

PROJET AUTOMATIQUE

Asservissement en position d'une antenne de réception avec MATLAB

Colin Mourard <colin.mourard@ensica.isae.fr>
Anthony Pagliai <anthony.pagliai@ensica.isae.fr>

ISAE - Formation ENSICA - Année 2014-2015

Table des matières

1	Analyse du système en boucle ouverte	2
1.1	Schéma fonctionnel du système	2
1.2	Représentation d'état et fonction de transfert	3
1.3	Etude de la commandabilité et de l'observabilité du système . . .	3
1.4	Etude de la stabilité du système	3
1.5	Simulation temporelle	4
1.6	Représentation graphique du système	4
2	Réglage par synthèse fréquentielle	5
2.1	Efficacité d'un régulateur proportionnel	5
2.2	Réglage du régulateur par avance de phase	5
2.3	Simulation temporelle	6
2.4	Réglage du proportionnel intégral	7
3	Réglage par retour d'état	8
3.1	Choix de la dynamique à adopter	8
3.2	Mise en place du retour d'état	8
3.3	Simulation temporelle sans effet intégral	9
3.4	Mise en place de l'effet intégral	9
4	Synthèse d'un estimateur	9
4.1	Construction de l'estimateur	9
4.2	Problèmes rencontrés avec notre estimateur	10

Introduction

On s'intéresse dans ce projet à une antenne de réception en station sol qui doit être asservie en position pour suivre un satellite de télécommunications. L'antenne est contrôlée via un moteur à courant continu admettant une tension d'entrée maximale de l'ordre de 300V.

Les consignes appliquées en entrée sont assimilées à des échelons d'amplitude 10 degrés.

L'objectif de ce projet est d'asservir au mieux cette antenne de réception en tenant compte des limitations de l'actionneur.

Les équations du moteur à courant continu sont données ci-dessous :

$$u = R_e i + L_e \frac{di}{dt} + k_e \omega_m \quad (1)$$

$$C_m = k_g i \quad (2)$$

$$C_m = J_T \frac{d\omega_m}{dt} \quad (3)$$

Les entrées considérées dans ce système sont :

- La tension d'entrée u (Rappel : $u_{max} \approx 300V$)
- Le couple perturbateur C_z induit par le vent.

Pour asservir l'antenne on dispose aussi de trois capteurs :

- Un capteur de position (k_{cp})
- Un capteur de vitesse (k_{cv})
- Un capteur de courant (k_{ci})

1 Analyse du système en boucle ouverte

1.1 Schéma fonctionnel du système

Pour réaliser le schéma fonctionnel du système en boucle ouverte, on utilise Simulink (dans MatLab) et on obtient le schéma fonctionnel de la FIGURE 1 présenté ci-dessous.

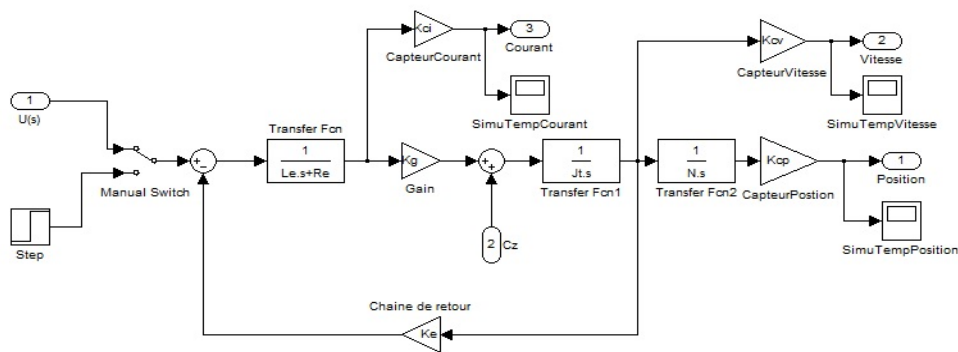


FIGURE 1 – Marges de gain et de phase du système en boucle ouverte

1.2 Représentation d'état et fonction de transfert

Le schéma fonctionnel ainsi obtenu grâce à Simulink nous permet de trouver la représentation d'état de ce système en boucle ouverte en utilisant la commande Matlab `linmod`. On obtient la représentation d'état suivante :

$$\begin{cases} A = \begin{bmatrix} 0 & 500 & 0 \\ 0 & 0 & 100 \\ 0 & -100 & -1000 \end{bmatrix} ; B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \\ C = \begin{bmatrix} 0.0033 & 0 & 0 \\ 0 & 15.7 & 0 \\ 0 & 0 & 500 \end{bmatrix} ; D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{cases}$$

On cherche ensuite à obtenir la fonction de transfert entre la tension en entrée et la position de l'antenne en sortie. Pour ça on utilise deux commandes MatLab, la première est `ss2tf` et nous permet d'obtenir la matrice de transfert entre l'entrée que l'on veut (placée en paramètre) et les différentes sorties. La deuxième est la commande `tf` qui nous permet d'extraire la bonne fonction de transfert en boucle ouverte $\frac{U(p)}{\Theta(p)}$:

$$\frac{U(p)}{\Theta(p)} = \frac{3,411.10^{-13}p^2 - 3,456^{-11}p + 166,7}{p^3 + 1000p^2 + 10000p}$$

1.3 Etude de la commandabilité et de l'observabilité du système

Nous cherchons à savoir si le système est commande et/ou observable. Pour cela, nous utilisons les fonctions MatLab `obsv` pour obtenir la matrice \mathcal{O} d'observabilité et `ctrb` pour obtenir la matrice \mathcal{C} de commandabilité. Ce sont toutes les deux des matrices de rang 3 (c'est un résultat que nous obtenons facilement avec la commande `rank` de MatLab) qui est la taille de la matrice A de la représentation d'état, donc on peut en conclure que le système est à la fois observable et commandable.

1.4 Etude de la stabilité du système

Nous nous intéressons ici aux pôles de la fonction de transfert que nous avons calculée au paragraphe 1.2. de ce document. Pour les calculer, il suffit de trouver les valeurs propres de la matrice A de la représentation d'état, c'est pour cela que nous utilisons la commande `eig` qui donne les valeurs propres d'une matrice.

Nous trouvons que les pôles sont :

$$\mathcal{P} = \{0; -10, 10; -989.90\}$$

Ce sont 3 pôles réels dont 2 sont à partie réelle strictement négative, donc sont stables. Le dernier pôle est zéro, le système fonctionne donc en limite de stabilité. Il va falloir modifier cela dans la suite pour ne pas risquer de se trouver dans une situation où le système pourrait dériver vers une instabilité.

1.5 Simulation temporelle

Afin d'étudier l'influence de la commande d'entrée (en tension) sur le système, et notamment sur la position de l'antenne, nous réalisons une simulation temporelle sous SimuLink avec les caractéristiques suivantes :

- Echelon en entrée d'amplitude 100V (300V possible au maximum)
- Pas de perturbation

Les comportements en courant et en vitesse nous intéressent moins, c'est pourquoi nous n'en parlerons pas. En revanche, nous pouvons voir sur la FIGURE 2 présentée ci-dessous que la réponse temporelle en position est linéaire lorsqu'on applique en entrée un échelon de tension de 100V. Cette observation est un marqueur de l'instabilité du système en boucle ouverte.

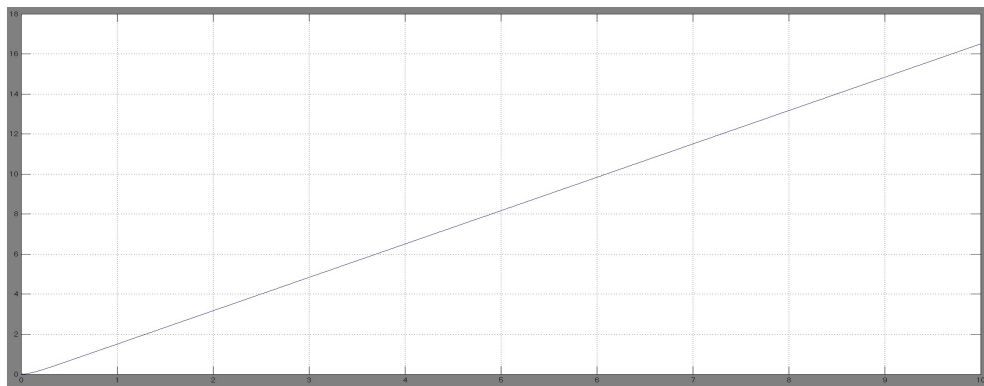


FIGURE 2 – Réponse temporelle en position à un échelon de tension de 100V en entrée

1.6 Représentation graphique du système

Sur la FIGURE 3 sont représentées les marges de gain et de phase (sur un tracé du diagramme de Bode) du système en boucle ouverte. L'utilisation de la commande MatLab `allmargin` nous donne accès aux valeurs de ces marges, à savoir 6E4 pour la marge de gain et 89,9 deg pour la marge de phase.

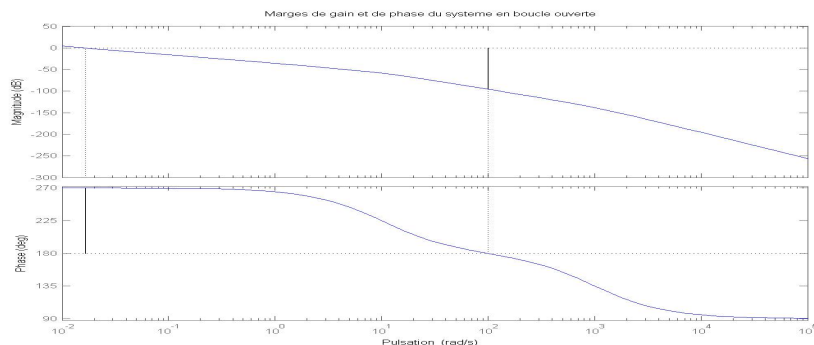


FIGURE 3 – Marges de gain et de phase du système en boucle ouverte

Etant donné notre nécessité d'augmenter la stabilité du système, plusieurs stratégies s'offrent à nous. Dans un premier temps, nous tentons de stabiliser notre système par un réglage fréquentielle.

2 Réglage par synthèse fréquentielle

2.1 Efficacité d'un régulateur proportionnel

Dans un premier temps, il nous paraît légitime de s'interroger quant à l'efficacité d'un régulateur proportionnel (i.e. un simple gain K). En effet, c'est une solution extrêmement simple à mettre en oeuvre.

Afin de connaître l'influence d'un tel régulateur sur notre système, nous utilisons la commande MatLab `rltool`. Cette commande nous permet de visualiser l'influence d'un gain proportionnel K à l'amont de notre système en boucle ouverte. La commande `rltool` s'applique pour des valeurs de K variant de 0 à $+\infty$.

Pour connaître l'influence du gain K sur $[-\infty; 0]$, il suffit d'appeler la commande `rltool` sur l'opposé de la fonction de transfert.

La conclusion est qu'aucune valeur de K ne permet de stabiliser le système. En effet, en se déplaçant sur le graphe ainsi tracé par MatLab, on constate que quelque soit la valeur de K , la fonction de transfert possède toujours au moins un pôle à partie réelle non strictement négative. Il faut donc envisager une solution technologique plus élaborée pour stabiliser notre système.

2.2 Réglage du régulateur par avance de phase

Nous choisissons ce réglage pour la simple et bonne raison qu'il permet d'améliorer les performances en stabilité du système sans détériorer ses performances en régime permanent. Pour satisfaire les spécifications en régime permanent de notre système, à savoir 300V au maximum pour l'entrée de notre système et des échelons en entrée de 10 deg, il nous faut tout de même ajouter un gain cas au régulateur d'avance de phase classique, ce qui conduit à un régulateur dont la fonction de transfert est donnée par :

$$R(p) = K_{av} \cdot \frac{1 + aT_{av}p}{1 + T_{av}p} \quad (4)$$

Première étape : réglage du gain K_{av} : Nous utilisons le théorème de la valeur initiale appliqué à une tension maximum d'entrée de 300V et à des

échelons en entrée de 10 deg, ce qui nous donne :

$$\begin{aligned}
 & \lim_{p \rightarrow +\infty} p.R(p) \cdot \frac{10}{p} \cdot \frac{\pi}{180} = 300 \\
 \Leftrightarrow & \lim_{p \rightarrow +\infty} p.K_{av} \cdot \frac{1 + aT_{av}p}{1 + T_{av}p} \cdot \frac{10}{p} \cdot \frac{\pi}{180} = 300 \\
 \Leftrightarrow & a.K_{av} \cdot 10 \cdot \frac{\pi}{180} = 300 \\
 \Leftrightarrow & a.K_{av} = \frac{300 \cdot 180}{10 \cdot \pi} \approx 1719
 \end{aligned}$$

Deuxième étape : réglage des paramètres de l'avance de phase : Pour ce faire, nous devons dans un premier temps multiplier la FTBO par la valeur $a.K_{av}$ mise en évidence ci-dessus. Nous trouvons une nouvelle fonction de transfert que nous appelons F_{av} . Son expression nous importe peu, donc nous ne la donnons pas, mais soulignons le fait qu'à ce stade, la fonction de transfert est non corrigée.

Dans un second temps, on s'intéresse aux nouvelles valeurs des marges de gain et de phase de la fonction de transfert F_{av} . Pour les trouver, nous utilisons à nouveau la commande MatLab `allmargin`. Cette commande nous donne une marge de phase de 32 deg. Le système peut être considéré comme stable pour une marge de phase de 45 deg, donc a priori la valeur de l'angle de correction ϕ_d que l'on doit appliquer serait $45 - 32 = 13$ deg. Pour avoir une marge dans notre calcul, nous majorons cette valeur de 20% avant de calculer a .

En inversant la relation liant l'angle ϕ_d et le paramètre a nous obtenons

$$a = \frac{1 + \sin \phi_d}{1 - \sin \phi_d} = 1,73$$

Par simple division, nous avons la valeur du gain $K_{av} \approx 993$.

Pour calculer le temps caractéristique T_{av} nous utilisons la relation $T_{av} = \frac{1}{\omega_c \sqrt{a}}$. La pulsation de coupure ω_c nous est donnée par la commande `allmargin` appliquée à la fonction de transfert en boucle ouverte et non corrigée F_{av} . Finalement, nous trouvons $T_{av} \approx 49ms$.

2.3 Simulation temporelle

Afin de vérifier l'efficacité de notre correcteur, nous effectuons une simulation temporelle avec un échelon de tension de 100V et au bout d'une durée de 5s nous imposons un échelon de perturbation de 5V. Ci-dessous sur la FIGURE 4 la réponse temporelle observée en position.

Nous remarquons que, contrairement à la première simulation temporelle, le système se stabilise rapidement. Notre objectif d'améliorer les performances de notre système en stabilité est atteint. Cependant, lors de l'apparition de la perturbation, le système se stabilise mais au bout d'un temps final de 10s, la tension de sortie est de 105V et non de 100V comme demandé par la consigne d'entrée du système. Cela veut dire que notre correcteur ne gère pas les erreurs

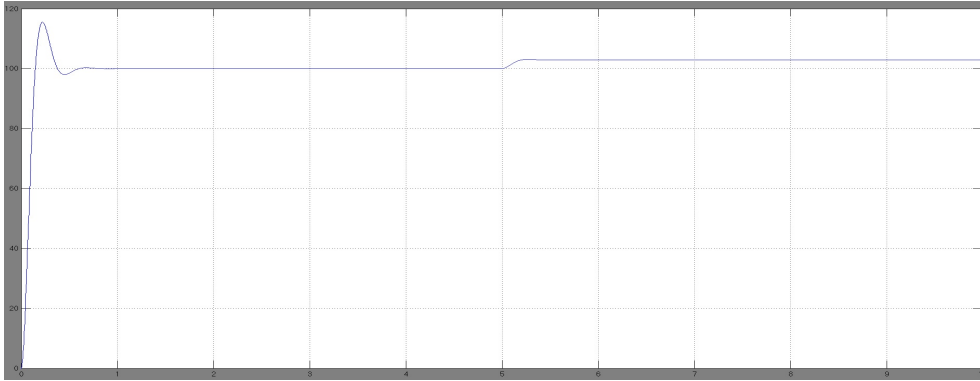


FIGURE 4 – Réponse temporelle à un échelon d’entrée de 100V et à une perturbation de 5V au bout d’une durée de 5s

liées aux perturbations. Il faut donc compléter notre solution technologique par un régulateur proportionnel intégral pour contrer ce phénomène.

2.4 Réglage du proportionnel intégral

Dans un premier temps nous récupérons la pulsation du système corrigé, que nous appelons ω_{cor} . Ensuite, nous savons que la fonction de transfert d’un régulateur proportionnel intégral est de la forme $R(p) = \frac{1+T_{pi}}{T_{pi}}$.

Encore nous faut-il régler la valeur de ce paramètre T_{pi} . Pour cela, nous savons qu’il est proportionnel à $\frac{1}{\omega_{cor}}$. Il nous faut donc déterminer ce coefficient de proportionnalité. Pour cela, nous effectuons plusieurs essais et nous lançons la même simulation temporelle que précédemment. Les résultats sont visibles sur la FIGURE 5 ci-dessous.

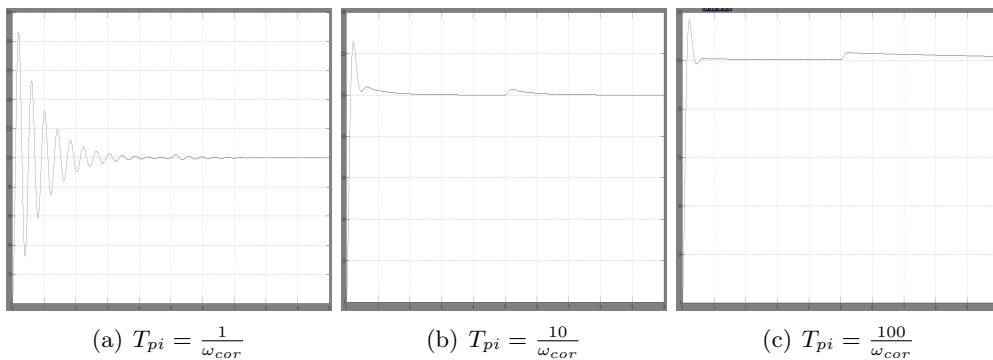


FIGURE 5 – Réponses temporelles à un échelon d’entrée de 100V et à une perturbation de 5V au bout d’une durée de 5s

On remarque que pour le choix de $T_{pi} = \frac{1}{\omega_{cor}}$, le système revient bien à 100V en sortie mais oscille beaucoup trop longtemps et avec une amplitude beaucoup trop importante (de 35V à 185V pour les plus amples oscillations) :

cette solution n'est donc pas acceptable.

Pour le choix de $T_{pi} = \frac{100}{\omega_{cor}}$, le système ne contre pas suffisamment rapidement la perturbation qui apparaît au bout de 5s, la solution n'est pas acceptable non plus.

On choisit donc $T_{pi} = \frac{10}{\omega_{cor}} \approx 0,78s$.

CONCLUSION : Le réglage par synthèse fréquentielle de notre système s'est avéré efficace, mais il est relativement complexe à mettre en oeuvre. L'avantage de cette solution est que nous pouvons contrôler notre système tout entier en n'ayant uniquement une mesure de la position. Nous pouvons donc tenter de régler notre problème de stabilité par une autre méthode, le réglage par retour d'état. Cette solution, moins complexe à mettre en oeuvre, nécessite cependant d'avoir accès à toutes les variables d'état puisque la commande réside en une boucle de retour sur le vecteur d'état.

3 Réglage par retour d'état

3.1 Choix de la dynamique à adopter

Nous avons un système d'ordre 3, il faut donc régler le retour d'état de manière à avoir 3 nouveaux pôles. Pour avoir des performances intéressantes, nous souhaitons avoir un mode oscillant dominant avec un amortissement satisfaisant. Pour se faire, il faut choisir un couple de pôles complexes tel que le facteur d'amortissement soit $\xi = 0,707$. Un tel amortissement est obtenu pour des pôles ayant même partie réelle et partie imaginaire (en valeur absolue). Comme le système doit être stable, il faut des pôles à partie réelle strictement négative. Comme le mode oscillant doit être dominant, le mode réel doit être plus rapide que le mode oscillant. Toutes ces raisons font que nous optons pour les pôles suivants :

$$\mathcal{P}_c = \{-10 + 10i; -10 - 10i; -20\}$$

3.2 Mise en place du retour d'état

Maintenant que nos nouveaux pôles sont définis, il faut calculer le vecteur K de retour d'état. Pour cela, nous utilisons la commande MatLab `place` qui calcule ce vecteur automatiquement.

Nous rebouclons le système avec ce retour d'état. Maintenant, la fonction de transfert n'est plus à gain statique unitaire. Ceci laisse apparaître sur nos simulations temporelles que nous ne sortons pas à 100V comme demandé en entrée de la simulation. Pour remédier à ce problème, nous calculons la fonction de transfert en boucle fermée avec la même procédure que le calcul de la fonction de transfert dans la Section 1 de ce document (commandes MatLab `linmod`, `ss2tf`, `tf` utilisées dans cet ordre). Ensuite, la commande MatLab `dcgain` nous donne la matrice des gains statiques entre toutes les entrées et toutes les sorties. Nous sélectionnons le gain statique correspondant à l'entrée de la Commande et la sortie Position. Nous divisons donc l'entrée de la commande par ce gain statique, ce qui donne un gain statique unitaire pour le système corrigé.

3.3 Simulation temporelle sans effet intégral

Nous effectuons des simulations temporelles sur le système corrigé, présentée sur la FIGURE 6 présentée à la page suivante. Nous remarquons que la dynamique est satisfaisante et conforme à nos attentes. En revanche, la perturbation n'est pas bien gérée par notre asservissement. Pour contrer l'effet néfaste de la perturbation, nous pouvons ajouter un intégrateur à notre système ; c'est l'objet du paragraphe suivant.

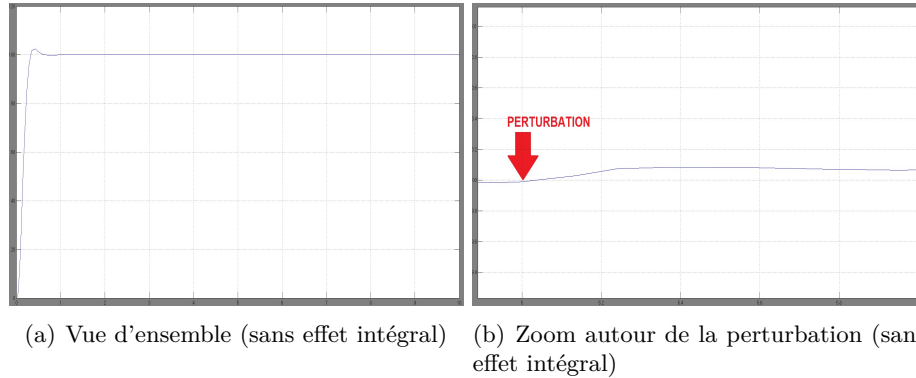


FIGURE 6 – Réponses temporelles à un échelon d'entrée de 100V et à une perturbation de 5V au bout d'une durée de 5s

3.4 Mise en place de l'effet intégral

Nous rajoutons un état qui n'est ni plus ni moins que l'état intégral de la Position. Ce nouvel état porte donc le nombre d'états du système à 4. Nous devons donc régler le retour d'état avec un nouveau pôle. Il doit être plus rapide que le mode dominant. Nous le réglons de manière empirique afin d'avoir la dynamique souhaitée après la perturbation. Ceci nous amène à choisir -235 comme dernier pôle.

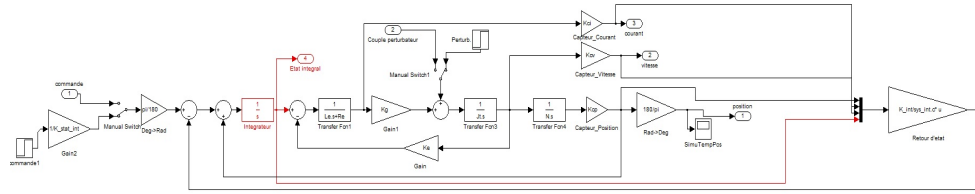
Ensuite, nous réglons à nouveau le gain statique comme précédemment. Nous avons présenté sur la FIGURE 7 le schéma du système corrigé en boucle fermée ainsi que es résultats des simulations temporelles avec prise en compte de l'effet intégral.

4 Synthèse d'un estimateur

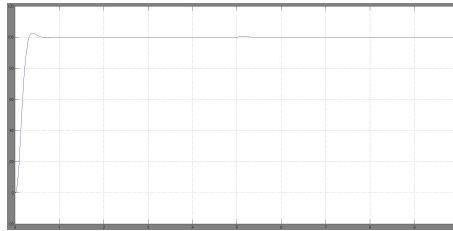
L'intérêt de l'estimateur réside dans le fait qu'à partir de la simple mesure de la position, on peut reconstruire l'ensemble du vecteur d'état, nous permettant d'asservir notre système avec la dynamique souhaitée.

4.1 Construction de l'estimateur

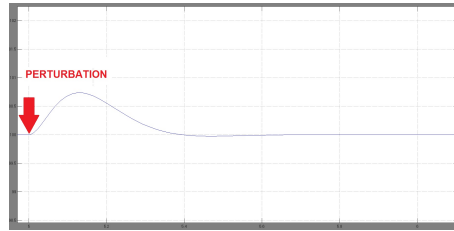
Dans un premier temps, nous avons placé le système réel, représenté en vert sur la FIGURE 8. Celui-ci n'est ni plus ni moins que le système en boucle ouverte



(a) Système corrigé en boucle fermée



(b) Simu. temporelle : Vue d'ensemble



(c) Simu. temporelle : Zoom autour de la perturbation

FIGURE 7 – Retour d'état avec prise en compte de l'effet intégral

de la Section 1 de ce document (cf. FIGURE 1). En revanche, pour le système réel, nous faisons l'hypothèse que nous ne disposons que de l'information de Position donnée par le capteur, c'est pourquoi une seule des sorties vertes est représentée.

Ensuite, nous avons dû concevoir notre estimateur. Pour cela, nous nous sommes appuyés sur le schéma représentation n'importe quel système écrit sous la forme de représentation d'état. Ainsi, il nous est apparu plus simple de savoir où faire reboucler notre matrice de gain d'observation, etc. L'estimateur utilise les matrices A, B, C, D de la représentation d'état du système en boucle ouverte. Il est représenté en bleu sur la FIGURE 8.

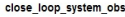
Enfin, pour fonctionner, l'estimateur doit reboucler (à la manière du réglage par retour d'état) avec la matrice de gain d'observation G . Cette boucle de retour est représentée en rouge sur la FIGURE 8.

En orange, nous avons simplement explicité l'opération de conversion de degrés en radians (et inversement) puisque l'échelon de commande correspond à un échelon de 10 deg. Ce n'est pas primordial dans la compréhension de ce système.

4.2 Problèmes rencontrés avec notre estimateur

La synthèse de notre estimateur nous a malheureusement posé quelques problèmes. Déjà, la simulation temporelle ne va pas jusqu'au terme des 10s, il nous est donc difficile de voir si la dynamique que nous lui avons imposée (avec la matrice G) est convenable ou pas.

Ensuite, quand bien même la simulation marcherait, (ce qui est le cas pour les 9 premières secondes simulées) le temps de calcul est très long, de l'ordre de



Enfin, pour le réglage de la matrice de gain d'observation G , nous savons que sa dynamique doit être plus rapide que celle du système réel (en effet, il est difficile d'observer un phénomène si le dispositif mis en place pour l'observer est moins rapide!), mais nous ne savons pas dans quelle mesure. Pour effectuer le réglage de ceux-ci efficacement, il aurait fallu que notre outil de simulation temporelle nous donne des résultats satisfaisants, ce qui n'a pas été le cas.