

EPFL-SECTION DE GENIE ELECTRIQUE ET ELECTRONIQUE
LABORATOIRE D'AUTOMATIQUE

Projet 8ème semestre

FEEDFORWARD CONTROL WITH PREVIEW
FOR A LINEAR MOTOR

Christian Künzler

Collaborateurs responsables: Mark Butcher, Alireza Karimi

année 2004 / 2005
semestre d'été

CONTENU

| | |
|--|-----------|
| 1 Introduction | 3 |
| 1.1 But du Projet..... | 3 |
| 1.2 Le système pour l'application des méthodes..... | 3 |
| 1.2.1 Description du Système..... | 4 |
| 1.2.2 Implémentation du régulateur RST..... | 5 |
| 2 Partie théorique | 7 |
| 2.1 Description du preview..... | 7 |
| 2.2 Première méthode (Model-free)..... | 9 |
| 2.2.1 La méthode..... | 9 |
| 2.2.2 Critère pour le choix de l'ordre de T et du preview..... | 11 |
| 2.3 Deuxième méthode (basé sur un modèle)..... | 12 |
| 2.3.1 Méthode originale..... | 12 |
| 2.3.2 Méthode légèrement modifiée..... | 16 |
| 2.4 Calcul de « R » et « S » à partir d'un régulateur PID..... | 18 |
| 3 Résultats | 20 |
| 3.1 Consignes..... | 20 |
| 3.2 Modèle utilisé – problème de conversion..... | 21 |
| 3.3 Analyse des régulateurs PID..... | 22 |
| 3.4 résultats de la Première méthode (model-free)..... | 24 |
| 3.4.1 Simulations..... | 24 |
| 3.4.2 Expériences..... | 25 |
| 3.5 Deuxième méthode..... | 27 |
| 3.5.1 Simulations..... | 27 |
| 3.5.2 Expériences..... | 29 |
| 4 Conclusion | 30 |
| 5 Annexes | 32 |
| 5.1 Bibliographie..... | 32 |
| 5.2 CD-Rom du Travail..... | 33 |

1 INTRODUCTION

1.1 BUT DU PROJET

Le but de ce projet est d'optimiser un régulateur par deux méthodes différentes: Une qui n'utilise qu'une seule mesure et qui n'a notamment pas besoin d'un modèle du système et une autre qui se base justement sur un modèle du système à régler. Les méthodes seront comparées par simulation, mais également par voie expérimentale sur un système réel. Afin de juger sur les améliorations introduites, les résultats seront comparés à un simple régulateur PID qui est utilisé momentanément sur le système.

Tous les deux méthodes implémentent le « preview »: Le signal d'entrée est envoyé en avance pour compenser le retard pur du système. Par contre, le sujet de ce travail a été légèrement modifié et aucune des deux méthodes n'a été appliquée à une commande à priori (« feedforward »). Il existe déjà une variante de la première méthode qui s'appliquerait à une commande à priori [2], par contre elle devrait être trouvée pour la deuxième méthode. Malheureusement, le temps n'a pas suffi à faire ceci, mais la simulation et l'application des deux processus d'optimisation à des régulateurs RST classiques (mais avec preview) montrent déjà les pouvoirs de ces méthodes et donnent ainsi une indication de ce qu'on peut s'attendre pour des commandes à priori.

Ce rapport est organisé de la manière suivante:

- Une introduction contenant la description du système sur lequel les méthodes ont été testées précède ...
- ... une partie théorique qui explique les méthodes et la notion de preview.
- Finalement des résultats de simulation et d'expériences sont donnés.

Une conclusion terminera le présent document pour résumer les résultats et indiquer les points forts et faibles de ces deux méthodes.

1.2 LE SYSTÈME POUR L'APPLICATION DES MÉTHODES

Aucune identification n'a été faite dans le cadre de ce projet et des modèles déjà existant ont été utilisés. Il n'est donc pas nécessaire à connaître en détail le système sur laquelle on a fait les expériences pratiques. Le lecteur qui veut en savoir plus est renvoyé aux documents [9] et [8].

Cependant, il est utile de rappeler les grandes principes afin de mieux comprendre les essais et la démarche. En plus, l'implémentation du régulateur RST a été changée et elle sera donc décrite dans ce chapitre.

1.2.1 DESCRIPTION DU SYSTÈME

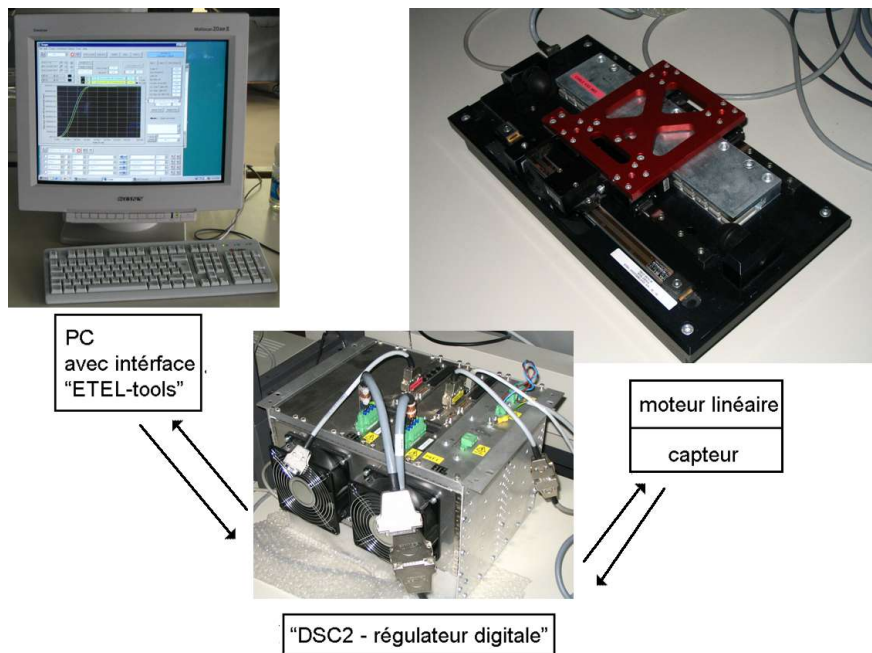


figure 1: les communications entre les différents éléments du système

L'élément à régler est un moteur linéaire fabriqué par l'entreprise ETEL. Ce moteur est de type « synchrone sans fer » avec des performances très élevées:

- Une accélération maximale jusqu'à 20G ou 200 m/s², selon le modèle.
- Une plage de vitesse qui va de 0.01mm/s jusqu'à 10m/s
- Une grande répétitivité: Pour la même consigne, le moteur va toujours trouver la même position (10µm à 100µm d'erreur selon le modèle)
- Une force du moteur importante

Le moteur est connecté à un capteur qui mesure la position en incréments (1 m = 204'800'000 incr.).

Un module électronique portant le nom « DSC2-régulateur digitale » s'occupe de la génération de la consigne, de la régulation (tous les coefficients du régulateur sont téléchargés dans cette carte) et de la génération du courant triphasé. Grâce à la carte, il est également possible de garder les dernières 1024 valeurs de différents signaux pour analyser les mouvements et les régulateurs utilisés.

Installé sur un ordinateur, l'interface « ETEL-tools » permet finalement à l'utilisateur de facilement communiquer avec la carte « DSC2 »: Il est possible d'envoyer des commandes telles qu'un changement de position ou des nouvelles caractéristique pour les consignes (cf « 3.1 Consignes »), mais également d'ajuster les paramètres du régulateur et d'analyser les différents signaux (valeurs théoriques et réelles de position, sortie du régulateur, sortie après le réseau de saturation, ...)

1.2.2 IMPLÉMENTATION DU RÉGULATEUR RST

Le système d'expérience permet un choix entre un régulateur RST et un régulateur PID. Il suffit de mettre simplement la bonne valeur dans le registre K79 de la carte « DSC 2 » via l'interface « ETEL-tools »:

- K79 = 0 => le régulateur PID est utilisé
- K79 = 90 => le régulateur RST est utilisé

Si on veut changer les caractéristiques du régulateur, il faut changer les coefficients des polynômes qui sont chacun sauvegardé dans un registre bien précis. Le régulateur RST n'a pas bien marché au début du projet et l'entreprise ETEL a installé une nouvelle firmware sur la carte « DSC2 ». C'est pour ça que les registres contenant les coefficients du régulateur RST ont changé par rapport au travaux effectués précédemment (Tschirren [7], Deschenaux [8] et Bays [9]).

Pour éliminer toutes les sources de malentendu, les polynômes et les registres associés sont décrits dans ce sous-chapitre. Commençons par la définition des polynômes tels qu'ils sont utilisés sur le système réel:

$$R(q^{-1}) = r_0 + r_1 \cdot q^{-1} + \dots + r_{10} \cdot q^{-10}$$

$$S(q^{-1}) = s_0 + r_1 \cdot q^{-1} + \dots + s_{10} \cdot q^{-10}$$

$$T(q^{-1}) = t_0 + t_1 \cdot q^{-1} + \dots + t_{20} \cdot q^{-20}$$

Le polynôme R est monique, ça veut dire que r_0 est égale à 1. Il est important de noter qu'il faut multiplier les polynômes ci-dessus avec le facteur 2^{16} avant de les télécharger à la carte « DSC2 », parce-que les valeurs seront divisées par ce même facteur dans la carte mentionnée (cf [7] pour plus d'informations sur la nature de cette division).

Les registres suivants sont associés aux polynômes R, S et T:

- K300 à K309: polynôme R
- K310 à K319: polynôme S
- K320 à K339: polynôme T

A cause de l'implémentation du régulateur, il est en plus nécessaire d'inverser les coefficients r_1 à r_9 de R.

Le tableau suivant résume toutes ces propriétés:

| Registre | Valeur à mettre | Commentaire |
|----------|------------------------|--|
| K300 | $r_0 * 2^{16} = 65536$ | Toujours mettre cette valeur dans ce registre. |
| K(300+i) | $-r_i * 2^{16}$ | $i=1...9$ |
| K(310+i) | $s_i * 2^{16}$ | $i=0...9$ |
| K(320+i) | $t_i * 2^{16}$ | $i=0...19$ |

En plus, il est possible de choisir la période d'échantillonnage du régulateur en modifiant la valeur dans K345. Si la valeur dans ce dernier registre vaut zéro, la fréquence d'échantillonnage est 24kHz, sinon elle est 24kHz divisé par la valeur dans ce registre: $f = 24\text{kHz} / (K345)$.

Lors d'un changement des coefficients, les valeurs modifiées se trouvant dans les registres décrits ci-dessus ne sont pas tout de suite chargées dans le régulateur afin d'éviter qu'un régulateur instable est appliqué. Une fois qu'on a effectué toutes les modifications nécessaires, et que le régulateur est stable, il faut déclencher le chargement. Ceci est fait en écrivant une valeur quelconque dans le registre K340.

2 PARTIE THÉORIQUE

Après un sous-chapitre consacré à la notion du « preview » et à son implémentation, les deux méthodes utilisées pour l'optimisation du polynôme T seront décrites. Elles sont tous les deux très bien documentées dans la littérature. Les références des documents sont mentionnées. C'est pour ça que je ne décris ici pas de détails, et surtout pas les preuves des théorèmes utilisés, mais je donne une brève description qui résume les méthodes et permet au lecteur non initié de facilement comprendre la base théorique des méthodes.

Ce chapitre décrit également les petites modifications qui ont été faites pour pouvoir appliquer les méthodes au système réel.

2.1 DESCRIPTION DU PREVIEW

Le preview, ou autrement dit la connaissance de la consigne en avance, est un des points principaux de ce travail et utilisé dans tous les deux méthodes. Il est donc important de mettre bien au clair cette notion et d'en citer les avantages.

Parce-que ni Matlab ni le système utilisé supporte des polynômes non causals, il est également important de décrire l'implémentation du preview.

Intérêt du preview:

La plupart des systèmes (= ensemble moteur et régulateur) contient un retard pur. Ceci est particulièrement vrai pour des systèmes utilisant un régulateur numérique: La numérisation nécessite un peu de temps et introduit ainsi un retard d'une période d'échantillonnage. Il va donc de soi qu'aucun régulateur classique (ça veut dire causal) ne peut suivre exactement la consigne: Même un régulateur d'ordre infinie ne pourrait pas compenser le retard!

Mais dans beaucoup de cas la consigne est bien connu à l'avance, et il est donc possible de compenser le retard du système simplement en envoyant la consigne avant que la sortie est supposée de prendre la valeur de la consigne. Les prochaines figures illustrent ce cas:

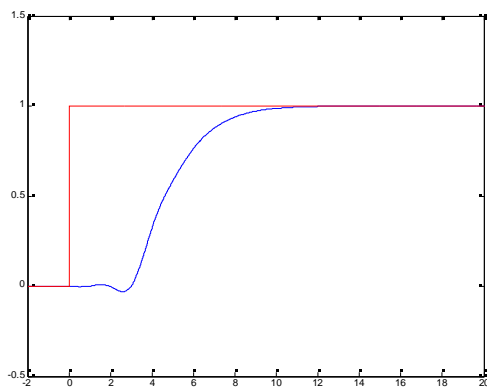


figure 2: sans preview.

rouge: consigne – bleu: réponse

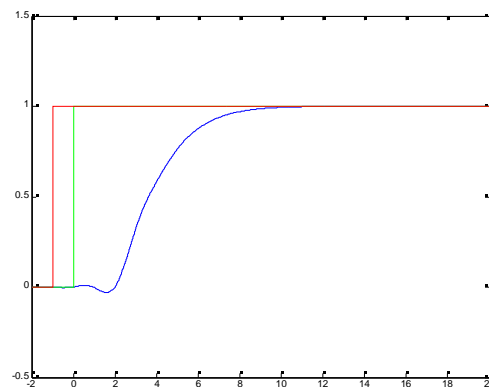


figure 3: avec preview

rouge: consigne avancé – vert: consigne originale –
bleu: réponse

Le régulateur RST classique (sans preview) est décrit à l'aide de l'équation suivante

$$R(q) \cdot u(k) = T(q) \cdot y_r(k) - S(q) \cdot y(k) \quad (1)$$

où $k \in \mathbb{Z}$ est l'instant discret de temps, $y_r(k)$ est la consigne, $y(k)$ est la sortie du système et $u(k)$ est le signal de commande.

L'introduction d'un preview modifie légèrement la consigne de la dernière équation: Comme il a été illustré dans les figures ci-dessus, la consigne est avancé de γ_0 périodes d'échantillonnages pour compenser le retard du système:

$$R(q) \cdot u(k) = T(q) \cdot \{q^{\gamma_0} \cdot y_r(k)\} - S(q) \cdot y(k) \quad (2)$$

Le terme q^{γ_0} peut être inclut dans le polynôme T: $T_{preview} = T(q) \cdot q^{\gamma_0}$ Il faut noter que ce nouveau polynôme n'est donc généralement plus causal.

Les deux méthodes proposent d'introduire un nombre de preview supplémentaire au retard pur du système: Il n'est donc pas uniquement utilisé pour compenser ce retard pur, mais il peut aussi aider (avec un polynôme T adéquat) de mieux suivre la forme de la consigne. En effet, un changement brusque de la consigne peut être détecté en avance par le régulateur et il peut donc tout de suite réagir. On écrit: $T_{preview} = T(q) \cdot q^{\gamma_0 + \gamma_1}$ où γ_1 dénote le nombre de preview supplémentaire. L'équation (2) devient:

$$R(q) \cdot u(k) = T(q) \cdot q^{\gamma_0 + \gamma_1} \cdot y_r(k) - S(q) \cdot y(k) \quad (3)$$

Remarque: Dans la suite de ce rapport, on notera simplement T au lieu de $T_{preview}$, même si le polynôme T n'est pas causal.

Implémentation du preview:

Ni Matlab ni le système supporte les régulateurs non causals, on ne peut donc pas implémenter l'équation (3) directement.

La première idée pour contourner ce problème consiste à multiplier l'équation (3) par

$$q^{-(\gamma_0 + \gamma_1)}. \text{ On trouve: } q^{-(\gamma_0 + \gamma_1)} R(q) \cdot u(k) = T(q) \cdot y_r(k) - q^{-(\gamma_0 + \gamma_1)} S(q) \cdot y(k)$$

Malheureusement, le polynôme R n'est plus monique. Il est donc impossible d'utiliser cette technique.

L'équation (3) ne peut donc pas être utilisé directement pour la réalisation du régulateur (sans changer le système de façon à ce qu'il soit capable de fournir la consigne en avance et de traiter des régulateurs non causaux). Pour quand-même pouvoir tester les méthodes et l'amélioration qui peut être atteint avec le preview, il faut donc implémenter l'équation (2) au lieu de (3), comme si on n'avait pas de preview. Après, pour calculer l'erreur entre la consigne et la sortie du système, on tiendra compte du preview en simplement retardant la consigne:

$$erreur = y(k) - q^{-(\gamma_0 + \gamma_1)} \cdot y_r(k) \quad (4)$$

L'analyse des résultats ne peut se faire donc pas directement sous ETEL-tools, mais on doit retarder la consigne après les mesures dans matlab.

2.2 PREMIÈRE MÉTHODE (MODEL-FREE)

La première méthode a été développée pour un régulateur quelconque avec un précompensateur. Elle peut être facilement utilisée pour un régulateur RST, où la partie T prend le rôle du précompensateur. Contrairement au document [2], je vais ici écrire les équations pour un régulateur RST. Ils ne diffèrent pourtant que pour le nom des différentes parties.

2.2.1 LA MÉTHODE

Regardons d'abord le schéma d'un régulateur RST:

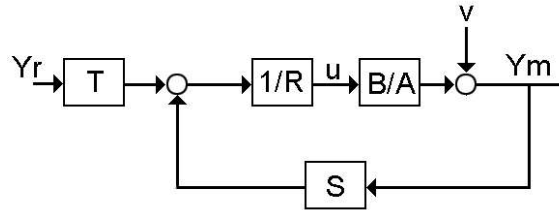


figure 4: régulateur RST

y_r est la consigne, $P=B/A$ le système (moteur), v le bruit, et y_m la sortie mesurée. Écrivons l'équation du polynôme T :

$T = t_0 \cdot q^\delta + t_1 \cdot q^{\delta-1} + \dots + t_n \cdot q^{\delta-n}$ où δ est un entier positif et dénote le nombre de preview (en périodes d'échantillons). n est le degré de T . On va associer un vecteur au polynôme T qui contient les coefficients: $\bar{t}^T = (t_0 \quad t_1 \quad \dots \quad t_n)$

Définition du critère:

Il est clair que si le régulateur suivait parfaitement la consigne en absence d'un bruit v , alors l'erreur mesurée ne contiendrait que la contribution de ce dernier. En d'autres mots, l'erreur n'est pas dépendant de la consigne et donc pas corrélé avec elle. Même si dans la pratique le régulateur n'est jamais parfait, la corrélation peut être utilisée comme critère d'optimisation. On va donc définir ce critère basé sur une fonction de corrélation:

$$f(\bar{t}) = E\{[y_r(k) - y(k)]\zeta(k)\} = E\{e(k) \cdot \zeta(k)\} \quad (5)$$

où $e(k)$ est l'erreur entre la sortie et l'entrée, E l'espérance mathématique et ζ un vecteur utilisé pour calculer la corrélation:

$$\zeta(k)^T = [y_r(k+nz), \dots, y_r(k), \dots, y_r(k-nz)] \quad (6)$$

Sa longueur vaut $l=2*nz+1$ et doit être supérieure à l'ordre du système P . On peut maintenant définir le critère:

$$J(\bar{t}) = \|f(\bar{t})\|^2 = f^T(\bar{t}) \cdot f(\bar{t}) \quad (7)$$

Dans la pratique, on n'a pas une infinité de données et on ne peut pas trouver l'espérance exacte (5). Une bonne approximation est:

$$\hat{f}(\bar{t}) = \frac{1}{N} \cdot \sum_{k=1}^N e(k) \cdot \zeta(k) \quad (8)$$

N est le nombre de données et doit être supérieure à la longueur l du vecteur ζ .

On obtient pour l'approximation du critère:

$$J_N(\bar{t}) = \hat{f}^T(\bar{t}) \cdot \hat{f}(\bar{t}) \quad (9)$$

Algorithme pour calculer un T optimal:

Pour calculer le T optimal, on a d'abord besoin de données: On va donc légèrement modifier le régulateur RST classique:

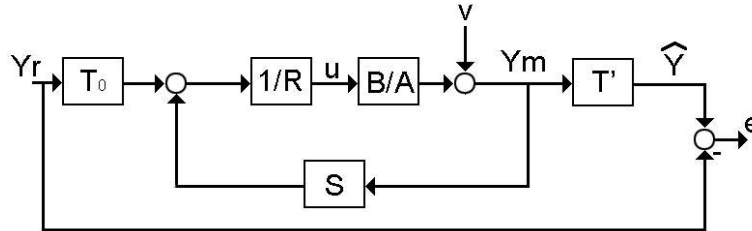


figure 5: schéma d'optimisation

T_0 est un polynôme de degré zéro afin de ne pas introduire des zéros qui ne sont pas optimisés. Son rôle est d'assurer le gain statique =1. Donc, $T_0 = S(z=1)$. Le précompensateur T' est maintenant placé à la fin de la boucle. Il est ensuite ajusté pour minimiser le critère de corrélation. Une fois qu'il a été calculé, on retrouve le schéma classique (figure 4) avec $T = T_0 \cdot T'$.

On va donc exciter le système décrit dans la figure ci-dessus par le signal d'entrée pour lequel on veut optimiser le régulateur et mesurer la sortie y_m . Le signal à la sortie du précompensateur T' (qui est plutôt un postcompensateur dans ce schéma) se calcule facilement:

$\hat{y} = y_m \cdot T'(\bar{t})$ et on trouve pour l'erreur $\epsilon_t = y_r - \hat{y} = y_r - y_m \cdot T'(\bar{t}) = y_r(k) - \phi^T(k) \cdot \bar{t}$ avec Φ un vecteur contenant des mesures:

$$\phi^T(k) = [y_m(k+\delta), y_m(k+\delta-1), \dots, y_m(k-n+\delta)]$$

Ceci mène à l'approximation de la fonction de corrélation suivante:

$$\hat{f}(\bar{t}) = \frac{1}{N} \sum_{k=1}^N \zeta(k) \cdot [y_r(k) - \phi^T(k) \cdot \bar{t}] = Z - Q \cdot \bar{t} \quad (10)$$

avec

$$Z = \frac{1}{N} \cdot \sum_{k=1}^N \zeta(k) \cdot y_r(k) \quad (11) \quad \text{et} \quad Q = \frac{1}{N} \cdot \sum_{k=1}^N \zeta(k) \cdot \phi^T(k) \quad (12)$$

\bar{t} se calcule maintenant selon l'algorithme des moindres carrés:

$$\bar{t} = (Q^T \cdot Q)^{-1} \cdot Q^T Z \quad (13)$$

Remarque: Si on calcule $\hat{y} = y_m \cdot T'$, le signal serait beaucoup bruité parce-que le bruit qui intervient dans les mesures (avant y_m) est amplifié par le précompensateur T' . Dans le montage final où le précompensateur est vraiment au début du régulateur, le bruit intervient complètement à la fin du régulateur et il n'est donc pas renforcé par celui-ci. Le bruit ne fausse pas les calculs du précompensateur dans la mesure où il est décorrélé avec l'entrée y_r .

2.2.2 CRITÈRE POUR LE CHOIX DE L'ORDRE DE T ET DU PREVIEW

Il est souvent important d'implanter un régulateur d'ordre minimal avec des performances élevées. Pour trouver le régulateur qui fait un bon compromis entre ces deux qualités opposées, on doit se servir d'un critère pour juger les différents régulateurs. Quel critère?

Comme il a été mentionné dans le paragraphe précédent, il n'est pas possible de simuler le régulateur optimisé à cause du bruit qui intervient avant le précompensateur. Il n'est donc pas possible de simuler l'erreur et de juger le régulateur sur la base de la norme de l'erreur, comme il va être le cas pour la deuxième méthode. Il faut donc trouver un autre critère.

La méthode minimise l'intercorrélation entre l'entrée et l'erreur, il est donc judicieux à définir un critère basé sur cette valeur. Dans la théorie de l'identification ils existent déjà de tels critères. On trouve par exemple dans le document [6] (page 167) le critère suivant:

$$\left| \frac{R_{y_r, e}(h)}{\sqrt{R_{e, e}(h) \cdot R_{y_r, y_r}(h)}} \right| \leq \frac{1.96}{\sqrt{N}} \quad (14)$$

où $R_{y_r, e}$ est l'intercorrélation approximé entre l'erreur e et la consigne y_r et $R_{e, e}$ et R_{y_r, y_r} sont les autocorrélations de e et y_r respectivement. N est le nombre d'estimation. Ce critère test donc la valeur de l'intercorrélation relatif à une distribution normale avec une intervalle de confiance de 95%.

Dans la littérature, on peut aussi trouver d'autres critères pour des méthodes de minimisation de corrélation. Dans [5] par exemple, le critère suivant est utilisé:

$$\bar{J} \leq \frac{P_r}{N} \cdot \chi^2(\alpha, L) \quad (15)$$

où $P_r = \sum_{\tau=-nz}^{+nz} R_{e, e}(\tau) \cdot R_{y_r, y_r}(\tau)$ et $L=2*nz+1$. Le critère à minimiser \bar{J} est dans le cas du document mentionné. $J_N(\bar{t}) = E(\hat{f}^T(\bar{t}) \cdot \hat{f}(\bar{t}))$ et donc différent du critère J utilisé dans cette méthode. Mais on sait que $\bar{J} \leq J$ et on peut donc quand-même essayer d'appliquer le test qui serait alors un peu moins stricte que pour le critère \bar{J} .

2.3 DEUXIÈME MÉTHODE (BASÉ SUR UN MODÈLE)

Référence: Cette méthode a été développée par D.Hoover et est décrite entièrement dans le document [3] et dans une forme résumée dans le document [4].

Introduction: Contrairement à la première méthode, celle-ci utilise un modèle identifié. La méthode originale utilise également la transformée en z de la consigne. Vu que je n'ai pas travaillé avec des consignes simples mais avec des S-courbes (cf chapitre 3.1), cette transformée n'est pas facile à trouver et d'ordre élevée. Il est possible de trouver des approximations en utilisant les méthodes d'identifications, mais ceci introduit des imperfections supplémentaires (déjà le modèle n'est qu'une approximation). Dans la seconde partie de ce chapitre, une légère modification a été introduite pour ne plus avoir besoin de cette transformée en z et pour pouvoir utiliser directement la consigne dans le domaine temporel.

2.3.1 MÉTHODE ORIGINALE

La méthode se divise dans différentes étapes:

- Calcul des polynômes R et S selon des méthodes classiques
- Calcul d'un polynôme T_0' qui sert comme polynôme de départ pour l'optimisation
- Définition du critère d'optimisation
- Optimisation du polynôme T_0'

Calcul des polynômes R et S selon des méthodes classiques:

Le calcul de ces deux polynômes est décrit en détail dans l'ouvrage [1] et il est également résumé dans le document [4]. Il est quand-même utile de décrire cette partie pour bien comprendre la suite.

Supposons que le système a été identifié et est décrit par le quotient de deux polynômes:

$\frac{B(q)}{A(q)}$ Ce modèle est strictement propre. L'équation qui décrit la sortie en fonction de la

consigne (boucle fermée) est:

$$y(k) = \frac{B \cdot T}{A \cdot R + B \cdot S} \cdot y_r(k) \quad (16)$$

On cherche à simplifier les zéros de B dans la mesure où il est possible sans aucun risque d'introduire des pôles instables ou pas assez amortis. Il faut donc séparer les zéros instables où pas assez amortis de B des zéros stables: $B = B^{stable} \cdot B^{instable}$. Si on inclut maintenant le terme B^{stable} dans le polynôme R en écrivant $R = R_1 \cdot B^{stable}$, il est possible de récrire l'équation de la boucle fermée (16):

$$y(k) = \frac{B^{instable} \cdot T}{A \cdot R_1 + B^{instable} \cdot S} \cdot y_r(k) \quad (17)$$

Si en plus on veut imposer des termes fixes dans les polynômes R et S (par exemple un intégrateur dans R et un terme $(1+z)$ dans S pour limiter le gain du bruit en haut fréquences, cf [7] pour les détails du choix de ces deux termes), on va poser:

$R = R_f R_1' \cdot B^{stable}$ et $S = S_f S'$ où les polynômes R_f et S_f contiennent ces termes fixes. L'équation (2) en boucle fermée s'écrit alors:

$$y(k) = \frac{B^{instable} \cdot T}{A \cdot R_f R_1' + B^{instable} \cdot S_f S'} \cdot y_r(k) \quad (18)$$

Les pôles de la boucle fermée sont donnés par le dénominateur de l'équation (18). Grâce aux polynômes R_f et S_f les pôles du modèle identifié peuvent être déplacé dans n'importe quelle région et en plus d'autres pôles peuvent être ajoutés. Supposons que le polynôme A_m contient les pôles souhaités de la boucle fermée. On peut donc poser l'équation suivante:

$$A \cdot R_f R_1' + B^{instable} \cdot S_f S' = A_m \cdot A_0 \quad (19)$$

où A_0 est un polynôme d'observateur dépendant des degrés des autres polynômes. Il est introduit pour pouvoir résoudre l'équation ci-dessus et va être éliminé par le polynôme T en posant $T = T' \cdot A_0$. L'équation (19) est une équation de Diophante avec les inconnus

R_1' et S' . L'ouvrage [1] donne des conditions précises pour les choix des degrés de ces polynômes. Notons qu'une condition d'existence de solutions de cette équation est également donnée dans l'ouvrage [1] mais que la solution n'est jamais unique. Il est désormais possible de trouver une solution qui satisfait la condition de causalité $\delta(R) \geq \delta(S)$

En combinant l'équation (19) avec l'équation (18) et en tenant compte du polynôme d'observateur dans T , l'équation pour la sortie de la boucle fermé devient:

$$y(k) = \frac{B^{instable} \cdot T'}{A_m} \cdot y_r(k) \quad (20)$$

Calcul d'un polynôme T_0' qui sert comme polynôme de départ pour l'optimisation:

L'optimisation cherche à minimiser l'erreur (entre la consigne et la sortie) pour une consigne particulière. Si la consigne $y_r(k)$ est un signal élémentaire, son transformé en z peut facilement être trouvé. Pour un saut d'unité par exemple, elle vaut $\frac{q}{q-1}$.

En générale, la transformé en z d'une consigne est donné par un quotient de deux polynômes: $\frac{B_c(q)}{A_c(q)}$. Dans le domaine temporel, la consigne devient:

$$y_r(k) = \frac{B_c(q)}{A_c(q)} \cdot \delta(k) \quad \text{où } \delta(k) \text{ est l'impulsion unitaire (égale à 1 à l'instant zéro, égale à 0$$

sinon). L'erreur peut donc être calculé en utilisant l'équation (20):

$$e(k) = q^{-(\gamma_0 + \gamma_1)} y_r(k) - y(k) = \left(q^{-(\gamma_0 + \gamma_1)} - \frac{B^{stable} \cdot T'}{A_m} \right) \cdot \frac{B_c(k)}{A_c(k)} \cdot \delta(k)$$

et après simplifications:

$$e(k) = \frac{(A_m - q^{\gamma_0 + \gamma_1} \cdot B^{stable} \cdot T') \cdot B_c}{q^{\gamma_0 + \gamma_1} \cdot A_m \cdot A_c} \cdot \delta(k) \quad (21)$$

Pour minimiser l'erreur, il faut donc trouver un T' qui minimise l'équation (21). Si on arrivait à trouver un T' qui vérifierait l'équation $q^{\gamma_0 + \gamma_1} \cdot B^{stable} \cdot T' = A_m$, alors l'erreur serait égale à zéro, et ceci même pour tous les signaux de consignes! Malheureusement, on a presque jamais les moyens pour le faire: En effet, le signal de commande deviendrait beaucoup trop grande et saturait dans la réalité. C'est la raison pour laquelle cette méthode n'utilise pas simplement l'erreur comme critère d'optimisation, mais également l'amplitude du signal de commande, comme nous allons le voir dans la troisième étape.

Similaire à la première partie de la méthode, où on voulait simplifier les zéros stables de B , on veut simplifier les pôles instables de A_c dans l'équation (21). On va donc séparer les zéros stables et bien amortis des zéros instables où pas assez amortis de la consigne:

$A_c = A_c^{stable} \cdot A_c^{instable}$. Si le dénominateur de l'équation (21) vérifie

$$A_c^{instable} \cdot P + q^{\gamma_0 + \gamma_1} \cdot B^{instable} \cdot T' = A_m \quad (22)$$

avec les inconnues P et T' , alors l'équation (21) peut être simplifiée et donne:

$$e(k) = \frac{P \cdot B_c}{q^{\gamma_0 + \gamma_1} \cdot A_m \cdot A_c^{stable}} \cdot \delta(k) \quad (23)$$

Et le signal de commande s'écrit:

$$u(k) = \frac{A \cdot T'}{B^{stable} \cdot A_m} \cdot y_r(k) \quad (24)$$

L'équation (22) est une deuxième équation de Diophante et donne des valeurs pour T' et P . De nouveau, ces solutions ne sont pas uniques. La totalité des solutions peut être décrit à l'aide des deux équations suivantes:

$$T' = T'_0 - A_c^{instable} \cdot Q \quad P = P_0 + q^{\gamma_0 + \gamma_1} \cdot B^{instable} \cdot Q \quad (25)$$

où Q est un polynôme quelconque dont le degré vérifie l'équation

$\delta(Q) \leq m - \delta(A)$ où $m \geq \delta(T')$. Les polynômes T'_0 et P_0 sont les uniques polynômes qui vérifient à la fois l'équation (22) et la condition suivante:

$$\delta(T') < \delta(A_c^{instable}) \quad (26)$$

Une fois qu'on a trouvé ces deux polynômes de départ (ils existent presque toujours, la condition d'existence est donnée dans le document [4]), on peut ajuster T' en se servant des équations (25) et en choisissant un polynôme Q adéquat. La dernière étape décrit comment on peut trouver le meilleur Q .

Définition du critère d'optimisation:

Comme il a déjà été décrit, il est souvent pas suffisant de se baser uniquement sur l'erreur pour définir le critère d'optimisation. Il est par contre nécessaire d'introduire un terme qui tient compte du signal de commande: Le critère est donc:

$$J = \alpha_1^2 \|e(k)\|^2 + \alpha_2^2 \|u(k)\|^2 \quad (27)$$

où les coefficients α_1 et α_2 servent à pondérer les deux termes. Il n'y a pas de méthode analytique pour bien choisir ces deux coefficients: Il faut choisir des valeurs et puis vérifier que le signal de commande ne dépasse pas la limite.

En introduisant les équations (23) , (24) et (25) dans (27) et en simplifiant légèrement, le critère peut être récrit en fonction de Q .

$$J(Q) = \alpha_1^2 \left\| \frac{(P_0 + q^{\gamma_0 + \gamma_1} \cdot B^{instable} \cdot Q) \cdot B_c}{q^{\gamma_0 + \gamma_1} \cdot A_m \cdot A_c^{stable}} \right\|^2 + \alpha_2^2 \left\| \frac{A \cdot (T_0' - A_c^{instable} \cdot Q)}{B^{stable} \cdot A_m} \right\|^2 \quad (28)$$

Optimisation du polynôme T_0' :

Cette étape contient deux théorèmes importants dont les preuves et les descriptions complètes ne sont pas données ici. Le lecteur intéressé est encore une fois renvoyé aux documents [3] et [4].

Pour simplifier les équations à la suite, écrivons l'équation (28) à l'aide des variables supplémentaires suivantes:

$$V^1 = \frac{P_0 \cdot B_c}{q^{\gamma_0 + \gamma_1} \cdot A_m \cdot A_c^{stable}} \quad (29)$$

$$W^1 = \frac{-B^{instable} \cdot B_c}{A_m \cdot A_c^{stable}} \quad (30)$$

$$V^2 = \frac{A \cdot T_0'}{B^{stable} \cdot A_m} \quad (31)$$

$$W^2 = \frac{A \cdot A_c^{instable}}{B^{stable} \cdot A_m} \quad (32)$$

notons que l'équation (27) peut s'écrire maintenant de la façon suivante:

$$J(Q) = \alpha_1 \cdot \|V^1 - W^1 \cdot Q\|^2 + \alpha_2 \cdot \|V^2 - W^2 \cdot Q\|^2 \quad (33)$$

Définissons les vecteurs ψ_1 et ψ_2 de la façon suivante:

$$\psi_j(V^j, W^j) = \begin{bmatrix} \langle V, W \rangle \\ \langle V, q \cdot W \rangle \\ \vdots \\ \langle V, q^n \cdot W \rangle \end{bmatrix} \quad (34)$$

où $\langle X, Y \rangle = \text{produit scalaire} = \sum_i (x_i \cdot y_i)$ avec x_i et y_i les réponses impulsionnels de X et Y . La variable n , dont le choix de la valeur est expliqué à la fin de ce chapitre, est le

degré de $Q = q_0 + q_1 \cdot q + \dots + q_n \cdot q^n$.

On définit également des matrices ϕ_1 et ϕ_2 de taille (n x n):

$$\phi_j(V^j, W^j) = \begin{bmatrix} \langle W, W \rangle & \langle q \cdot W, W \rangle & \dots & \langle q^n \cdot W, W \rangle \\ \langle W, q \cdot W \rangle & \langle q \cdot W, q \cdot W \rangle & \dots & \langle q^n \cdot W, q \cdot W \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle W, q^n \cdot W \rangle & \langle q \cdot W, q^n \cdot W \rangle & \dots & \langle q^n \cdot W, q^n \cdot W \rangle \end{bmatrix} \quad (35)$$

Un vecteur \bar{q} va être associé à Q : $\bar{q} = \begin{pmatrix} q_0 \\ q_1 \\ \dots \\ q_n \end{pmatrix}$

La solution pour le problème de minimisation lié au critère (33) se calcule selon l'équation suivante (sans démonstration):

$$(\alpha_1 \cdot \psi_1 + \alpha_2 \cdot \psi_2) = (\alpha_1 \cdot \phi_1 + \alpha_2 \cdot \phi_2) \cdot \bar{q} \quad (36)$$

L'équation (36) permet de trouver Q et donc T par le biais de l'équation (25).

Remarque:

Si on a un preview (égale à $\gamma_0 + \gamma_1$) qui est plus grand que le retard lié au système ($= \gamma_0$, donc $\gamma_1 > 0$) ou un degré de T qui est plus grand que celui de R, alors le régulateur devient non causal et il n'est pas possible de faire l'optimisation (parce-que matlab n'aime pas les système non causals). Il faut donc rendre le système causal en simplement augmentant le degré de R, S et aussi A_m par κ (multiplier les deux polynômes par q^κ), où κ se calcule selon la formule suivante: $\delta(Q) \leq \delta(R) - \delta(A_0) - \delta(A_c^{instable}) + \kappa$ Il est préférable de prendre le κ le plus petit.

2.3.2 MÉTHODE LÉGÈREMENT MODIFIÉE

L'un des deux désavantages de la méthode est la nécessité de connaître la transformée en z de la consigne (l'autre, c'est le modèle du système qui doit être identifié et qui n'est donc jamais parfait). Si les transformées pour les consignes simples sont connues, il est difficile de les trouver pour des signaux quelconques. Il est possible de les approximer, mais ceci introduit des erreurs. Pour avoir un modèle exacte de la transformé, il faut augmenter l'ordre de ce modèle. Il faut quand-même vérifier que le nombre des pôles instables n'est pas beaucoup trop grand, parce-que ceci augmenterait l'ordre du polynôme T par le biais de l'équation (25). Toutefois, ceci n'est pratiquement jamais le cas, parce-qu'on utilise généralement des consignes stables. Cette dernière remarque est pleinement exploitée dans la suite.

Notons d'abord que pour la construction des vecteurs Ψ_j et pour les matrices Φ_j on utilise les réponses impulsionnelles de V et W pour calculer les produits scalaires (équations (34) et (35)). Mais la réponse impulsionnelle de la transformée en z de la consigne donne cette même consigne dans le domaine temporel. On peut donc écrire la réponse impulsionnelle de V et W :

$$V_{impuls}^1 = \frac{P_0 \cdot B_c}{q^{y_0+y_1} \cdot A_m \cdot A_c^{stable}} \cdot \delta(k) \cdot \frac{A_c^{instable}}{A_c^{instable}} = \frac{P_0 \cdot A_c^{instable}}{q^{y_0+y_1} \cdot A_m} \cdot y_r(k) \quad (37)$$

et de même:

$$W_{impuls}^1 = \frac{-B^{instable} \cdot B_c}{A_m \cdot A_c^{stable}} \cdot \delta(k) \cdot \frac{A_c^{instable}}{A_c^{instable}} = \frac{-B^{instable} \cdot A_c^{instable}}{A_m} \cdot y_r(k) \quad (38)$$

Les définitions de V^2 et W^2 ne font pas intervenir la consigne, il n'est donc pas nécessaire de les changer. Maintenant, le vecteur Ψ_1 et la matrice Φ_1 qui sont utilisés pour calculer \bar{q} sont simplement:

$$\Psi_1(V_{impuls}^1, W_{impuls}^1) = \begin{bmatrix} \sum_i V_{impuls}^1(i) \cdot W_{impuls}^1(i) \\ \sum_i V_{impuls}^1(i) \cdot [q \cdot W_{impuls}^1](i) \\ \vdots \\ \sum_i V_{impuls}^1(i) [q^n \cdot W_{impuls}^1](i) \end{bmatrix} \quad (39)$$

et

$$\Phi_1 = \begin{bmatrix} \sum_i [W_{impuls}^1] \cdot [W_{impuls}^1] & \sum_i [qW_{impuls}^1] \cdot [W_{impuls}^1] & \dots & \sum_i [q^n W_{impuls}^1] \cdot [W_{impuls}^1] \\ \sum_i [W_{impuls}^1] \cdot [qW_{impuls}^1] & \sum_i [qW_{impuls}^1] \cdot [qW_{impuls}^1] & \dots & \sum_i [q^n W_{impuls}^1] \cdot [qW_{impuls}^1] \\ \vdots & \vdots & \ddots & \vdots \\ \sum_i [W_{impuls}^1] \cdot [q^n W_{impuls}^1] & \sum_i [qW_{impuls}^1] \cdot [q^n W_{impuls}^1] & \dots & \sum_i [q^n W_{impuls}^1] \cdot [q^n W_{impuls}^1] \end{bmatrix} \quad (40)$$

Cependant, pas toutes les traces de la transformée en z de la consigne ne sont disparues: Le terme $A_c^{instable}$ est resté. Il est désormais possible d'exprimer cet terme: Écrivons la consigne dans le domaine temporel discret:

$$C_{temporel} = c_0 \cdot \delta(0) + c_1 \cdot \delta(1) + \dots + c_k \cdot (\delta(k)) \quad , k \in \mathbb{N}$$

Ceci nous permet d'écrire directement la transformée en z:

$$C_{ztrans} = c_k \cdot q^k + \dots + c_1 \cdot q + c_0 \quad , k \in \mathbb{N}$$

$A_c^{instable}$ vaut donc 1.

Première remarque: Dans la pratique, la valeur maximale de k est toujours limité. La transformée en z ainsi obtenu n'est donc qu'une approximation. La valeur de la réponse impulsionnelle de C_{ztrans} est égale à zéro pour tout instant $> k$. Mais la vraie consigne

garde sa valeur finale jusqu'à l'éternité (1 pour les s-courbes utilisées). Ceci ne limite pourtant pas la méthode utilisée si k est assez grand pour avoir toute la partie intéressant de la consigne (il n'y a pas besoin d'optimiser la partie où la consigne reste constant). En plus, la méthode originale se base également sur des réponses impulsionnelles de durée finie, il n'y a donc pas de différences de ce point de vue.

Deuxième remarque: Avant de calculer Q il faut d'abord calculer T'_o et P_o à l'aide de l'équation (19). Mais la condition pour le degré de T'_o (26) ne peut pas être satisfait parce que le degré de $A_c^{instable}$ vaut zéro et le degré de T'_o devrait donc être négatif. Il faut se servir d'une astuce: Si on écrit $C_{ztrans} = C_{ztrans} \cdot \frac{q-1}{q-1}$, alors $A_c^{instable} = (q-1)$ et son degré vaut 1 et il

est maintenant possible de trouver une solution à l'équation (19) qui vérifie (26). On a introduit un facteur (arbitrairement choisi égale à q-1) dans A_c , il est donc nécessaire d'introduire le même facteur dans le nominateur B_c qui intervient dans les équations (37) et (38). On réécrit donc:

$$V_{impuls}^{1\ prime} = \frac{P_o \cdot B_c}{q^{\gamma_0 + \gamma_1} \cdot A_m \cdot A_c^{stable}} \cdot \delta(k) \cdot \frac{A_c^{instable}}{A_c^{instable}} = V_{impuls}^1 \cdot (q-1) \quad (41)$$

et de même:

$$W_{impuls}^{1\ prime} = \frac{-B_c^{instable} \cdot B_c}{A_m \cdot A_c^{stable}} \cdot \delta(k) \cdot \frac{A_c^{instable}}{A_c^{instable}} = W_{impuls}^1 \cdot (q-1) \quad (42)$$

et on utilisera $V_{impuls}^{1\ prime}$ et $W_{impuls}^{1\ prime}$ au lieu de V_{impuls}^1 et W_{impuls}^1 dans la suite (pour la construction de ϕ_1 et ψ_1).

2.4 CALCUL DE « R » ET « S » À PARTIR D'UN RÉGULATEUR PID

Le but de ce projet est de comparer les résultats d'optimisations d'un régulateur RST à un régulateur PID déjà existant. Il est judicieux de choisir les polynômes R et S équivalent au régulateur auquel on veut comparer les résultats. Ceci est facilement possible.

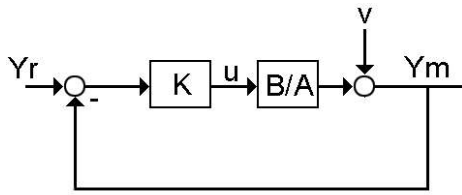


figure 6: régulateur PID

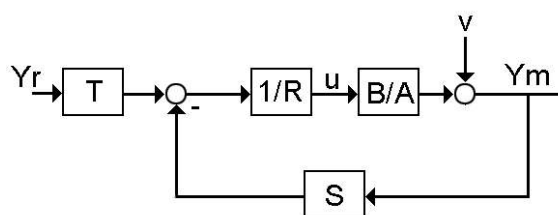


figure 7: régulateur RST

On trouve l'équation suivante pour le régulateur PID:

$$\begin{aligned} u(k) &= K(q^{-1}) \cdot (y_r(k) - y(k)) = \frac{N(q^{-1})}{D(q^{-1})} \cdot (y_r(k) - y(k)) \\ \Rightarrow u(k) &= \frac{1}{D(q^{-1})} \cdot [N(q^{-1}) \cdot y_r(k) - N(q^{-1}) \cdot y(k)] \end{aligned} \quad (43)$$

L'équation pour le RST est:

$$u(k) = \frac{1}{R(q^{-1})} \cdot [T(q^{-1}) \cdot y_r(k) - S(q^{-1}) \cdot y(k)] \quad (44)$$

En comparant les équations (27) et (28) on trouve:

$$\begin{aligned} R(q^{-1}) &= D(q^{-1}) \\ S(q^{-1}) &= N(q^{-1}) \\ T(q^{-1}) &= N(q^{-1}) \end{aligned} \quad (45)$$

Résultats:

Tschirren [7] a donné l'équation du régulateur PID dans son rapport:

$$\begin{aligned} K(q^{-1}) &= KI + K2 \cdot (1 - q^{-1}) \cdot 2^8 + \frac{K4}{1 - q^{-1}} \\ \Rightarrow K(q^{-1}) &= \frac{(KI + K2 \cdot 2^8 + K4) - (KI + 2 \cdot K2 \cdot 2^8) \cdot q^{-1} + K2 \cdot 2^8 \cdot q^{-1}}{1 - q^{-1}} \end{aligned} \quad (46)$$

K1, K2 et K4 sont les registres du système ETEL qui caractérisent le régulateur PID. Ce régulateur est donné pour une fréquence de 24kHz. Nous voulons un régulateur de 6kHz, il faut donc convertir ce régulateur avant de trouver l'équivalent en RST:

$$\begin{aligned} K(q^{-1}) &= KI + \frac{K2}{4} \cdot (1 - q^{-1}) \cdot 2^8 + \frac{K4 \cdot 4}{1 - q^{-1}} \\ \Rightarrow K(q^{-1}) &= \frac{(KI + \frac{K2}{4} \cdot 2^8 + 4 \cdot K4) - (KI + \frac{K2}{2} \cdot 2^8) \cdot q^{-1} + \frac{K2}{4} \cdot 2^8 \cdot q^{-1}}{1 - q^{-1}} \end{aligned} \quad (47)$$

Pour $K1=6500$, $K2=2700$, $K4=12$ et le premier modèle de Tschirren on trouve:

$$\begin{aligned} R(q^{-1}) &= [65536 - 65536 \cdot q^{-1}] \cdot 2^{-16} \\ S(q^{-1}) &= [179348 - 352100 \cdot q^{-1} + 172800 \cdot q^{-2}] \cdot 2^{-16} \\ T(q^{-1}) &= S(q^{-1}) \end{aligned}$$

Le facteur 2^{-16} est ajouté dans le système d'EDEL. Les valeurs à introduire dans le système sont donc les valeurs dans les parenthèses (sans ce facteur), les valeurs pour les simulations sont les valeurs avec ce facteur.

Remarque: Pour la deuxième méthode on utilise un modèle de référence qui mène au polynômes R et S ci-dessus. Ceci se trouve facilement à l'aide de l'équation de Diophante (19), avec cette fois ci les inconnues A_m et A_0 et les paramètres R et S . Le résultat est:

$$\begin{aligned} A_m \cdot A_0' &= 1.0000 q^6 - 3.7901 q^5 + 6.4065 q^4 - 6.3427 q^3 + 3.6741 q^2 - 0.9901 q^1 + 0.0423 = A_m \\ A_0 &= q \end{aligned}$$

3 RÉSULTATS

3.1 CONSIGNES

Les deux méthodes sont appliquées à deux consignes que peu différentes: Tous les deux sont de types « s-courbes ». Le nom « s-courbe » signifie que la consigne n'a pas de changements brusques ni de vitesse ni d'accélération pour éviter des chocs mécaniques. Les équations de telles sortes de consignes sont décrites dans le manuel [10].

La vitesse maximale, l'accélération maximale ainsi que le « jerk-time » (temps maximal pour augmenter l'accélération de zéro à sa valeur maximale) sont des paramètres qui peuvent être définis par l'utilisateur et introduits au système par le biais de l'interface « ETEL-tools ». Les méthodes ont été testées pour deux spécifications différentes (qui ont déjà été défini pour le travail de Tschirren [7]):

| | s-courbe I | s-courbe II |
|-----------------------------|--------------------|---------------------|
| position finale (amplitude) | 5 mm | 15 mm |
| vitesse maximale | 0.2 m/s | 0.8m/s |
| accélération maximale | 5 m/s ² | 60 m/s ² |
| « jerk-time » | 0 | 50 incréments |

tableau 1: paramètres des consignes

La deuxième consigne (s-courbe II) est donc beaucoup plus exigeante que la première. La figure ci-dessous illustre ces deux courbes:

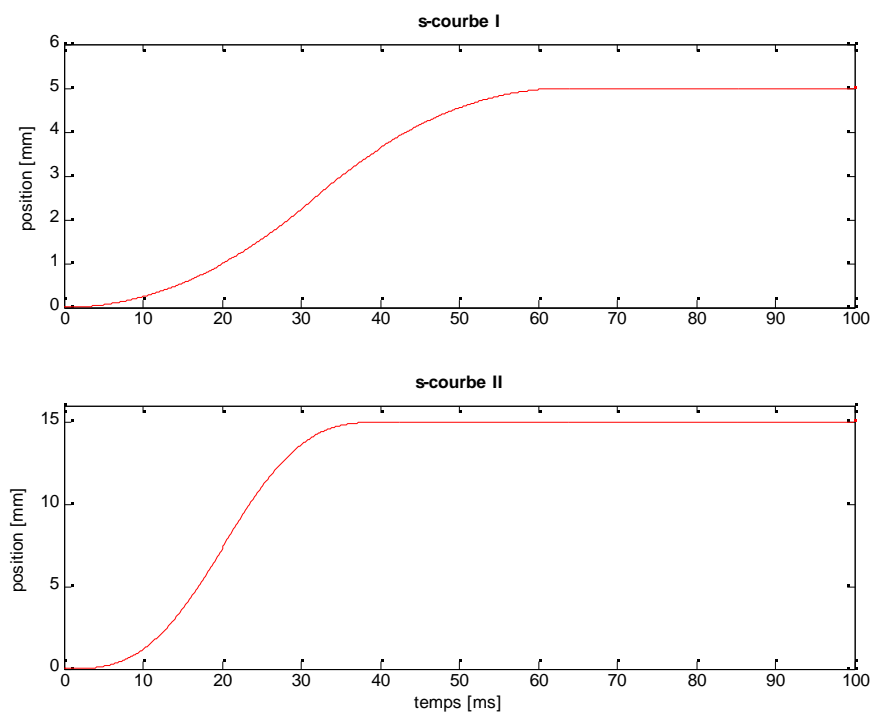


figure 8: les deux consignes

3.2 MODÈLE UTILISÉ – PROBLÈME DE CONVERSION

La deuxième méthode utilise un modèle. Vu que des identifications pour ce système ont déjà été faites plusieurs fois ([7] et [9]), il n'est pas forcément nécessaire de les refaire. Le travail de Tschirren [7] est plus récent que celui de Bays [9], il est donc préférable de travailler avec un modèle de lui (en effet il a trouvé trois modèles légèrement différentes pour trois moteurs différentes.)

Cependant, un problème a été découvert lors de ce projet:

Tschirren a indiqué la relation suivante entre un incrément (unité du capteur de position) et l'unité standard (mètre): $1\text{m} = 409'600'000$ incréments. Mais sur le système actuel on a $1\text{m} = 204'800'000$.

Si uniquement la définition d'un incrément a changé, ça ne va pourtant pas changer les pôles du modèle qui est donné en mètre/ampère (la conversion entre les ampères et les incréments-ampères est resté) et ce changement peut facilement être introduit dans le modèle par un facteur de correction. On peut pourtant se demander si le système entier n'a pas changé dans le temps?

Pour mettre la dernière question au clair et pour contourner les incertitude concernant la conversion, la réponse à un saut d'unité a été simulée pour deux régulateurs PID différentes et comparé avec un essai sur le système réel. Les deux régulateurs PID_1 et PID_2 sont défini dans le chapitre suivant (« 3.3 Analyse des régulateurs PID »). Pour les simulations, le premier des trois modèles de Tschirren a été utilisé.

Dans tous les quatre figures ci-dessous, la simulation est dessiné en rouge et l'expérience en bleu.

Les deux premières figures montrent les résultats avec la conversion actuelle.

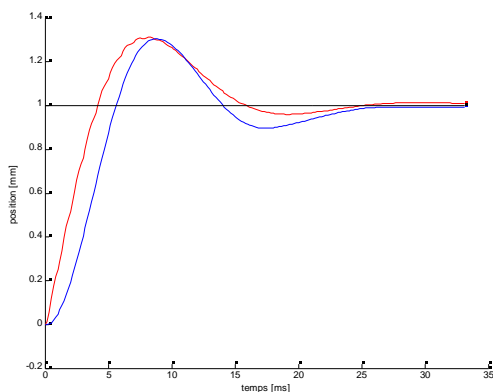


figure 9: PID1

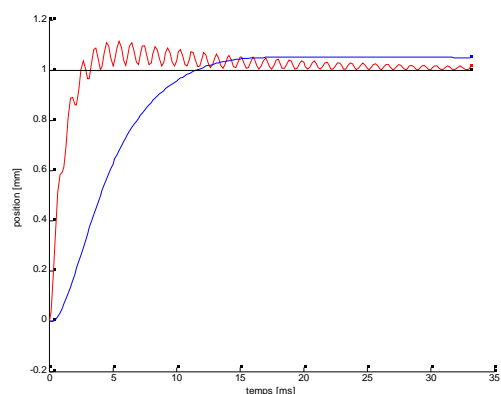


figure 10: PID2

Et maintenant les résultats pour la conversion telle que Tschirren l'a indiqué:

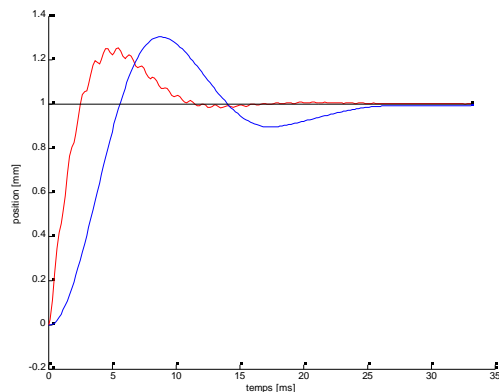


figure 11: PID1

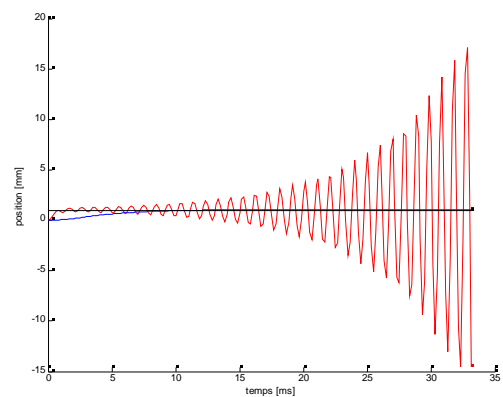


figure 12: PID2

Ces résultats montrent clairement que la conversion $1m = 204'800'000$ est juste et donne de bons résultats. Il y a toujours une certaine différence entre la simulation et la réalité qui découle de l'imperfection du modèle (ordre fini, modèle linéaire) et qui peut me fausser les résultats de la deuxième méthode.

3.3 ANALYSE DES RÉGULATEURS PID

Afin de pouvoir juger sur l'amélioration qui a été atteinte par l'optimisation, les résultats sont comparés à deux régulateurs PID dont les valeurs caractéristiques sont donné dans le tableau ci-dessus. L'équation d'un telle régulateur est donné dans le chapitre « 2.4 Calcul de R et S à partir d'un régulateur PID » (équation (47)). Notons que les polynômes R et S utilisé dans la suite ont été calculé à partir du deuxième régulateur PID (cf chapitre 2.4)

| | PID ₁ | PID ₂ |
|--------------------------|------------------|------------------|
| K1 (terme proportionnel) | 4500 incréments | 6500 incréments |
| K2 (terme dérivatif) | 800 incréments | 2700 incréments |
| K4 (terme intégratif) | 12 incréments | 12 incréments |
| K21 (feedforward) | 0 incréments | 0 incréments |

tableau 2: coefficients de deux régulateurs PID

Le critère utilisé pour juger sur tous les régulateurs est l'erreur quadratique. La deuxième méthode a incorporé ce critère dans son critère d'optimisation. La première en utilise une autre (corrélation), mais dans le même but de minimiser l'erreur de poursuite. La définition de l'erreur quadratique est donnée dans la formule suivante:

$$\text{erreur quadratique} = \sum_{k=1}^N e(k)^2 \quad (48)$$

Le nombre N dépend du type de la consigne (uniquement la monté est jugé, mais pas la partie constante de la courbe).

Le tableau suivant résume les erreurs quadratiques en fonction du régulateur et de la consigne:

| consigne \ régulateur | PID ₁ | PID ₂ |
|-----------------------|-------------------------|------------------------|
| s-courbe I, N=450 | 0.64985 mm ² | 0.20167mm ² |
| s-courbe II, N=300 | 17.7882 mm ² | 6.9239 mm ² |

tableau3: erreur quadratique pour les différentes PID

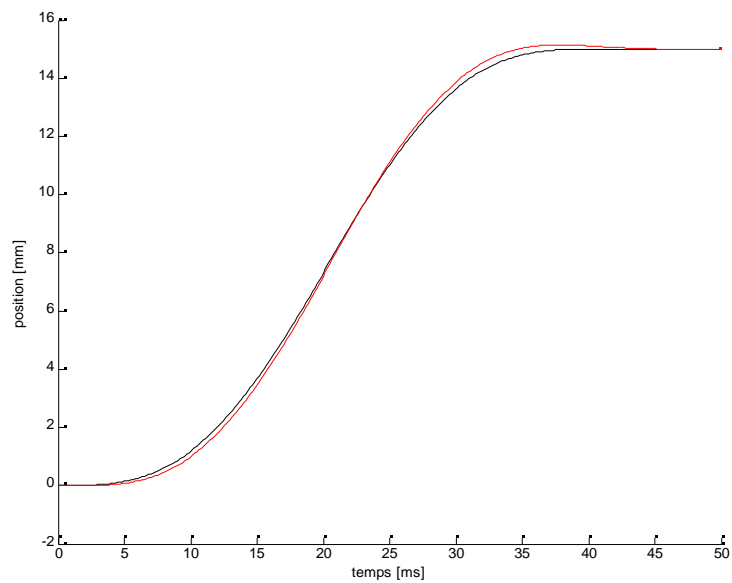


figure 13: suivi de la s-courbe II avec le régulateur PID₂

IL reste à ajouter que tous les quatre régulateurs n'ont pas passé le test de l'indépendance (14)

3.4 RÉSULTATS DE LA PREMIÈRE MÉTHODE (MODEL-FREE)

3.4.1 SIMULATIONS

Comme il a déjà été mentionné plusieurs fois, il n'est pas possible de faire une simulation valable pour la sortie de la boucle fermée à cause du bruit intervenant. La figure ci-dessous montre clairement ceci (degré de T: 8, preview: 2, consigne: s-courbe II)

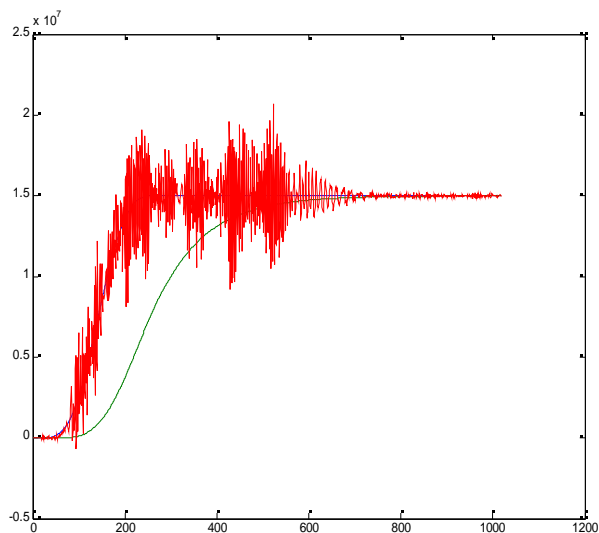


figure 14:

vert: réponse non optimisé - bleu(caché) consigne - rouge: sortie simulé

Les tests aident à choisir le degré de T et le nombre de preview qu'il faut appliquer au système réel. Les deux tables suivantes résument ces résultats (x = le test a passé, rien= le test a failli). Le tableau 4 montre les résultats pour l'optimisation de la consigne s-courbe I et le tableau 5 pour la s-courbe II.

| ordre preview | test d'indépendance(14) | | | | | | | | | | test khi-deux(15) | | | | | | | | | |
|------------------|-------------------------|---|---|---|---|---|---|---|---|----|-------------------|---|---|---|---|---|---|---|---|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0-5 | | | | | | | | | | | x | | x | x | x | x | x | x | x | x |
| 6 | | x | | | | | | | | | x | | x | x | x | x | x | x | x | x |
| 7 | | x | x | | | | | | | | x | | x | x | x | x | x | x | x | x |
| 8 | | x | x | | | | | | x | | x | | x | x | x | x | x | x | x | x |
| 9 | | x | x | | | x | | | | | x | | x | x | x | x | x | x | x | x |
| 15 | | x | | | | | | x | x | x | x | | x | x | x | x | x | | x | x |
| 18 | | x | | x | x | x | x | | x | x | x | | x | x | | | | | | |
| 20 | | x | x | x | x | x | x | x | x | x | x | | x | x | | | | | | |

tableau 4: résultats des tests. Simulation pour la s-courbe I

| ordre preview | test d'indépendance (14) | | | | | | | | | | test khi-deux (15) | | | | | | | | | |
|------------------|--------------------------|---|---|---|---|---|---|---|---|----|--------------------|---|---|---|---|---|---|---|---|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | | | | | | | | | | | x | x | x | x | x | x | x | | | |
| 1 | | | | | | | | | | | x | x | x | x | | | x | | | x |
| 2 | | | | | | | | | x | | x | x | x | | | | x | | | x |
| 3 | | | | | | | x | x | x | | x | x | x | | | | x | | | |
| 4 | | | | | | x | | | x | | x | x | x | | | | x | | | |
| 5 | | | | | | x | | x | x | x | x | x | x | | | | | | | |
| 6 | | | | | | | x | x | x | x | x | x | x | | | | | | | |
| 7 | | x | x | | x | x | x | x | x | x | x | x | x | | | x | x | x | | x |
| 8 | | | x | | | x | x | x | x | x | x | x | | x | x | x | | x | x | x |
| 9 | | | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |

tableau 5: résultats des tests. Simulation pour la s-courbe II

Commentaires:

- Le test khi-deux donne des résultats bizarres. On s'attend à ce que la probabilité monte avec l'ordre croissant du T, ce qui n'est pas du tout le cas. Il est donc pas utilisable telle quelle (Dans le chapitre 2.2.2 il a déjà été mentionné que le test n'a pas été conçu exactement pour cette méthode. Il semble maintenant vérifié que ce test est en effet pas fiable).
- Le test d'indépendance entre la consigne et l'erreur donne des résultats attendus, en tout cas pour la s-courbe II et un peu moins pour la s-courbe I. C'est donc sur celui que je vais me baser pour choisir les degrés de T et le nombre de preview à implémenter.

Le but est de choisir un degré de T et un preview le plus bas que possible, mais de façon à ce que le test passe. En examinant la table, on trouve...

... pour la s-courbe I: [$\delta(T)=2$, preview=6]

... pour la s-courbe II: [$\delta(T)=10$, preview=2]

3.4.2 EXPÉRIENCES

Parce-que les tests ne sont pas tout à fait fiables et pour bien vérifier la méthode, d'autres régulateurs que ceux qui sont mentionnés dans le sous-chapitre précédent ont été appliqué au système réel. Les tableaux suivants montrent les erreurs quadratiques à ces régulateurs pour les deux consignes.

| S-courbe I | | |
|------------|-------------|---|
| preview | $\delta(T)$ | erreur quadratique mesurée [mm ²] |
| 6 | 2 | 0.15792 |
| 20 | 2 | 0.058411 |
| 20 | 3 | 0.11927 |
| 20 | 4 | 0.13008 |

tableau 6: erreurs quadratiques mesurés pour différentes régulateurs, consigne = s-courbe I

| S-courbe II | | |
|-------------|-------------|---|
| preview | $\delta(T)$ | erreur quadratique mesurée [mm ²] |
| 2 | 10 | 0.16123 |
| 6 | 7 | 0.10821 |
| 9 | 3 | 0.17406 |

tableau 7: erreurs quadratiques mesurés pour différentes régulateurs, consigne = s-courbe II

Commentaires:

- Pour tous les deux consignes, une amélioration considérable comparé aux régulateurs PID a été apporté, et surtout pour la deuxième consigne (s-courbe II).
- Les tests d'indépendance ont faillis pour tous les résultats expérimentaux. Ceci montre encore une fois les limites de ces tests. Pourtant, les régulateurs choisis donnent de bons résultats: le test d'indépendance ne peut donc pas être complètement fausses.

Le graphe suivant illustre ces bons résultats pour le deuxième régulateur du tableau 7.

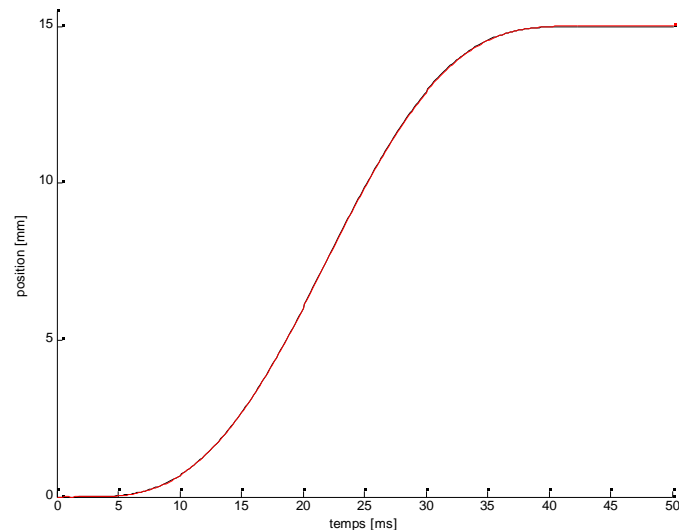


figure 15: résultat expérimental de la deuxième méthode

3.5 DEUXIÈME MÉTHODE

3.5.1 SIMULATIONS

Cette méthode donne très vite de très bons résultats, comme on le voit dans le tableau qui résume les erreurs quadratiques [mm^2] en simulations pour différentes valeurs de l'ordre de T et de preview:

| s-courbe II, $\alpha_1=1, \alpha_2=0$ | | | | | |
|---------------------------------------|--------|------------|-------------|-------------|-------------|
| preview \ $\delta(T)$ | 2 | 3 | 4 | 5 | 6 |
| 0 | 7.0363 | 0.0045764 | 5.5198e-006 | 3.6993e-006 | 0.011855 |
| 1 | 4.9443 | 0.00036033 | 4.1912e-007 | 1.902e-007 | 0.002022 |
| 2 | 3.2741 | 0.00046753 | 3.116e-008 | 8.6117e-009 | 5.4663e-005 |
| 3 | 1.9893 | 0.0029634 | 4.9945e-007 | 9.4274e-008 | 5.5044e-005 |
| 4 | 1.0546 | 0.0063614 | 4.2309e-006 | 1.7677e-006 | 8.1302e-006 |
| 5 | 0.4357 | 0.0095807 | 1.616e-005 | 8.2622e-006 | 0.00029163 |
| 6 | 0.0998 | 0.011904 | 4.0929e-005 | 2.2887e-005 | 0.0023673 |
| 7 | 0.0150 | 0.012939 | 8.0995e-005 | 4.7812e-005 | 0.0077206 |
| 8 | 0.1508 | 0.012572 | 0.0001357 | 8.3369e-005 | 0.017107 |
| 9 | 0.4780 | 0.010937 | 0.00020118 | 0.00012776 | 0.030291 |
| 10 | 0.9687 | 0.008369 | 0.00027084 | 0.00017742 | 0.046044 |

tableau 8: erreurs quadratiques pour différentes T en simulation – rouge: les minima de chaque colonne

Remarque: A partir d'un ordre de T égale à 5, les matrices qui sont utilisé lors des calculs d'optimisation deviennent très proche des matrices singulier (Le polynômes A et B du modèle A/B ont un degré de 4 respectivement 3). Les calculs des 2 dernières colonnes ne sont donc plus exacte (ce qui se voit très bien dans la dernière colonne).

Ce tableau montre bien qu'un preview peut améliorer les performances d'un régulateur. Pourtant, le degré de T est encore plus important que le preview.

Les deux figures qui suivent montrent encore une fois la puissance de la méthode et la notion de preview. Ils correspondent à la simulation d'un régulateur avec un polynôme T de degré 4 et un preview égale à 2 . A gauche le signal de sortie sans tenir compte du preview (qui n'a pas pu être directement implémenté au système, comme il a été mentionné dans la partie théorique), à droite avec le preview:

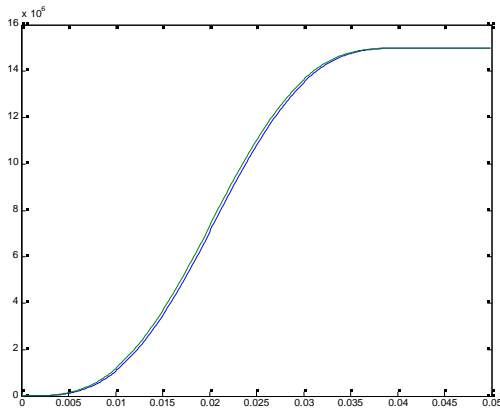


figure 16: Simulation. vert: consigne - bleu: réponse

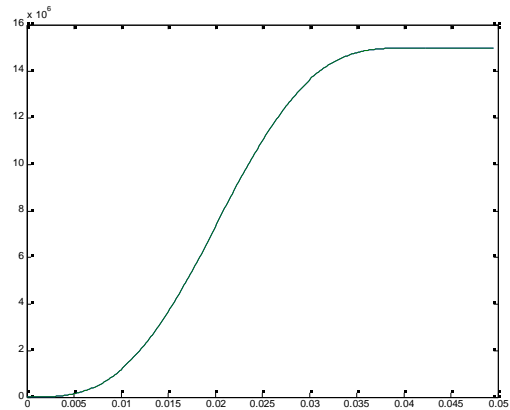


figure 17: Simulation. vert: consigne - bleu: réponse

La courbe à gauche est entièrement couverte par la consigne. On ne voit pas d'erreur, ce qui est excellent! Ceci montre que cette deuxième méthode est beaucoup plus puissante que la première.

Cependant, il faut se méfier des bons résultats de simulations: L'optimisation a été fait pour un modèle particulier et les simulations se basent sur exactement le même modèle. Or, dans la pratique, on arrive jamais à identifier un système de manière tout à fait exacte, et il y a toujours une légère différence entre le système réel et le modèle.

Une autre simulation permet de voir et de quantifier cet effet: Au lieu de faire la simulation avec le même modèle, elle est faite avec un qui est légèrement différent de celui qui a été utilisé pour l'optimisation. Tschirren [7] a donné trois différents modèles correspondant à trois différents moteurs du même type. On va donc optimiser T sur la base du modèle du premier moteur et ensuite simuler le modèle du deuxième avec les paramètres obtenus. Les pôles et zéros des deux différents modèles sont indiqués dans la figure suivante:

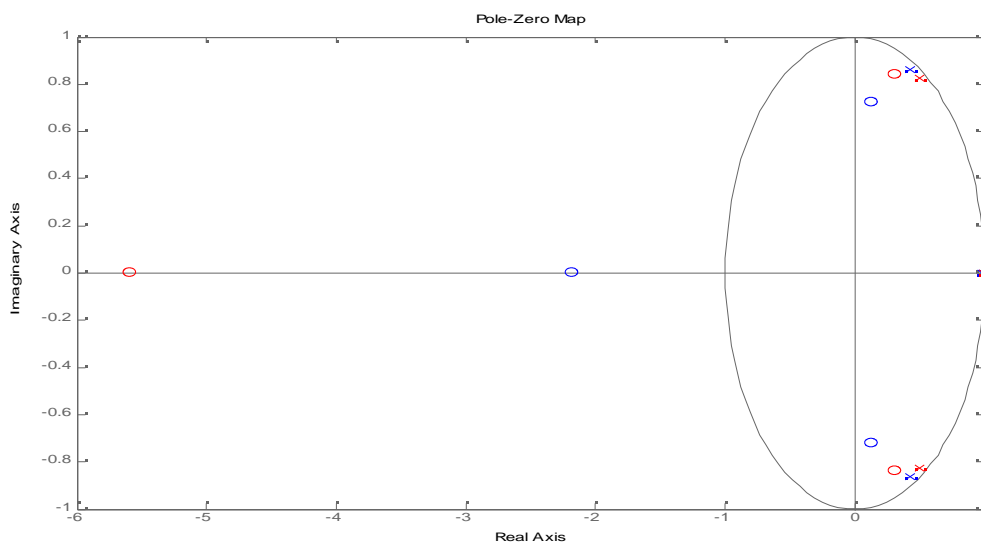


figure 18: bleu: premier modèle - rouge: deuxième modèle

Pour un ordre de T égale à 4 et un preview égale à 2, on trouve une erreur quadratique de 7.2086 mm^2 , qui est beaucoup plus grand que dans la précédente simulation (avec deux fois le même modèle). Il est même pire que le deuxième régulateur PID ! L'erreur est clairement visible dans la réponse:

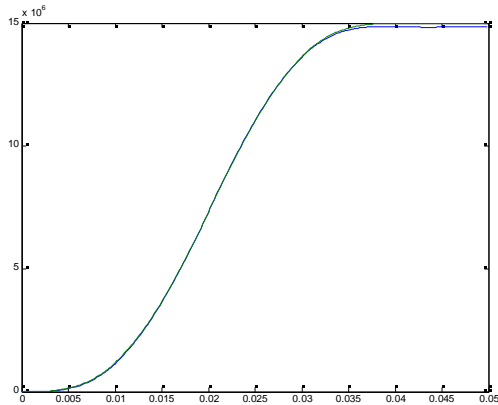


figure 19: vert: consigne - bleu réponse

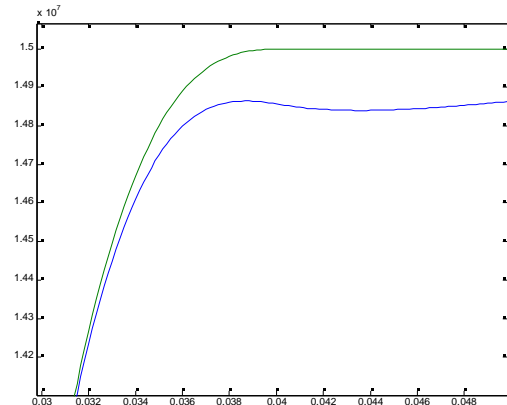


figure 20: détail de la figure à gauche

Contrairement à ce qu'on pourrait croire en regardant les figures, ce n'est pas le gain statique qui est faut, mais c'est vraiment le polynôme T qui est mal optimisé!

Ceci montre clairement la nécessité d'avoir un bon modèle du processus à régler.

3.5.2 EXPÉRIENCES

Les expériences montrent bien la dépendance du modèle. Les deux figures suivantes montrent les résultats pour un régulateur avec un degré de T égale à 2 et un preview égale à 7, une fois optimisé pour le premier modèle de Tschirren et une fois pour le deuxième:

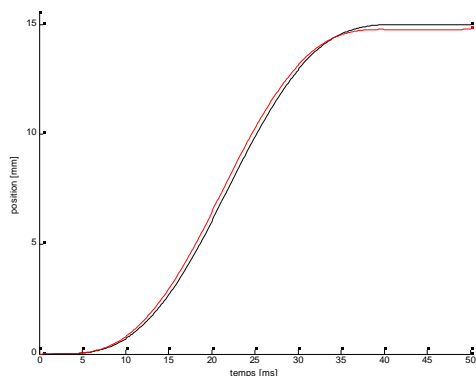


figure 21: Expérience avec le régulateur optimisé pour le 1^{er} modèle. rouge: réponse - noir: consigne

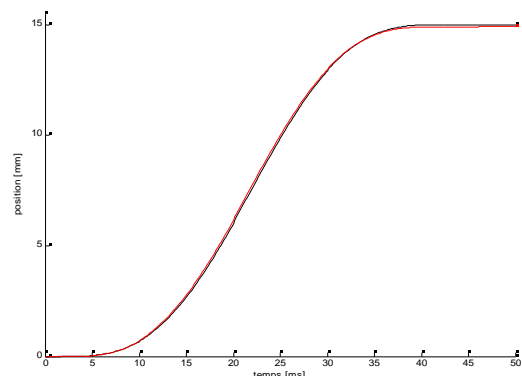


figure 22: Expérience avec le régulateur optimisé pour le 2^{ème} modèle. rouge: réponse - noir: consigne

Le deuxième modèle est donc plus proche de la réalité. Le tableau suivant résume les erreurs quadratiques pour des essais faites avec ce deuxième modèle:

| S-courbe II | | |
|--------------------|-------------------------------|--|
| preview | $\delta(T)$ | erreur quadratique mesurée [mm²] |
| 7 | 2 | 01/02/18 |
| 1 | 3 | 2.3698 |
| 2 | 4 | 2.3335 |

tableau 9: erreurs quadratiques mesurés pour différents régulateurs, consigne = s-courbe II

La méthode donne déjà des résultats intéressants. Cependant, il est absolument indispensable de faire une nouvelle identification pour avoir un modèle exact du processus.

4 CONCLUSION

Les simulations et expériences ont bien montré que les deux méthodes peuvent être utilisées pour l'optimisation du polynôme d'un régulateur RST. Les performances des régulateurs PID ont été dépassées dans tous les cas. En plus, il a été démontré que le preview ajouté peut améliorer considérablement les résultats, mais que le degré du polynôme est plus important que ce dernier.

La première méthode est plus facile à appliquer et donne de bons résultats. Par contre, les tests pour juger le degré et le nombre de preview à implémenter ne sont pas très fiables et devraient être améliorés.

La deuxième méthode donne des résultats de simulations très motivants: Il est possible de trouver de très bons régulateurs avec des ordres et nombre de preview peu élevés (et notamment moins élevés que pour la première méthode). Malheureusement, elle n'est que partiellement prouvée par les résultats expérimentaux: Une nouvelle identification serait indispensable pour justifier entièrement la méthode.

La petite modification de la deuxième méthode a bien apporté des avantages: Il n'est désormais plus nécessaire de trouver la transformée en z de la consigne, ce qui facilite considérablement l'application de la méthode, surtout pour des consignes relativement complexes. En plus, les erreurs sont plus petites que si on travaillait avec une approximation de la transformée en z.

Perspectives:

- Comme il a été prévu au début de ce projet, tous les deux méthodes pourraient être réécrit pour le cas des régulateurs a priori. Ceci a déjà été fait pour la première méthode [2], il serait possible de le faire pour la deuxième.
- La première méthode n'inclut pas un terme pour limiter le signal de commande, comme il est le cas pour la deuxième. Le document [2] propose déjà une telle modification de la méthode. Pour les consignes examinées lors ce travail, le signal de commande n'a jamais été saturé. Dès qu'on travaille avec des consignes plus exigeantes (comme un saut d'unité par exemple), cette modification serait indispensable.

Lausanne, le 24 juin 2005

Christian Künzler

5 ANNEXES

5.1 BIBLIOGRAPHIE

Ouvrages:

- [1] R.Longchamp, « Commande Numérique de Systèmes Dynamiques », Presses Polytechniques et Universitaires Romandes, Lausanne, 1995
- [2] A.Karimi, M.Butcher, R.Longchamp, « Model-Free Precompensator and Feedforward Tuning Based on the Correlation Approach, CTI_projet no. 7049.1 IWS-IW de l'Office fédéral de la Formation Professionnelle et de la Technologie.
- [3] D.Hoover, « Optimal Tracking with Preview », Thèse No. 2756, EPFL, Lausanne, 2003)
- [4] D.N.Hoover, R.Longchamp, J. Rosenthal, «Two-degree- of-freedom l2-optimal tracking with preview », Automatica 40 (2004), pages 155-162
- [5] A.Karimi, L.Miskovic, D.Bonvin, « Iterative correlation-based controller tuning », International Journal of adaptive control and signal processing, June 2004
- [6] D.Bonvin, « Identification de systèmes dynamiques », polycopié EPFL, Lausanne 2003

Anciens projets de semestre et de diplôme:

- [7] J. Tschirren, « Synthèse d'un régulateur RST robuste en performance et en stabilité par loop shaping », Travail de Diplôme, EPFL, printemps 2003
- [8] E. Deschenaux, « Commande d'un moteur synchrone à aimants permanents », Projet de semestre, EPFL, printemps 2004
- [9] X. Bays, « Ajustement d'un régulateur RST pour la commande d'un moteur linéaire synchrone sans fer », Travail de Diplôme, EPFL, printemps 2003

Manuels:

- [10] ETEL, « DSC2-P Hardware Manual », version B, ETEL, 2002

Sites Internet:

- [11] www.etel.ch

5.2 CD-ROM DU TRAVAIL

Un disque contenant les fichiers relatifs à ce travail est ajouté à ce rapport. Une brève description de chacun de ces fichiers est donnée ici:

Fichiers liés au rapport:

- Rapport_kuenzler.sxw: Ce rapport sous forme de fichier « OpenOffice ».
- Rapport_kuenzler.pdf: Ce rapport sous forme de fichier pdf.
- Rapport_kuenzler.doc: Ce rapport sous forme de fichier « Microsoft Word ». Il faut noter que le rapport a été écrit avec « OpenOffice » et que le fichier peut contenir des erreurs de mise en page.

Code matlab:

Ce dossier contient tous les fichiers matlab qui ont été utilisé pour l'optimisation et la simulation ainsi que des données des modèles et consignes:

- RST_tuning_1.m: Le code pour l'optimisation selon la première méthode
- RST_tuning_2.m: Le code pour l'optimisation selon la deuxième méthode
- solve_dioph.m: Une fonction pour résoudre les équations de Diophante
- mfpt.m: « model-free-precompensater-tuning », une fonction utilisé pour la première méthode
- Result_analyse.m: Analyse des résultats expérimentaux
- PID_simulate.m: Simulation des régulateurs PID
- create_Am_from_RS.m: Création d'un modèle à partir d'un polynôme R et S
- get_ETELtool_data.m: Pour lire les données expérimentales dans un fichier qui a été crée avec ETEL-tools
- S_curves.mat: Les données des consignes
- Am_PID.mat: Les modèles qui mènent aux polynômes R et S équivalent au PID
- tschirren.mat: Les trois modèles identifiés par Tschirren [7]

Fichier avec des données expérimentales:

Ces fichiers ont été crée par ETEL-tools et peuvent être importé dans matlab grâce à la fonction « get_ETELtool_data.m »