minimise le critère $\lambda(x)$. La méthode du "recuit simulé" la plus simple résout le problème par un processus itératif et stochastique comme suit. Après une initialisation aléatoire du premier point de calcul x_0 , l'algorithme génère, à chaque itération, un pas aléatoire δx suivant une loi de probabilité de type gaussienne $g(\delta x)$.

$$g(\delta x) = \frac{1}{(2\pi T(k))^{-n/2}} e^{-\frac{(\delta x)^2}{2T(k)}} \quad (6.19)$$

Le nouveau point de calcul est :

$$x_{k+1} = x_k + \delta x. \quad (6.20)$$

Deux cas peuvent se présenter :

1. Le nouveau point est meilleur : $\lambda(x_{k+1}) < \lambda(x_k)$,
le point est accepté et le processus recommence.
2. Le nouveau point est plus mauvais : $\lambda(x_{k+1}) > \lambda(x_k)$,
le nouveau point peut être accepté suivant une probabilité :

$$p_k = \exp\left[-\frac{\lambda(x_{k+1}) - \lambda(x_k)}{T(k)}\right] \quad (6.21)$$

A chaque itération la "température" $T(k)$ diminue suivant la loi :

$$T(k) = \frac{T_0}{\ln(1+k)}. \quad (6.22)$$

Une valeur minimale de T_0 est proposée par Hajcek qui assure une convergence statistique vers un minimum global.

Ainsi lorsque la température est grande, l'algorithme peut quitter un minimum local, soit par un saut direct, soit par un point intermédiaire de critère plus grand. Au fur et à mesure que la température décroît, la longueur du pas diminue et la probabilité d'accepter un pas δx qui fait augmenter le critère décroît. L'algorithme finit par générer des pas de faible amplitude et n'accepter pratiquement plus de pas qui font augmenter le critère, les pas qui le font diminuer étant toujours acceptés. Cette méthode permet statistiquement de trouver un minimum global mais demande un très grand nombre d'itérations.

6.5.3 Extensions - Améliorations.

Par extension, de nombreuses méthodes se rapportent à celle-ci du moment que la probabilité d'acceptation du pas est de la forme de celle présentée par l'équation (6.21).

Parmi les améliorations proposées dans la littérature, notons celle appelée : "Fast Annealing" où $g(\delta x)$ et $T(k)$ sont choisis tels que :

$$g(\delta x) = \frac{T(k)}{((\delta x)^2 + T(k)^2)^{(n+1)/2}} \quad (6.23)$$

et

$$T(k) = \frac{T_0}{k} \quad (6.24)$$

L'auteur propose donc une loi de décroissance plus rapide, tout en conservant la mobilité du point de calcul.

D'autre part, notons aussi la modification de [?] qui introduisent dans la méthode du Simplex une probabilité d'acceptation du pas. Cet algorithme présente l'avantage de converger vers l'algorithme classique du Simplex (voir [?]) lorsque la température décroît.

6.5.4 Mise en œuvre du "recuit simulé".

Pour la mise en œuvre de la méthode du "recuit simulé", nous avons choisi l'algorithme proposé par [?].

L'algorithme présente trois phases distinctes mais imbriquées :

1. lorsque la température est grande vis-à-vis du critère, pratiquement tous les déplacements sont acceptés, le Simplex s'agrandit et explore l'espace,
2. lorsque la température est du même ordre de grandeur que le critère, le Simplex converge vers les grandes "vallées" tout en acceptant des déplacements qui augmentent le critère, ce qui lui permet de sortir des rugosités de celui-ci,
3. lorsque la température est faible vis-à-vis du critère, l'algorithme détermine le minimum local qui, selon toute probabilité, est le minimum global.

Choix de la température. Il est nécessaire de choisir une température initiale T_0 et une loi de décroissance au cours du temps. B. Hajcek a démontré qu'une décroissance de la forme :

$$T(k) = \frac{T_0}{\ln(1+k)} \quad (6.25)$$

associée à une probabilité d'acceptation du pas de la forme :

$$p_k = \exp \left[-\frac{\lambda(x_{k+1}) - \lambda(x_k)}{T(k)} \right] \quad (6.26)$$

assure statistiquement la détermination d'un minimum global à condition de choisir T_0 supérieur au maximum de profondeur de tous les minima locaux, minimum global exclu.

Le maximum de profondeur de tous les minima locaux est impossible à déterminer, c'est pourquoi certains auteurs [?] préconisent l'emploi d'une valeur largement supérieure à une approximation de ce second membre.

Nous avons borné cette profondeur maximale par :

$$T_0 = 10 \times \text{la valeur maximale du critère à l'initialisation}, \quad (6.27)$$

la valeur minimale du critère étant minimisée par 0.

Bibliographie

Liens

<http://www.mathworks.com/products/fuzzylogic.html>

http://www.mathtools.net/MATLAB/Fuzzy_Logic/index.html

<http://www.dbai.tuwien.ac.at/marchives/fuzzy-mail/>

<http://europa.eu.int/comm/research/rtdinfo/fr/24/02.html>

<http://www.gala.univ-perp.fr/~polit/chap0.html>

<http://www.eru.ulaval.ca/ptt15225>

http://vcampus.u-strasbg.fr/uticeweb/mhiri/projet_V2

<http://www.mathworks.fr/products/controldesign/modanal.shtml>

Chapitre 7

Travaux dirigés

TDAO : Illustration de plusieurs aspect de l'identification des systèmes

7.1 Strecj - Broïda - Programmation non linéaire

7.1.1 Strecj - Broïda

- Ouvrez le fichier simulink intitulé "TDAOident.mdl", lancez la simulation.
- Appliquez la méthode de Broïda puis celle de Strecj et implantez-les.
- Relancez la simulation, et observez les "qualités" des deux modèles.
- En utilisant les *Process Models* de l'interface *ident* de matlab, retrouver les modèles déterminés précédemment (Strecj et Broïda). Le processus à identifier est en fait un système du troisième ordre sans zéros ni retard, déterminez à l'aide des *Process Models* de l'interface *ident* un modèle du système. Comparer les valeurs numériques obtenues aux valeurs vraies.
- Pour aller plus loin, synthétisez un correcteur PI sur le modèle de Broïda puis implantez-le sur le système "réel". Observer la qualité de ce contrôle pourtant calculé sur un modèle ostensiblement éloigné de la réalité.

7.1.2 Programmation non linéaire

- Ouvrez le fichier "pnl.m" et comprenez le contenu.
- Dans la fenêtre de commande matlab, tapez "save ES Uin tout Yout" puis, "X=fminsearch('pnl',[-1 -1 -1 1])" (tapez "help fminsearch" et lisez la doc.)
- la valeur de X présente le résultat sous la forme "3 pôles, gain"
- implantez ce nouveau modèle et relancez la simulation.

7.1.3 Limites de la Programmation non linéaire

- lancez "X=fminsearch('pnl',[0 0 0 0])", comparez les solutions.
- Dans le fichier "pnl.m" changez l'utilisation de x0 en introduisant des zéros et plus de pôles.

7.2 Identification des systèmes instables ou intégrateurs

Ouvrez le fichier simulink intitulé "instable.mdl". Ce système est instable mais stabilisé par le gain de 0.1 et le bouclage. Les entrées - sorties sont dans le *workspace* de Matlab et s'importent facilement dans l'interface *ident* (Tapez "ident" dans la fenêtre de commande de Matlab), voir figure 7.1.

Note : les modèles calculés dans "ident" s'exportent vers le *workspace* par glisser déplacer dans la case *to workspace* puis du workspace vers simulink par une commande comme :

```
[A1,B1,C1,D1]=ssdata(amx2122)
```

- Observez les caractéristiques spectrales de la SBPA et comparez à celles du signal carré.

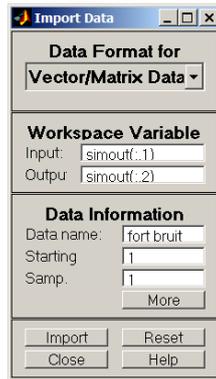


FIGURE 7.1 – Comment importer des variables dans ident de Matlab.

- Après avoir annulé le bruit, importez les données puis calculez un modèle de type ARX752.
 - observez la qualité de la simulation.
 - observez les pôles et les zéros de ce modèle.
 - déduisez-en un modèle plus simple et tout aussi correct.
 - vérifiez que vous obtenez bien les coefficients du modèle en boucle fermée.
- Observez la différence d'estimation des paramètres du système en fonction de :
 - la puissance de bruit (0, 0.01 et 0.0001)
 - les différents modèles (en particulier, ARX, ARMAX, BOX-JENKINS)
 - les signaux utilisés pour l'estimation (carré ou SBPA)
 - la position de la source d'excitation (entrée de la boucle ou juste devant le système à identifier)
- Reprenez ce qui précède en prenant comme entrée la sortie du correcteur.

7.3 Choix de la source d'excitation

L'objectif est de déterminer la bonne séquence binaire pseudo-aléatoire pour exciter correctement le système.

- Ouvrez le fichier "sbpa_ident.mdl"
- Le réglage de la période de la SBPA s'effectue en déclarant la variable TeSBPA=1 (ici 1 seconde).
- Déterminer un premier modèle en utilisant un "process model" avec 2 pôles simples.
- Déduisez-en une approximation des constantes de temps max et min du système.
- Déterminer une SBPA susceptible de bien exciter le système dans toute la plage de fréquences que vous voulez identifier.
- Identifier à nouveau le système.

7.4 Les modèles paramétriques

Il s'agit de prédire le cours d'une action du CAC40 à partir des cours des autres actions.

- Exécuter le fichier "trait_don".
- Importez les variables "entree" comme entrée et "sortie" comme sortie.
- Calculez et observez les réponses des différents modèles (ARX, ARMAX, BOX-JENKINS...).
- En particulier, notez l'erreur de prédiction 5 pas en avance.
- Peut-on devenir riche ? Si oui, comment ?
- Valider vos modèles à l'aide des données 'entreeV' comme entrée et "sortieV".

7.5 Cas pratique

Le système modélisé dans le fichier "MEMSswitch.mdl" contient un modèle "réaliste" d'un interrupteur fabriqué en technologies MEMS que l'on désire utiliser comme un bras de levier commandé en position.

L'objectif est de déterminer un modèle paramétrique du système.

- L'ordre de grandeur de l'entrée est de 50V.
- Le comportement est du type second ordre.
- Les caractéristiques dynamiques varient un peu en fonction de la tension de commande continue.

La mise en place des données physiques du système est faite par le fichier "donneesMEMS.m".

Débrouillez-vous !

Deuxième partie

Annexes

Examen d'Identification et Commande Floue

Cet examen consiste à modéliser et à commander le système de commande d'altitude d'un avion. Il s'agit dans un premier temps de déterminer un modèle simple du système de façon graphique, puis de commander le système par un régulateur flou. Enfin, le système étant corrigé par ce régulateur flou on déterminera un modèle paramétrique plus précis.

1 Méthodes graphiques

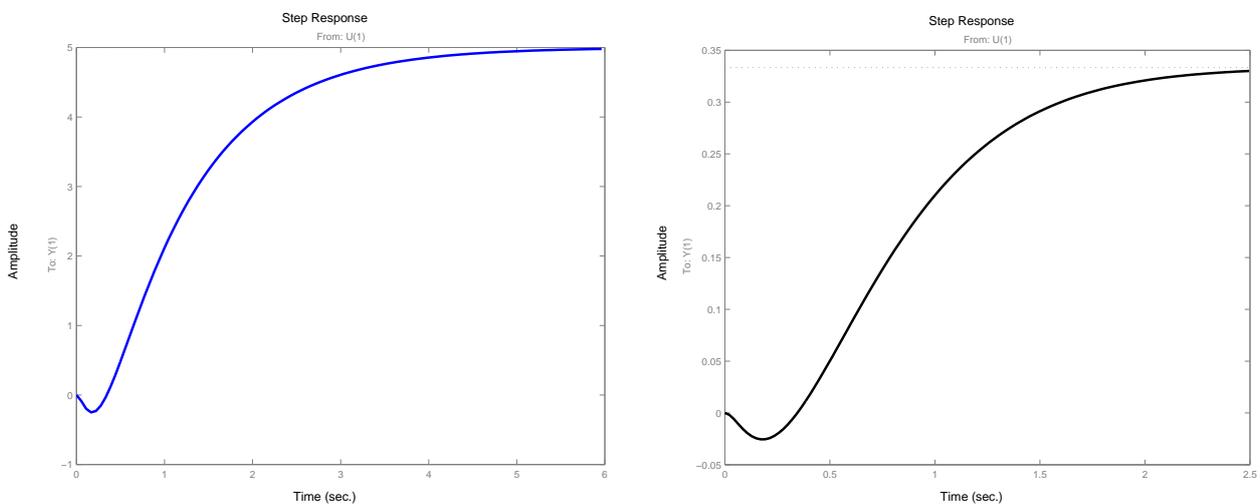


FIGURE 2 – Réponses du système à un échelon : à gauche en boucle ouverte, à droite en boucle fermée.

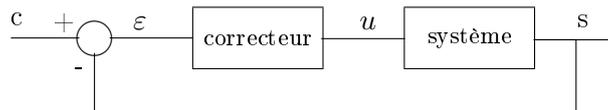


FIGURE 3 – Identification en boucle fermée

1. La réponse à un échelon du système en boucle ouverte est donnée sur la figure 2 (gauche). En faisant l'hypothèse que le système est un système du premier ordre avec un retard pur, donnez les valeurs numériques d'un tel modèle.
2. Toujours sur cette même figure appliquez la méthode de Strejc afin d'obtenir un modèle plus précis.
3. Ce même système est bouclé par un retour unitaire et un gain de 0,1 en guise de correcteur comme indiqué sur la figure 3. Sa réponse à l'échelon unité est donnée sur la figure 2 (droite).
 - (a) Déterminer un modèle du système en boucle fermée.
 - (b) En négligeant le retard pur, déduisez-en un modèle en boucle ouverte.
4. Finalement, on insère un gain variable dans la boucle et il s'avère que le système oscille pour un gain $k = 1$.
 - (a) Ce résultat est-il concordant avec vos modèles (raisonnement qualitatif) ?

- (b) Au vu des résultats précédents, proposez un modèle à trois constantes de temps
5. Ce système présente de façon évidente un zéro dit instable !
- (a) Proposez alors un modèle du système incluant ce zéro instable.
- (b) Donnez une valeur numérique très approchée (pratiquement sans calcul) de ce zéro agrémenté d'une ligne de commentaire sur ce choix.

2 Commande par contrôleur flou

Finalement, le système précédent est commandé par un contrôleur flou.

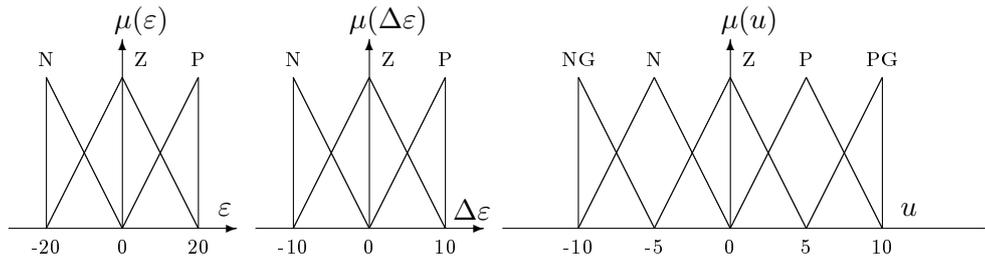


FIGURE 4 – Fuzzification des variables

$\varepsilon \setminus \Delta\varepsilon$	N	Z	P
N	NG	N	Z
Z	N	Z	P
P	Z	P	PG

avec opérateurs logiques : ET = min OU = max
 méthode d'inférence : somme-produit
 défuzzification : moyenne des maximums.

- En supposant $\varepsilon = 10$ et $\Delta\varepsilon = 0$, calculez la sortie u du contrôleur.
- Toujours en supposant $\Delta\varepsilon = 0$, calculez la sortie u du contrôleur en fonction de la valeur de ε . Tracez la courbe $u = f(\varepsilon)$ en précisant les points importants et la forme de la courbe entre ses points (droites, paraboles, ...)
- Même question mais en utilisant les hypothèses suivantes :
 - opérateurs logiques : ET = min OU = max
 - méthode d'inférence : max-min
 - défuzzification : centre de gravité

3 Modèles paramétriques

Le système étant corrigé par le régulateur flou précédent, on décide de déterminer un modèle paramétrique du système corrigé. Dans ce qui suit, et quels que soient vos résultats précédents, on supposera que le système présente :

- 3 constantes de temps de l'ordre de : 0,1, 0,05 et 0,02
- Calculer la fréquence d'échantillonnage nécessaire pour pouvoir identifier correctement ces trois constantes de temps.
 - Calculer la longueur de la séquence binaire pseudo aléatoire en fonction de cette fréquence d'échantillonnage pour bien identifier le gain du système.
 - Quelle est alors le temps d'excitation du système ?

4. On opte pour une méthode d'estimation des paramètres dite de la matrice instrumentale, définie par :

$$\hat{\theta} = (Z^T X)^{-1} Z^T y \quad \text{avec} \quad \left\{ \begin{array}{l} \hat{\theta}, \quad \text{Paramètres estimés;} \\ Z, \quad \text{Matrice instrumentale;} \\ y, \quad \text{Vecteur des sorties mesurées;} \\ X, \quad \text{Construite à partir du jeu d'E/S } (y = X\theta + e); \\ e, \quad \text{Résidus.} \end{array} \right.$$

Montrez que pour que cet estimateur soit sans biais, il faut que

$$\begin{aligned} E[Z^T X] &\neq 0 \\ E[Z^T e] &= 0 \end{aligned}$$

Corrigé de l'Examen d'Identification et Commande Floue

1 Méthodes graphiques

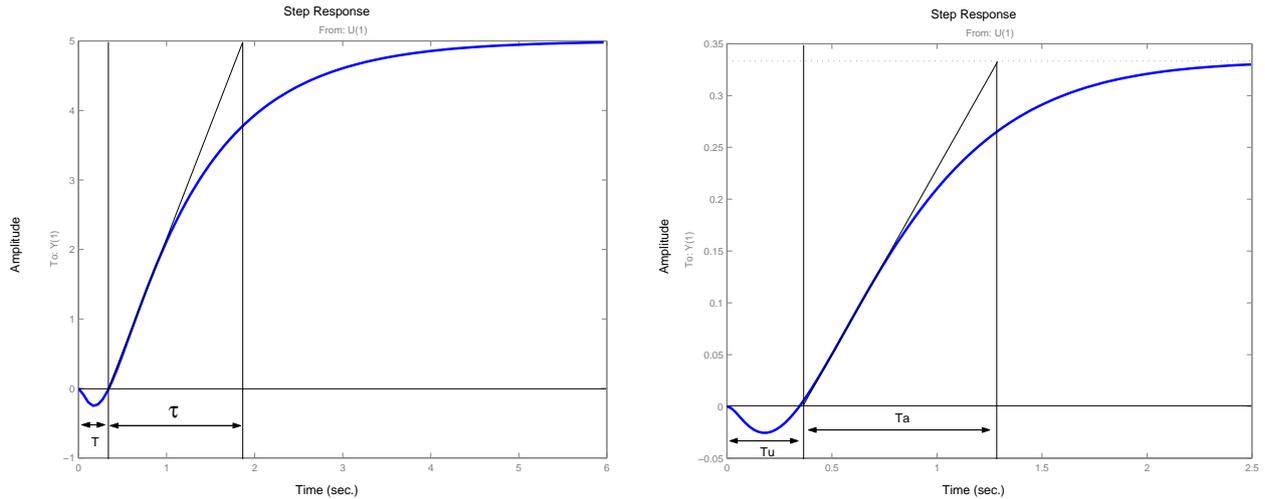


FIGURE 5 – Réponses du système à un échelon : à gauche en boucle ouverte, à droite en boucle fermée.

1. Modèle du premier ordre

On mesure $T = 0,3s$ et $\tau = 1,9 - 0,3 = 1,6s$ le modèle est donc :

$$G(p) = \frac{Ae^{-Tp}}{1 + \tau p} = \frac{5e^{-0,3p}}{1 + 1,6p}$$

2. Modèle Strejc en boucle ouverte

On mesure $Tu = 0,3s$ et $Ta = 1,6s$. En appliquant la méthode :

$$\begin{aligned} Tu/Ta &= 0,1875 \Rightarrow n = 2 \\ Ta/\tau &= 2,718 \Rightarrow \tau = Ta/2,718 = 0,5887 \\ Tu/\tau &= 0,282 \Rightarrow Tu' = 0,1660 \\ Tr &= Tu - Tu' \Rightarrow Tr = 0,134 \end{aligned}$$

d'où le modèle :

$$G_s(p) = \frac{Ae^{-Trp}}{(1 + \tau p)^n} = \frac{5e^{-0,166p}}{(1 + 0,5887p)^2}$$

3. En boucle fermée

(a) Modèle du système en boucle fermée,

On mesure $Tu = 0,33s$ et $Ta = 1s$. En appliquant la méthode :

$$\begin{aligned} Tu/Ta &= 0,330 \Rightarrow n = 4 \\ Ta/\tau &= 4,463 \Rightarrow \tau = Ta/4,463 = 0,2241 \\ Tu/\tau &= 1,425 \Rightarrow Tu' = 0,319 \\ Tr &= Tu - Tu' \Rightarrow Tr = 0,01 \end{aligned}$$

d'où le modèle :

$$G_s(p) = \frac{Ae^{-Trp}}{(1 + \tau p)^n} = \frac{0,34e^{-0,01p}}{(1 + 0,224p)^4}$$

(b) Modèle en boucle ouverte déduit :

$$G_s(p) = \frac{A/k}{\tau^4 p^4 + 4\tau^3 p^3 + 6\tau^2 p^2 + 4\tau p + 1 - A}$$

4. Finalement, on insère un gain variable dans la boucle et il s'avère que le système oscille pour un gain $k = 1$.

(a) A priori oui, les deux modèles obtenus sont susceptibles d'être instables dans une boucle fermée si le gain est suffisamment grand.

(b) le plus simple est de reprendre le modèle obtenu par la méthode de Strejc sur la boucle ouverte :

$$G_s(p) = \frac{Ae^{-Trp}}{(1 + \tau p)^n} \simeq \frac{A}{(1 + \tau p)^2(1 + Trp)}$$

5. Introduction du zéro instable

(a) Encore une fois, en reprenant le modèle précédent et en développant au premier ordre l'exponentielle, on obtient :

$$G_s(p) = \frac{Ae^{-Trp}}{(1 + \tau p)^n} \simeq \frac{A(1 - Trp)}{(1 + \tau p)^2}$$

(b) $Tr = 0.166$. En fait pour tenir compte des termes que l'on néglige on augmente "un peu" la valeur de Tr . Ainsi $Tr' = 1,5Tr$ donne de bien meilleurs résultats.

2 Commande par contrôleur flou

1. Si $\varepsilon = 10$ et $\Delta\varepsilon = 0$, la sortie vaut 2,5.

2. En supposant $\Delta\varepsilon = 0$, calculez la sortie u du correcteur en fonction de ε

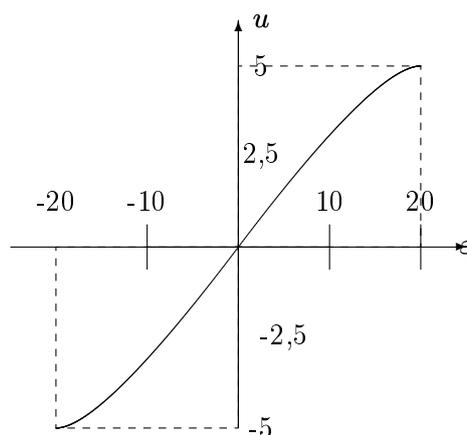


FIGURE 6 – $u = f(\varepsilon)$ (courbe exagérée !)

3. avec :

- opérateurs logiques : ET = min OU = max
- méthode d'inférence : max-min
- défuzzification : centre de gravité

3 Modèles paramétriques

Le système présente 3 constantes de temps de l'ordre de : 0,1, 0,05 et 0,02.

1. Une SBPA est un bruit blanc sur $0,3f_e$ donc :

$$0,3f_e \geq \frac{1}{2\pi\tau_{min}} \Rightarrow f_e \geq \frac{1}{0,3 \times 2\pi\tau_{min}} = \frac{1}{0,3 \times 2\pi \times 0,02} = 26,5Hz$$

2. Le plateau maximum est de NT_e avec $T_e = 1/f_e$ et $2^N - 1$ la longueur de la SBPA.
Il faut donc que :

$$NT_e \geq 3 \text{ à } 5\tau_{max} \Rightarrow N \geq 3 \text{ à } 5f_e\tau_{max} = 3 \text{ à } 5 \times 26,5 \times 0,1 = 8 \text{ à } 13$$

$N = 10$ semble un bon compromis.

3. Le temps d'excitation est :

$$T_{excitation} = (2^N - 1)T_e = 1023T_e = 38,6s$$

mais en général on envoie deux séquences consécutives pour éviter de prendre en compte le transitoire au début de la séquence, donc $T_{excitation} \simeq 77,2s$

4. Méthode de la matrice instrumentale :

$$\hat{\theta} = (Z^T X)^{-1} Z^T y \text{ avec } \begin{cases} \hat{\theta}, & \text{Paramètres estimés;} \\ Z, & \text{Matrice instrumentale;} \\ y, & \text{Vecteur des sorties mesurées;} \\ X, & \text{Construite à partir du jeu d'E/S } (y = X\theta + e); \\ e, & \text{Résidus.} \end{cases}$$

$$\begin{aligned} \hat{\theta} &= (Z^T X)^{-1} Z^T (X\theta + e) \\ &= \theta + (Z^T X)^{-1} Z^T e \end{aligned}$$

pour que $E[\hat{\theta}] = \theta$, il faut que :

$$E[(Z^T X)^{-1} Z^T e] = 0$$

donc que :

$$\begin{aligned} E[Z^T X] &\neq 0 \\ E[Z^T e] &= 0 \end{aligned}$$

3^{ème} année
Option mécatronique

Examen d'Identification

Cet examen consiste à modéliser un bras de robot présentant deux types de dérives lentes. En premier lieu, la constante de temps principale varie dans le temps puis le système présente une dérive lente de la valeur moyenne de sortie.

1 Méthodes graphiques

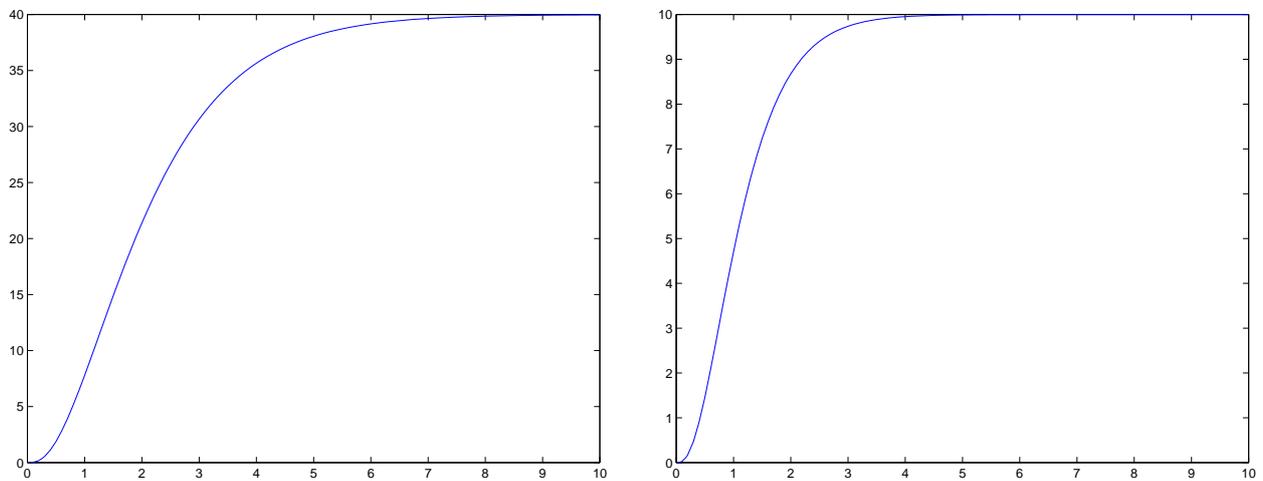


FIGURE 7 – Réponse du système à un échelon au début de l'identification (gauche) et après une heure (droite).

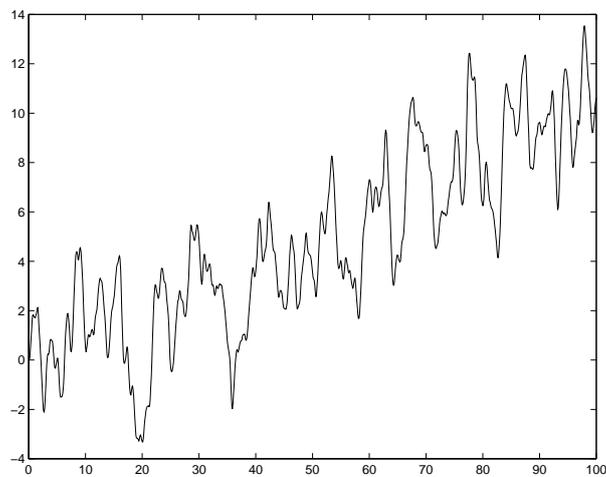


FIGURE 8 – Réponse du système à une excitation de type SBPA

2 Modèles de Strecj

1. La réponse à un échelon du système en boucle ouverte est donnée sur la figure 7(gauche). En faisant l'hypothèse que le système est un système du premier ordre avec un retard pur, donnez les valeurs numériques d'un tel modèle.
2. Toujours sur cette même figure appliquez la méthode de Strecj afin d'obtenir un modèle plus précis.
3. Appliquez Strecj sur la figure 7(droite) et déduisez-en un nouveau modèle du système.
4. En faisant l'hypothèse d'une variation linéaire de la constante de temps dominante et à l'aide des deux modèles précédents, proposez un modèle du système $H(p, t)$

3 Modèles ARX

Afin d'étudier plus précisément les autres constantes de temps du système, vous décidez de soumettre le système à une séquence binaire pseudo-aléatoire. Le résultat est donné fig. 8

1. Rappeler le principe de calcul des modèles ARX et la formule appliquée pour obtenir une estimation des paramètres du système.
2. La première tentative de modélisation ARX est très mauvaise ! Quels sont les deux phénomènes qui "empêchent" d'avoir un modèle ARX correct.
3. Afin de compenser la dérive lente de la valeur moyenne, vous envisagez deux méthodes
 - (a) Boucler le système, et faire une identification en boucle fermée,
 - (b) Enlever cette dérive par un traitement numérique hors ligne (un peu comme on enlève la valeur moyenne).
4. Proposez un correcteur simple pour le bouclage du système
5. Donnez la relation entre $y(t)$ valeurs de la figure 8 et $y_1(t)$ qui représente $y(t)$ sans la dérive. Les valeurs numériques approchées seront prises sur la figure 8.
6. Après calcul des modèles ARX, les modèles restent passablement mauvais avec un léger avantage pour le modèle calculé à partir des valeurs en boucle fermée, pourquoi ?
7. Les méthodes précédentes ne donnant pas de modèles suffisants pour l'application envisagée, proposez une méthode qui permet d'identifier ce système malgré toutes ses imperfections.

3^{ème} année
Option mécatronique

Examen de commande floue

Cet examen est destiné à déterminer un estimateur flou de pluie en fonction de la pression atmosphérique, du taux d'hygrométrie et de la température. Les valeurs numériques seront estimées grossièrement à partir des graphiques donnés où les abscisses sont en jours.

4 Fuzzification des variables

- Après avoir observé les relations entre quantité de pluie et les variables d'entrée en figure ?? (pression atmosphérique, taux d'hygrométrie, température et leurs dérivées) peu de temps avant le début des précipitations, proposez un choix d'ensembles flous pertinents. justifiez brièvement le choix du nombre d'ensembles flous par variable et la forme de ses ensembles.
- La forme de la variable de sortie est donnée figure 9. Etablissez les règles d'inférence pour chaque précipitation.

L'estimateur s'avère être trop cher (trop de capteurs) et votre cahier des charges devient donc "estimateur de pluie uniquement fonction de la pression atmosphérique et de ses variations".

- proposez, dans ce cas, les règles d'inférence sous la forme d'un tableau :

$p \setminus \Delta p$	N	Z	P
N			
Z			
P			

- Proposez une méthode pour chaque point suivant :
 - opérateurs logiques : ET = ? OU = ?,
 - méthode d'inférence,
 - défuzzification.
- Après un premier test en simulation il s'avère que votre estimateur prédit correctement dans 70% des cas mais le service marketing vous explique qu'il serait bon de donner outre le fait qu'il pleuve ou pas une estimation de la probabilité d'orage. L'orage est probable s'il y a une baisse de 1 à 2 hPa de la pression atmosphérique en une heure. Proposez un deuxième estimateur flou donnant cette indication.
- Question subsidiaire pour ceux qui ont suivi le cours d'identification. D'après vous, un modèle de type ARX serait-il meilleur ou moins bon pour ce type de prévisions ?

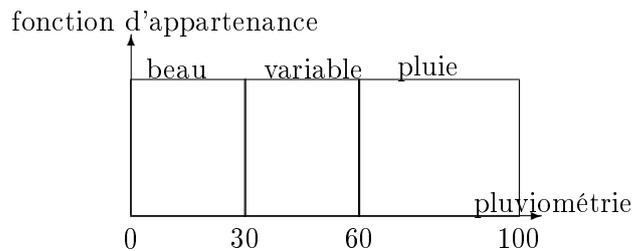


FIGURE 9 – Fuzzification de la variable de sortie

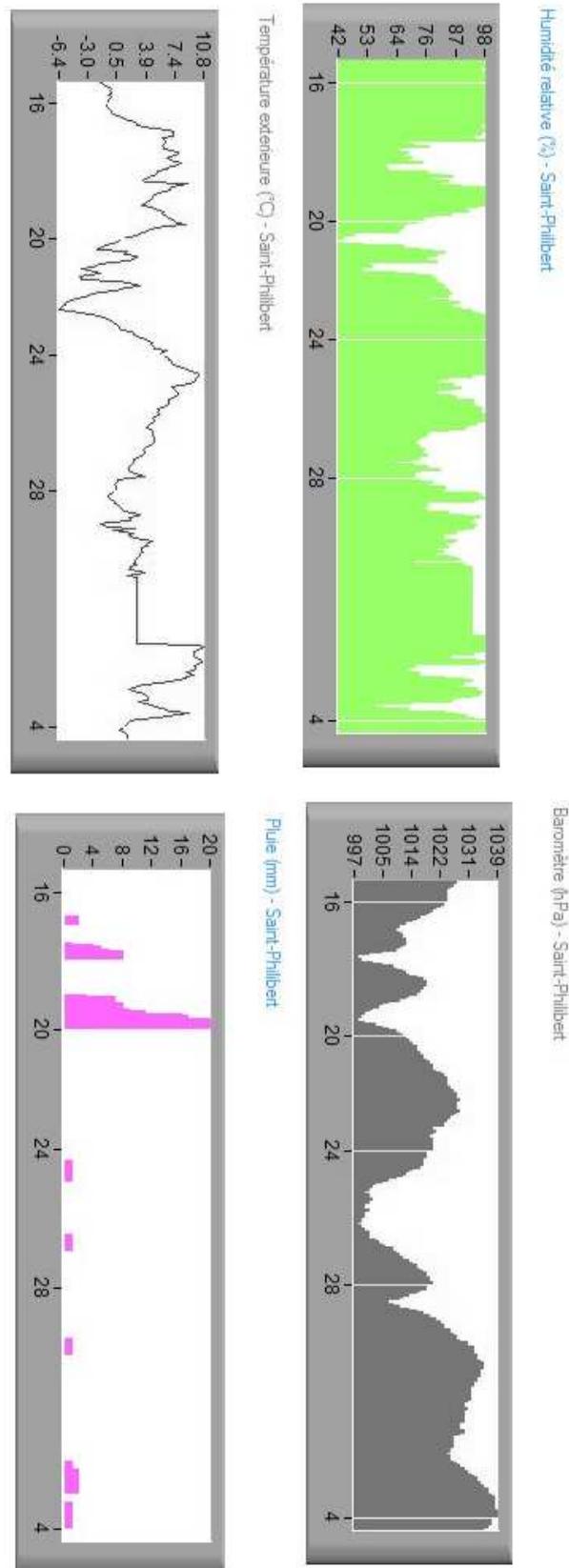


FIGURE 10 – Données météorologiques à St-Philibert (abscisses en jours)

3^{ème} année
 Janvier 2006
 Option mécatronique

Examen d'Identification des systèmes

Tous documents autorisés

Cet examen consiste à modéliser un système thermique particulièrement pénible à modéliser. Les questions suivantes sont liées mais peuvent être traitées dans le désordre.

1 Modèle de connaissances

La mise en équation du comportement dynamique du système donne le schéma linéaire donné en figure 11 avec les valeurs suivantes : $R_2 = 0,2$; $R_1 = 2,1$; $C = 1$; $L = 1$.¹ P est la puissance thermique introduite dans le système et T est la température que l'on veut asservir.

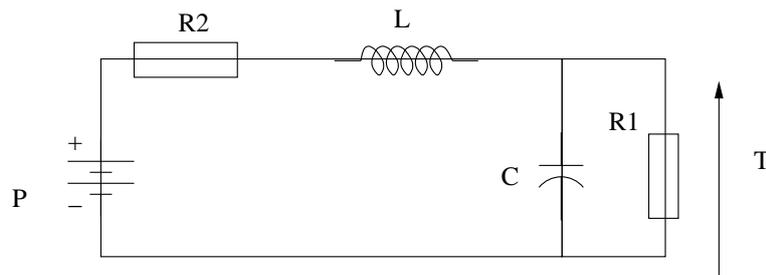


FIGURE 11 – Modèle thermique du système.

1. Déterminer la fonction de transfert linéaire continue (en p) du système $\frac{T(p)}{P(p)}$.
2. Calculer le facteur d'amortissement m du système.
3. Ce système s'avère trop peu amorti, on cherche alors à augmenter cet amortissement en agissant sur la seule variable R_1 (résistance thermique entre l'intérieur et l'extérieur de la chambre). Comment doit varier R_1 ?

2 Modèle non-paramétrique

Le modèle précédent étant particulièrement grossier, vous décidez de soumettre le système à un échelon unité démarrant à l'instant $t = 0$.

1. La réponse à un échelon du système en boucle ouverte est donnée sur la figure 12. En faisant l'hypothèse que le système est un système du second ordre avec un retard pur, donnez la forme générale du modèle.
2. Donnez les valeurs numériques des différents coefficients du modèle.

¹ Les inductances thermiques n'existent pas ! Il s'agit en fait d'un système actif modélisable en première approximation par une inductance.

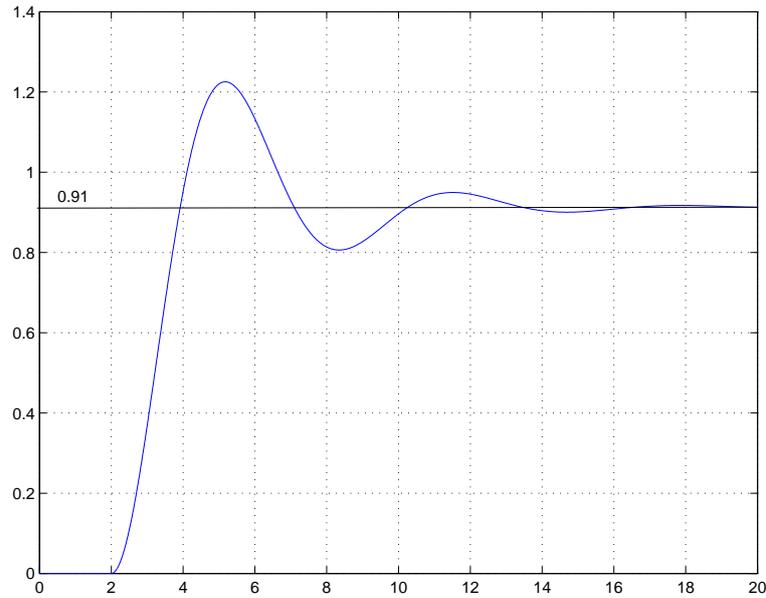


FIGURE 12 – Réponse impulsionnelle du système.

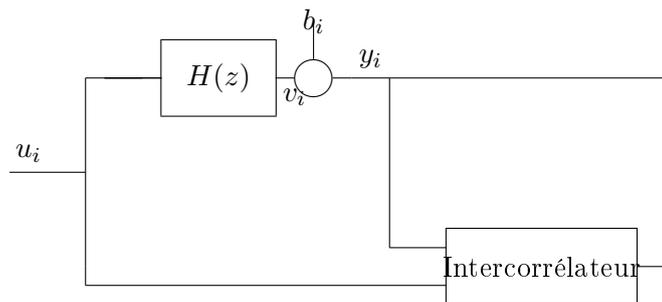


FIGURE 13 – Identification par intercorrélation.

3 Modèle par intercorrélation

Les deux modèles précédents ayant donné quelques nouvelles connaissances sur le système, vous décidez de mettre en œuvre une identification paramétrique fondée sur une identification par intercorrélation. Dans cette partie les réponses demandent **1 ligne** de calcul et une ou plusieurs interprétations du résultat. Pour les pinailleurs mathématiques, et ils ont raison de l'être, on intervertira les signes \sum et \lim sans démonstration.

$$y(i) = \sum_{j=0}^{\infty} (h(j) u(i-j)) + b(i)$$

avec :

- $y(i)$ = la sortie à l'instant i
- $h(j)$ = la séquence de pondération
- $u(i)$ = l'entrée à l'instant i
- $b(i)$ = le bruit à l'instant i

Par définition la fonction d'intercorrélation entre deux signaux y et u est :

$$C_{yu}(k) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{i=-N}^N y(i) u(i-k)$$

De la même façon, la fonction d'autocorrélation d'un signal u est :

$$C_{uu}(k) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{i=-N}^N u(i) u(i-k)$$

1. Quel est le lien entre $H(z)$ et $h(i)$ (cours des années précédentes)²
2. Calculez $C_{yu}(k)$ en fonction de l'entrée, du système et du bruit.
3. Introduisez dans ce résultat la fonction d'autocorrélation de l'entrée.
4. Montrez que si l'entrée et le bruit ne sont pas corrélés on obtient :

$$C_{yu}(k) = \sum_{j=0}^{\infty} h(j) C_{uu}(j-k)$$

5. Montrez que si l'entrée u est un bruit blanc parfait alors :

$$C_{yu}(k) = A h(k)$$

ou A est un coefficient lié à la puissance du bruit.

6. Que proposez vous pour réaliser ce bruit blanc ?
7. La réponse du système est donnée en figure 14. Est-elle compatible avec celle présentée en figure 12.

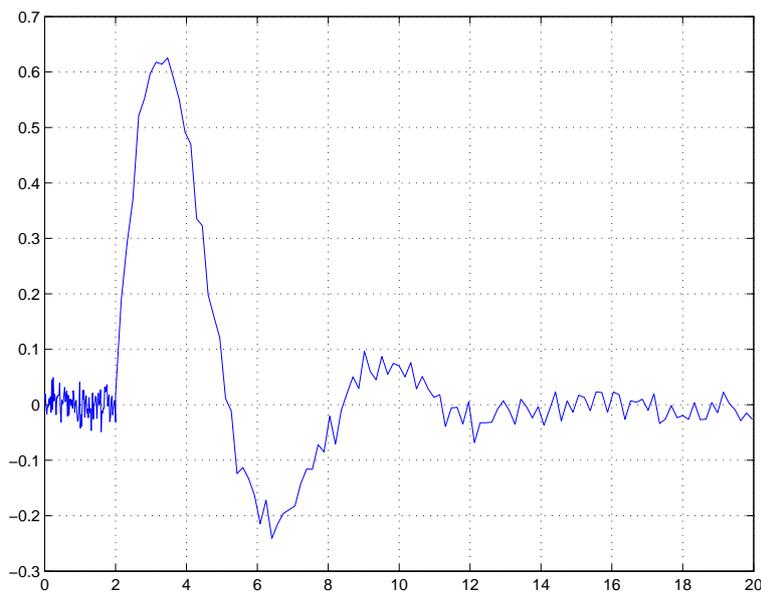


FIGURE 14 – Réponse de l'intercorrélation entrée-sortie du système.

2. Un peu d'aide? Calculer la sortie y_k pour k variant de 0 à 5 de $G(z) = \frac{1}{z-0,5}$ soumis à $\delta(z)$

4 Filtrage des mesures

Tout comptes fait, vous décidez de procéder à une identification paramétrique classique, et vous lancez une campagne de mesures entrée-sortie du système. Malheureusement, le stagiaire qui devait faire cela, trouvant que le signal de sortie était trop bruité n'a rien trouvé de plus malin que de filtrer les données de la sortie avec un filtre passe bas.

1. Que pouvez-vous faire pour utiliser au mieux ces données ?
2. Essayer de le démontrer.
3. Le premier résultat de calcul d'un modèle ARX 10 10 3 est donné en figure 15. Quel modèle tenteriez-vous ensuite ?

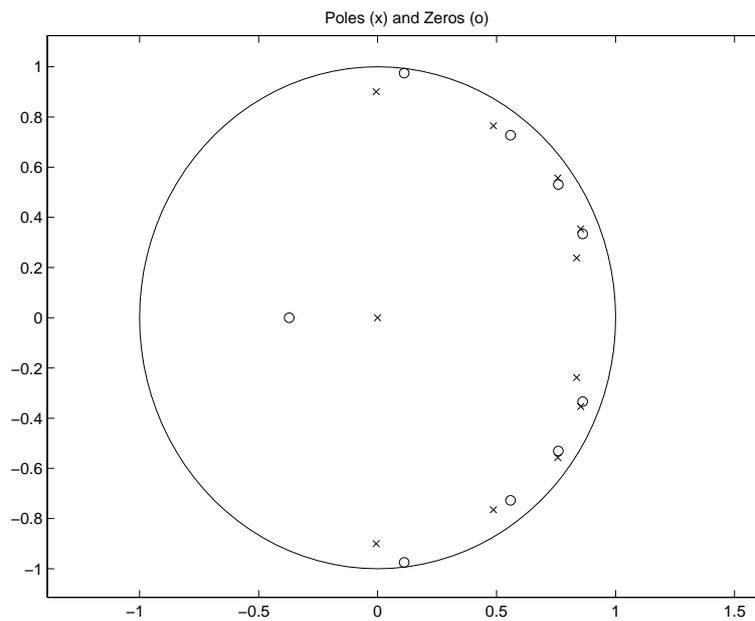


FIGURE 15 – Pôles et zéros du modèle ARX 10 10 3.

3^{ème} année
 Janvier 2006
 Option mécatronique

Corrigé de l'Examen d'Identification

1 Modèle de connaissances

1. Fonction de transfert linéaire continue

$$\frac{T(p)}{P(p)} = \frac{R_1}{R_1 LC p^2 + (L + R_1 R_2 C)p + R_1 + R_2}$$

2. Facteur d'amortissement

$$m = \frac{1}{2} \frac{L + R_1 R_2 C}{\sqrt{(R_1 + R_2) R_1 LC}}$$

3. R_1 doit diminuer, il faut donc augmenter les pertes.

2 Modèle non-paramétrique

1. Forme générale du modèle.

$$H(p) = \frac{Ae^{-\tau p}}{1 + \frac{2m}{w_0}p + \frac{p^2}{w_0^2}}$$

2. Coefficients du modèle.

$$\begin{aligned} A &= 0,91 \\ \omega_0 &= 1,04 \\ m &= 0,32 \\ \tau &= 2 \end{aligned}$$

τ est "visible" mais s'obtient par calcul en appliquant :

$$\begin{aligned} D\% &= 100e^{\frac{-\pi m}{\sqrt{1-m^2}}} && \text{donne } m \\ T_{\text{pseudo-période}} &= \frac{2\pi}{w_0\sqrt{1-z^2}} && \text{donne } w_0 \\ t_{pic_mesuré} - t_{pic_calculé} &= t_{pic_mesuré} - \frac{\pi}{w_0\sqrt{1-z^2}} && \text{donne } \tau \end{aligned}$$

3 Modèle par intercorrélation

1. $h(i)$ est la séquence de pondération de la fonction de transfert $H(z)$.
 - En temporel, tout système est un convoluteur : sa sortie est le produit de convolution entre l'entrée et sa séquence de pondération.
 - En z , cette convolution est transformée en un produit (c'est pour cela qu'on utilise les transformées de Laplace et en z !).

La séquence de pondération s'obtient facilement en calculant la sortie du système soumis à une impulsion de Dirac.

2. $C_{yu}(k)$

$$\begin{aligned}
C_{yu}(k) &= \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{i=-N}^N y(i) u(i-k) \\
&= \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{i=-N}^N u(i-k) \sum_{j=0}^{\infty} h(j) u(i-j) + b(i) \\
&= \sum_{j=0}^{\infty} h(j) \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{i=-N}^N u(i-k) u(i-j) + \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{i=-N}^N b(i) u(i-k) \\
&= \sum_{j=0}^{\infty} h(j) \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{i=-N}^N u(i) u(i-(j-k)) + \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{i=-N}^N b(i) u(i-k)
\end{aligned}$$

3. Introduisez dans ce résultat la fonction d'autocorrélation de l'entrée

$$C_{yu}(k) = \sum_{j=0}^{\infty} h(j) C_{uu}(j-k) + C_{bu}(k)$$

4. Pour que :

$$C_{yu}(k) = \sum_{j=0}^{\infty} h(j) C_{uu}(j-k)$$

il faut donc que

$$C_{bu}(k) = 0$$

Ce qui est le cas si l'entrée est le bruit sont décorrélés.

5. Si l'entrée u est un bruit blanc parfait alors

$$C_{uu}(k) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{i=-N}^N u(i) u(i-k) = A\delta(k)$$

Par définition d'un bruit blanc, sa fonction d'autocorrélation est une impulsion de Dirac et donc :

$$C_{yu}(k) = A h(k)$$

ou A est la "puissance" du bruit, en fait c'est le carré de sa valeur efficace.

6. Que proposez vous pour réaliser ce bruit blanc ? Une séquence binaire pseudo aléatoire !

7. Oui, l'une est la réponse indicielle (entrée = échelon), l'autre la réponse impulsionnelle (entrée = impulsion). Observez la pseudo-période, pour le gain, il faut un peu plus de calcul...

4 Filtrage des mesures

1. Que pouvez-vous faire pour utilisez au mieux ces données ?

Filtrer les entrées avec le même filtre !

2. Essayer de le démontrer.

Bonne chance ! (une démonstration est donnée dans le Ljung).

3. On observe facilement que 6 ou 8 zéros compensent pratiquement 6 ou 8 pôles. La prochaine tentative serai donc ARX 4 4 3 ou ARX 2 2 3

3^{ème} année
 Janvier 2007
 Option mécatronique

Examen d'Identification des systèmes

Tous documents autorisés

1 Modèle non-paramétrique (5 pts)

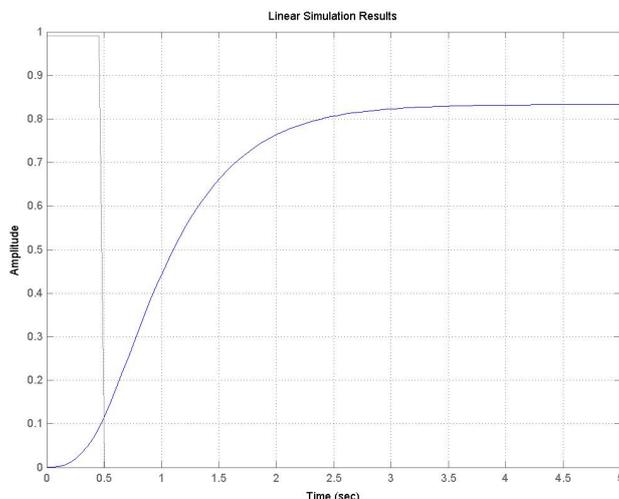


FIGURE 16 – Réponse du système à un créneau unité de durée $t = 0,5s$.

1. Identifier le système avec le méthode de Strejc, comme si la réponse était la réponse à un échelon.
2. Déduisez-en la fonction de transfert du système.

2 Modèle paramétrique (5 pts)

Un système sans bruit soumis à une rampe unité donne la sortie suivante.

k	0	1	2	3	4	5
u_k	0	1	2	3	4	5
y_k	0	0	7	7	42	-14

On se propose d'identifier un modèle paramétrique de la forme

$$y_k = -a_1 y_{k-1} + b_1 u_{k-1}^2$$

1. Ecrivez la relation précédente pour $k = 4$ et $k = 3$.
2. Montrez que ces deux équations peuvent se mettre sous une forme matricielle de la forme :

$$y = X\theta + e$$

3. Déterminez les coefficients a_1 et b_1 .

3 Moindres carrés récursifs (5 pts)

L'estimation de paramètres par la méthode des moindres carrés simples présente un inconvénient majeur, la nécessité de calculer l'inverse d'une matrice, ce qui est long et parfois impossible sur un microcontrôleur. On se propose à travers cet exercice de déterminer une forme récursive de cette estimation.

A l'instant N on peut écrire :

$$\begin{bmatrix} y_N \\ y_{N-1} \\ y_{N-2} \\ y_{N-3} \\ \vdots \\ y_{n+2} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} -y_{N-1} & -y_{N-2} & \cdots & -y_{N-n} & u_N & \cdots & u_{N-p} \\ -y_{N-2} & -y_{N-3} & & -y_{N-n-1} & u_{N-1} & & u_{N-p-1} \\ -y_{N-3} & \vdots & & \vdots & u_{N-2} & & u_{N-p-2} \\ \vdots & -y_{n+1} & & \vdots & \vdots & & u_{N-p-3} \\ -y_{n+1} & -y_n & & \vdots & \vdots & & u_{N-p} \\ -y_n & -y_{n-1} & \cdots & -y_1 & u_{n+1} & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix} + \begin{bmatrix} e_N \\ e_{N-1} \\ e_{N-2} \\ \vdots \\ e_{n+2} \\ e_{n+1} \end{bmatrix} \quad (1)$$

sous forme condensée :

$$Y_N = X_N \theta_N + E_N$$

dont l'estimation optimale est :

$$\hat{\theta}_N = (X_N^T X_N)^{-1} X_N^T Y_N$$

1. Ecrivez la forme (1) à l'instant $N + 1$ en utilisant

$$X_{N+1} = \begin{bmatrix} x_{N+1} \\ X_N \end{bmatrix}, \quad Y_{N+1} = \begin{bmatrix} y_{N+1} \\ Y_N \end{bmatrix} \quad \text{et} \quad E_{N+1} = \begin{bmatrix} e_{N+1} \\ E_N \end{bmatrix}$$

2. Donnez le vecteur ligne x_{N+1}
3. L'estimation à l'instant $N + 1$ s'écrit : $\hat{\theta}_{N+1} = (X_{N+1}^T X_{N+1})^{-1} X_{N+1}^T Y_{N+1}$, écrivez $\hat{\theta}_{N+1}$ en fonction de $X_N, x_{N+1}, Y_N, y_{N+1}$.

Note :

$$\begin{bmatrix} a \\ A \end{bmatrix}^T \begin{bmatrix} a \\ A \end{bmatrix} = A^T A + a^T a$$

4. Sachant que :

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

Montrez que :

$$\hat{\theta}_{N+1} = \left((X_N^T X_N)^{-1} - (X_N^T X_N)^{-1} x_{N+1}^T [1 + x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T]^{-1} x_{N+1} (X_N^T X_N)^{-1} \right) (X_N^T Y_N + x_{N+1}^T y_{N+1})$$

5. En posant $a = 1 + x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T$, montrez que :

$$\hat{\theta}_{N+1} = \hat{\theta}_N + (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} (y_{N+1} - x_{N+1} \hat{\theta}_N) \quad (2)$$

6. Que représente le terme $(y_{N+1} - x_{N+1} \hat{\theta}_N)$, interprétez alors l'équation (2)

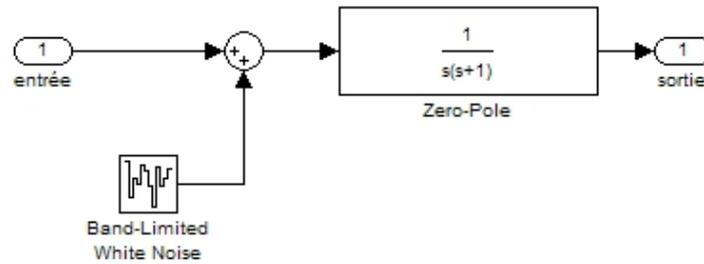


FIGURE 17 – Modèle issu d'un modèle de connaissances

4 Modélisation paramétrique (5 pts)

Un premier modèle issu d'un modèle de connaissance, est donné en figure 17.

1. Voyant que le système est intégrateur, comment procédez vous pour obtenir un modèle paramétrique du système ?
2. Déterminer les paramètres d'une séquence binaire pseudo aléatoire pour procéder à une identification paramétrique de ce système.
3. Quel modèle entre ARX et ARMAX donnera les meilleurs résultats d'après vous.

3^{ème} année
 Janvier 2006
 Option mécatronique

Corrigé de l'Examen d'Identification

1 Modèle non-paramétrique (5 pts)

1. Modèle Strejc

On mesure $Tu = 0,3s$ et $Ta = 1,3s$. En appliquant la méthode :

$$Tu/Ta = 0,2308 \Rightarrow n = 2$$

$$Ta/\tau = 2,718 \Rightarrow \tau = Ta/2,718 = 0,4783$$

$$Tu/\tau = 0,282 \Rightarrow Tu' = 0,1349$$

$$Tr = Tu - Tu' \Rightarrow Tr = 0,1651$$

$$\text{Gain} : 0.83$$

d'où le modèle :

$$G(p) = \frac{Ae^{-Trp}}{(1 + \tau p)^n} = \frac{0.83e^{-0,134p}}{(1 + 0,5887p)^2}$$

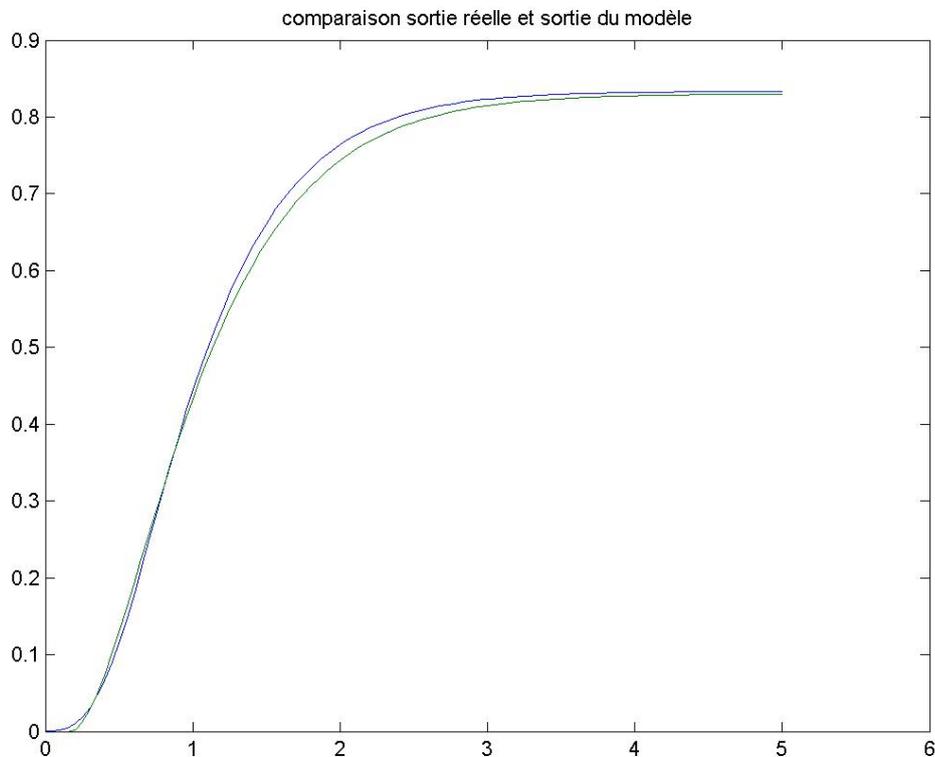


FIGURE 18 – comparaison sortie réelle et sortie du modèle.

2. On écrit l'égalité : réponse indicielle du modèle = réponse à un créneau du système.

$$G(p) \times \frac{1}{p} = G_2(p) \times \frac{1 - e^{-0.5p}}{p} \implies G_2(p) = \frac{Ae^{-Trp}}{(1 + \tau p)^n} \frac{1}{1 - e^{-0.5p}} \simeq \frac{2Ae^{-Trp}}{p(1 + \tau p)^n}$$

2 Modèle paramétrique (5 pts)

On se propose d'identifier un modèle paramétrique de la forme

$$\begin{bmatrix} 42 \\ 7 \end{bmatrix} = \begin{bmatrix} -7 & 9 \\ -7 & 4 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = y = X\theta + \underbrace{e}_{=0}$$

la solution est :

$$\hat{\theta} = (X^T X)^{-1} X^T y = \begin{bmatrix} 3 \\ 7 \end{bmatrix}$$

3 Moindres carrés récurrents (5 pts)

L'estimation de paramètres par la méthode des moindres carrés simples présente un inconvénient majeur, la nécessité de calculer l'inverse d'une matrice, ce qui est long et parfois impossible sur un microcontrôleur. On se propose à travers cet exercice de déterminer une forme récursive de cette estimation.

A l'instant N on peut écrire :

$$\begin{bmatrix} y_N \\ y_{N-1} \\ y_{N-2} \\ y_{N-3} \\ \vdots \\ y_{n+2} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} -y_{N-1} & -y_{N-2} & \dots & -y_{N-n} & u_N & \dots & u_{N-p} \\ -y_{N-2} & -y_{N-3} & & -y_{N-n-1} & u_{N-1} & & u_{N-p-1} \\ -y_{N-3} & \vdots & & \vdots & u_{N-2} & & u_{N-p-2} \\ \vdots & -y_{n+1} & & \vdots & \vdots & & u_{N-p-3} \\ -y_{n+1} & -y_n & & \vdots & \vdots & & u_{N-p} \\ -y_n & -y_{n-1} & \dots & -y_1 & u_{n+1} & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix} + \begin{bmatrix} e_N \\ e_{N-1} \\ e_{N-2} \\ \vdots \\ e_{n+2} \\ e_{n+1} \end{bmatrix} \quad (3)$$

sous forme condensée :

$$Y_N = X_N \theta_N + E_N$$

dont l'estimation optimale est :

$$\hat{\theta}_N = (X_N^T X_N)^{-1} X_N^T Y_N$$

1.

$$\begin{bmatrix} y_{N+1} \\ y_N \\ y_{N-1} \\ y_{N-2} \\ y_{N-3} \\ \vdots \\ y_{n+2} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} -y_N & -y_{N-1} & \dots & -y_{N-n+1} & u_{N+1} & \dots & u_{N-p+1} \\ -y_{N-1} & -y_{N-2} & \dots & -y_{N-n} & u_N & \dots & u_{N-p} \\ -y_{N-2} & -y_{N-3} & & -y_{N-n-1} & u_{N-1} & & u_{N-p-1} \\ -y_{N-3} & \vdots & & \vdots & u_{N-2} & & u_{N-p-2} \\ \vdots & -y_{n+1} & & \vdots & \vdots & & u_{N-p-3} \\ -y_{n+1} & -y_n & & \vdots & \vdots & & u_{N-p} \\ -y_n & -y_{n-1} & \dots & -y_1 & u_{n+1} & & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix} + \begin{bmatrix} e_{N+1} \\ e_N \\ e_{N-1} \\ e_{N-2} \\ \vdots \\ e_{n+2} \\ e_{n+1} \end{bmatrix} \quad (4)$$

2. $x_{N+1} = -y_N - y_{N-1} \dots - y_{N-n+1} u_{N+1} \dots u_{N-p+1}$

3.

$$\begin{aligned}\widehat{\theta}_{N+1} &= (X_{N+1}^T X_{N+1})^{-1} X_{N+1}^T Y_{N+1} \\ &= (X_N^T X_N + x_{N+1}^T x_{N+1})^{-1} (X_N^T Y_N + x_{N+1}^T y_{N+1})\end{aligned}$$

4. Il suffit de développer

$$\widehat{\theta}_{N+1} = \frac{\left((X_N^T X_N)^{-1} - (X_N^T X_N)^{-1} x_{N+1}^T [1 + x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T]^{-1} x_{N+1} (X_N^T X_N)^{-1} \right) (X_N^T Y_N + x_{N+1}^T y_{N+1})}{(X_N^T Y_N + x_{N+1}^T y_{N+1})}$$

5.

$$\begin{aligned}\widehat{\theta}_{N+1} &= \frac{\left((X_N^T X_N)^{-1} - (X_N^T X_N)^{-1} x_{N+1}^T [1 + x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T]^{-1} x_{N+1} (X_N^T X_N)^{-1} \right) (X_N^T Y_N + x_{N+1}^T y_{N+1})}{(X_N^T Y_N + x_{N+1}^T y_{N+1})} \\ \widehat{\theta}_{N+1} &= \underbrace{\left((X_N^T X_N)^{-1} - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} x_{N+1} (X_N^T X_N)^{-1} \right) (X_N^T Y_N + x_{N+1}^T y_{N+1})}_{\widehat{\theta}_N} \\ &= \underbrace{(X_N^T X_N)^{-1} X_N^T Y_N}_{\widehat{\theta}_N} + (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} x_{N+1} \underbrace{(X_N^T X_N)^{-1} X_N^T Y_N}_{\widehat{\theta}_N} \\ &\quad - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} \\ &= \widehat{\theta}_N + (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} x_{N+1} \widehat{\theta}_N \\ &\quad - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} \\ &= \widehat{\theta}_N - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(x_{N+1} \widehat{\theta}_N \right) \\ &\quad + (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(-x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} \right) + (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} \\ &= \widehat{\theta}_N - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(x_{N+1} \widehat{\theta}_N \right) \\ &\quad + (X_N^T X_N)^{-1} x_{N+1}^T \left(-a^{-1} x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T y_{N+1} + y_{N+1} \right) \\ &= \widehat{\theta}_N - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(x_{N+1} \widehat{\theta}_N \right) \\ &\quad + (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(-x_{N+1} (X_N^T X_N)^{-1} x_{N+1}^T + a \right) y_{N+1} \\ &= \widehat{\theta}_N - (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(x_{N+1} \widehat{\theta}_N \right) \\ &\quad + (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} y_{N+1} \\ &= \widehat{\theta}_N + (X_N^T X_N)^{-1} x_{N+1}^T a^{-1} \left(y_{N+1} - x_{N+1} \widehat{\theta}_N \right)\end{aligned}$$

6. Le terme $(y_{N+1} - x_{N+1} \widehat{\theta}_N)$ représente l'erreur d'estimation à l'aide des paramètres précédents. L'équation est alors : la nouvelle estimation est l'ancienne estimation corrigée par un terme proportionnel à l'erreur d'estimation précédente.

4 Modélisation paramétrique (5 pts)

1. Par un bouclage simple, suivi d'une identification.
2. Les paramètres dépendent nettement du choix fait pour le bouclage.
Par exemple, bouclage avec un gain $A=0,25 \implies$ deux pôles simples en $-0,5 \implies \tau \simeq 2s$

Pour une bonne identification du gain statique on choisira :

$$NTe = 3 \text{ à } 5 \tau$$

Pour une bonne excitation sur le spectre entre 0 et la plus haute fréquence de coupure on choisira :

$$0,3 \frac{1}{T_e} = \frac{1}{2\pi\tau}$$

Bon échantillonnage $\implies T_e \simeq 3,7s$ Bonne identification du gain statique $\implies n \simeq \frac{5\tau}{T_e} = 3$.

3. ARMAX donnera les meilleurs résultats.

Examen d'Identification des systèmes

Tous documents autorisés

Cet examen consiste à déterminer un modèle de la peau humaine soumise à un échelon de compression. La peau est un matériaux complexe (viscohyperélastique) très variable d'un individu à l'autre et même d'une région à l'autre sur un même individu. L'objectif est de déterminer un modèle de peau "local". Les valeurs numériques des différents modèles sont volontairement différentes. Les questions suivantes sont liées mais peuvent être traitées dans le désordre.

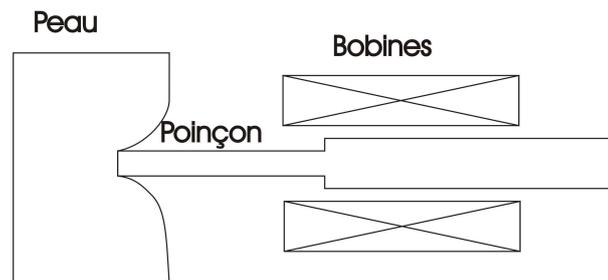


FIGURE 19 – Système étudié.

1 Modèle de connaissances (5 pts)

Un modèle simplifié de la peau est donné par la figure 20

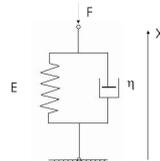


FIGURE 20 – Modèle de peau viscoélastique.

$$\sigma(t) = E\varepsilon(t) + \eta \frac{d\varepsilon(t)}{dt} \quad \text{avec :}$$

σ	=	contrainte
ε	=	déformation = $\varepsilon = \Delta x/x$
E	=	module d'Young équivalent
η	=	coefficient de frottement visqueux

le modèle de la force exercée par le poinçon sur la peau est donné par :

$$\frac{F(p)}{i(p)} = \frac{K_f}{1 + \tau p} \quad \text{avec :}$$

F	=	force
i	=	courant dans la bobine
$\tau = L/R$	=	constante de temps = inductance/résistance

Applications numériques : $E=300$ Mpa, $\eta/E = 100$ ms, $K_f = 10$, $\tau=1$ ms, épaisseur totale de la peau : 1mm, diamètre du poinçon : $\Phi=1$ mm.

- Déterminer la fonction de transfert du système $H(p) = \frac{x(p)}{i(p)}$.
- Déterminer les termes T, D, V et Q utilisés dans le formalisme de Lagrange.

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} + \frac{\partial V}{\partial q_i} = Q_i$$

2 Modèle non-paramétrique

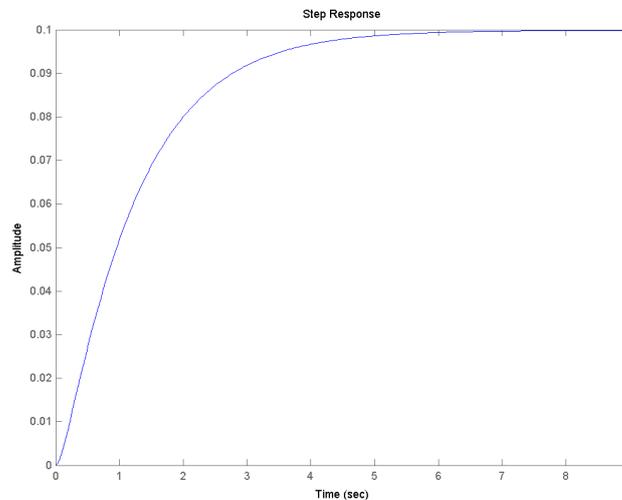


FIGURE 21 – Réponse indicielle de la peau.

1. Identifier le système avec la méthode de Broïda.
2. Identifier le système avec la méthode de Strejc.
3. Quel modèle vous semble le plus pertinent ?

3 Signal d'excitation

Notre système étant très sensible à la valeur moyenne du signal d'excitation, on décide d'utiliser une séquence ternaire pseudo aléatoire (STPA). Une méthode de synthèse de séquences ternaires pseudo aléatoires consiste à les déduire d'une SBPA par l'algorithme suivant :

$$y(i) = \frac{1}{2}(x(i) - x(i-1))$$

$$z = \{y(0), y(2), y(4), \dots, y(1), y(3), \dots\}$$

avec : x = séquence binaire pseudo aléatoire (valeurs : -1 ou 1)
 y = séquence intermédiaire
 z = séquence ternaire pseudo aléatoire

1. Représentez le diagramme temporel d'une séquence binaire pseudo aléatoire utilisant $n = 3$ bits soit de longueur $L = 2^n - 1$ en prenant $b_0 = b_2 \oplus b_1$.
2. En prenant une période d'échantillonnage $T_e = 100$ ms, quelles sont les constantes de temps τ_{min} et τ_{max} que l'on peut correctement identifier en utilisant la SBPA précédente ?
3. Avec la méthode proposée, déduisez-en une séquence ternaire pseudo aléatoire.
4. Quelle est la constante de temps τ_{max} que l'on est susceptible de bien identifier ?
5. Calculer la valeur moyenne de la STPA calculée précédemment.
6. Calculez sa fonction d'autocorrélation (! long!).

4 Modèle paramétrique

On décide de procéder à une identification paramétrique récursive. Le modèle contient 5 paramètres : a_1, a_2, a_3, b_0 et b_1 .

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}$$

1. Donner la taille des vecteurs et matrices suivants (notation du cours) : $\hat{\theta}_N, K_N, P_N$. Attention il y a des erreurs dans le poly!³
2. Le terme b_0 tends vers 0, que peut-on en déduire concernant le retard pur du système ?

5 Modèle d'ordre non entier

En fait il s'agit d'un modèle incluant une partie approchant un modèle d'ordre fractionnaire ($p^\alpha, \alpha \in \mathbb{R}$).

le modèle est de la forme :

$$H(z) = K \prod_{i=0}^n \frac{z - z_{zi}}{z - z_{pi}}$$

avec :

$$\begin{aligned} z_{p_{i+1}} &= z_{p_i} * \lambda \\ z_{z_i} &= z_{p_i} * \nu \\ z_{p_0} &= z_0 \\ \nu &< \lambda \end{aligned}$$

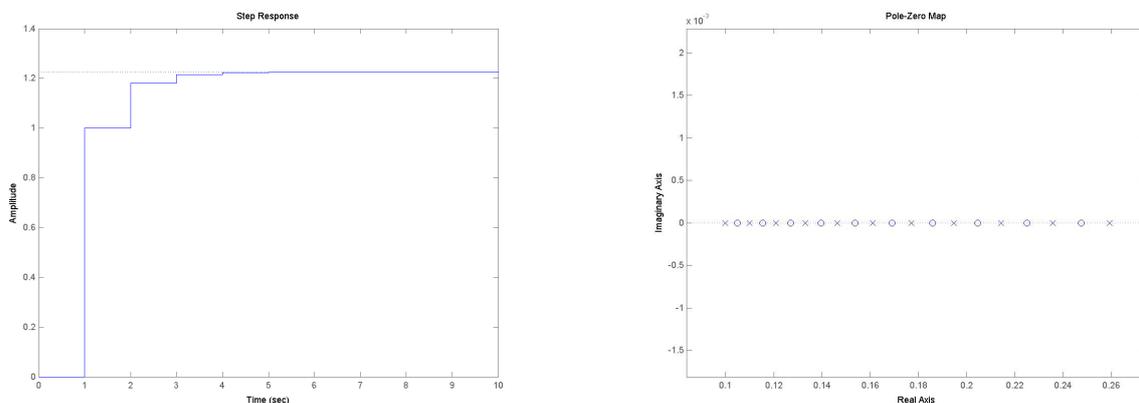


FIGURE 22 – Caractéristiques d'un modèle d'ordre fractionnaire : réponse indicielle et pôles et zéros.

1. Proposez une méthode pour déterminer les paramètres λ, ν, z_0 et K optimums vis-à-vis d'un critère que vous définirez.

Liens

<http://www.mathworks.com/products/fuzzylogic.html>

3. désolé!

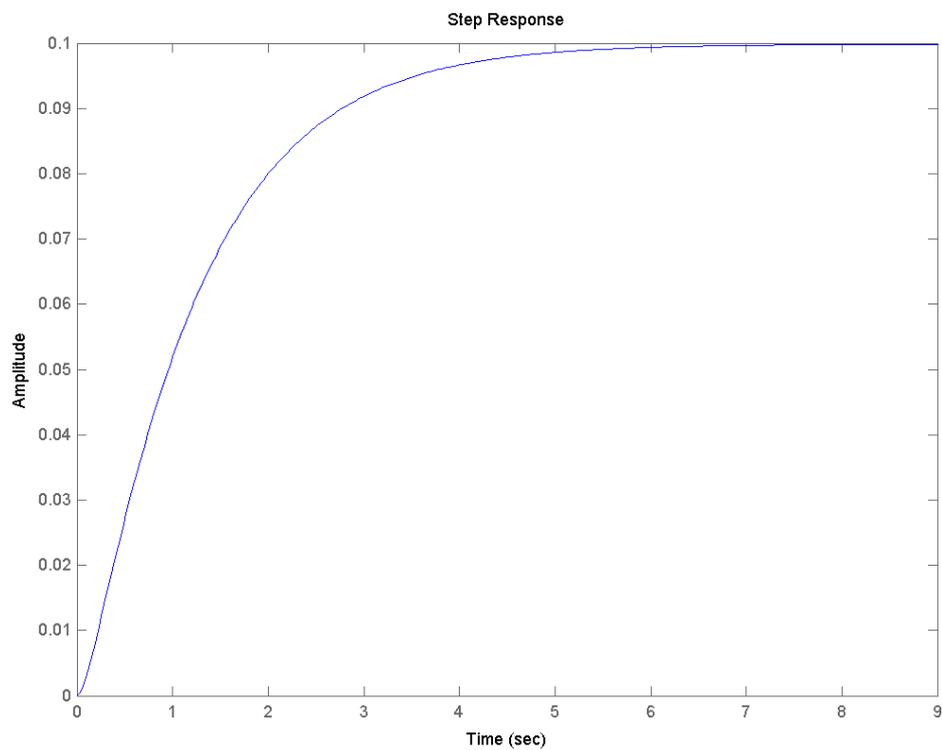


FIGURE 23 – Réponse indicielle de la peau.

http://www.mathtools.net/MATLAB/Fuzzy_Logic/index.html

<http://www.dbai.tuwien.ac.at/marchives/fuzzy-mail/>

<http://europa.eu.int/comm/research/rtdinfo/fr/24/02.html>

<http://www.gala.univ-perp.fr/~polit/chap0.html>

<http://www.eru.ulaval.ca/ptt15225>

http://vcampus.u-strasbg.fr/uticeweb/mhiri/projet_V2

<http://www.mathworks.fr/products/controldesign/modanal.shtml>

Bibliographie