

# Commande linéaire avancée

---

## Modélisation, analyse et commande d'un système à trois bacs d'eau

---

Evgeny SHULGA  
Paul DANDIGNAC

29 octobre 2017

M2 EEA RODECO Commande

Université Paul Sabatier Toulouse III

---

## Sommaire

<b>1</b>	<b>Modélisation du procédé trois bacs</b>	<b>2</b>
<b>2</b>	<b>Analyse du procédé autour du point d'équilibre</b>	<b>3</b>
2.1	Stabilité, commandabilité et observabilité . . . . .	3
2.2	Étude de la fonction de transfert . . . . .	3
<b>3</b>	<b>Asservissement du procédé</b>	<b>6</b>
3.1	Cahier des charges . . . . .	6
3.2	Première approche par retour d'état - méthode de Bass-Gura . . . . .	6
3.3	Seconde approche par retour d'état - mise sous forme commandable par blocs . . . . .	9
<b>4</b>	<b>Conclusion</b>	<b>12</b>

## 1 Modélisation du procédé trois bacs

Nous considérons le procédé trois bacs avec les débits d'entrées variables  $Q_1(t)$  et  $Q_2(t)$ . Les différentes hauteurs  $H_1(t)$ ,  $H_2(t)$  et  $H_3(t)$  respectent par conséquent l'inégalité  $H_1(t) \geq H_3(t) \geq H_2(t) \geq 0$ .

A partir de l'équation de Torricelli nous pouvons exprimer le changement de la hauteur d'eau dans chaque bac par le système d'équations différentielles suivant :

$$\begin{cases} \frac{dH_1(t)}{dt} = \frac{1}{S} * (Q_1(t) - a_{13} \sqrt{|h_1 - h_3|} * \text{sign}(h_1 - h_3)) \\ \frac{dH_3(t)}{dt} = \frac{1}{S} * (a_{13} \sqrt{|h_1 - h_3|} * \text{sign}(h_1 - h_3) - a_{32} \sqrt{|h_3 - h_2|} * \text{sign}(h_3 - h_2)) \\ \frac{dH_2(t)}{dt} = \frac{1}{S} * (Q_2(t) + a_{32} \sqrt{|h_3 - h_2|} * \text{sign}(h_3 - h_2) - a_{20} \sqrt{|h_2|}) \end{cases} \quad (1.0.1)$$

Les valeurs  $a_{13}$ ,  $a_{32}$  et  $a_{20}$  sont déjà calculé.

Nous pouvons calculer le point d'équilibre pour une commande constante  $Q_1(t) = Q_{10}$ ,  $Q_2(t) = Q_{20}$ . Les hauteurs d'eau de chaque bac à cet équilibre sont égal  $H_{10}$ ,  $H_{20}$  et  $H_{30}$  et se calculent de façon suivante :

$$\begin{cases} H_{20} = \left( \frac{Q_{10} + Q_{20}}{a_{20}} \right)^2 \\ H_{30} = H_{20} + \left( \frac{Q_{10}}{a_{32}} \right)^2 \\ H_{10} = H_{30} + \left( \frac{Q_{10}}{a_{13}} \right)^2 \end{cases} \quad (1.0.2)$$

Pour linéariser le système autour de ce point d'équilibre nous devons faire un changement de base en prenant des variables suivantes

$$\begin{cases} h_2(t) = H_2(t) - H_{20} \\ h_3(t) = H_3(t) - H_{30} \\ h_1(t) = H_1(t) - H_{10} \\ q_1(t) = Q_1(t) - Q_{10} \\ q_2(t) = Q_2(t) - Q_{20} \end{cases} \quad (1.0.3)$$

comme des variations des vraies valeurs autour de point d'équilibre. Ensuite nous pouvons développer des équations différentielles dans une série de Taylor :

$$\begin{cases} \dot{h}_1 = \frac{1}{S} * \left( q_1 - \frac{a_{13}}{2\sqrt{H_{10}-H_{30}}} * h_1 + \frac{a_{13}}{2\sqrt{H_{10}-H_{30}}} * h_3 \right) \\ \dot{h}_3 = \frac{1}{S} * \left( \frac{a_{13}}{2\sqrt{H_{10}-H_{30}}} * h_1 + \left( -\frac{a_{13}}{2\sqrt{H_{10}-H_{30}}} - \frac{a_{32}}{2\sqrt{H_{30}-H_{20}}} \right) * h_3 + \frac{a_{32}}{2\sqrt{H_{30}-H_{20}}} * h_2 \right) \\ \dot{h}_2 = \frac{1}{S} * \left( q_2 + \frac{a_{32}}{2\sqrt{H_{30}-H_{20}}} * h_3 + \left( -\frac{a_{32}}{2\sqrt{H_{30}-H_{20}}} - \frac{a_{20}}{2\sqrt{H_{20}}} \right) * h_2 \right) \end{cases} \quad (1.0.4)$$

Nous avons enlevé les fonctions de signe comme nous utilisons l'hypothèse que  $H_1(t) \geq H_3(t) \geq H_2(t)$ , sinon il aurait fallu garder les fonction de signe et la linéarisation ne serait pas facile à calculer si tant est que possible.

En mettant les valeurs de résistance de flux  $R_{13} = \frac{2\sqrt{H_{10}-H_{30}}}{a_{13}}$ ,  $R_{32} = \frac{2\sqrt{H_{30}-H_{20}}}{a_{32}}$  et  $R_{20} = \frac{2\sqrt{H_{20}}}{a_{20}}$  nous obtenons le système linéaire dans la représentation d'espace d'état suivant :

$$\begin{cases} \dot{h}(t) = \begin{bmatrix} -\frac{1}{SR_{13}} & \frac{1}{SR_{13}} & 0 \\ \frac{1}{SR_{13}} & -\frac{1}{S} \left( \frac{1}{R_{13}} + \frac{1}{R_{32}} \right) & \frac{1}{SR_{32}} \\ 0 & \frac{1}{SR_{32}} & -\frac{1}{S} \left( \frac{1}{R_{32}} + \frac{1}{R_{20}} \right) \end{bmatrix} h(t) + \begin{bmatrix} \frac{1}{S} & 0 \\ 0 & 0 \\ 0 & \frac{1}{S} \end{bmatrix} \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} \\ y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} h(t) \end{cases} \quad (1.0.5)$$

avec  $h(t) = [h_1(t) \quad h_3(t) \quad h_2(t)]^T$

## 2 Analyse du procédé autour du point d'équilibre

### 2.1 Stabilité, commandabilité et observabilité

Le calcul des valeurs propres de matrice  $A$  nous a donné des v.p. à partie réelle négative ce qui nous indique que le vecteur d'équilibre est stable asymptotiquement pour le système autonome. C'est logique si on considère le système réel, étant donné que sans commande, toute l'eau va s'écouler dans le réservoir.

Le modèle est commandable à partir de l'entrée 1 (matrice de commandabilité associée à  $(A, B(:,1))$  de rang plein) et à partir de l'entrée 2 (matrice de commandabilité associée à  $(A, B(:,2))$  de rang plein). Donc le modèle est commandable avec les 2 entrées.

La matrice de commandabilité est suivante :

$$\begin{bmatrix} 64.9351 & 0 & -0.4668 & 0 & 0.0067 & 0.0035 \\ 0 & 0 & 0.4668 & 0.4826 & -0.0102 & -0.0198 \\ 0 & 64.9351 & 0 & -1.7160 & 0.0035 & 0.0489 \end{bmatrix} \quad (2.1.1)$$

Nous calculons maintenant les indices de commandabilité. Comme les trois colonnes de cette matrice sont linéairement indépendantes  $\rho_2 = 1$  et par conséquent  $\rho_1 = 2$ .

D'après l'hypothèse que  $H_1(t) \geq H_3(t) \geq H_2(t)$  ce résultat est cohérent puisque la somme d'indices de commandabilité est égale au rang de la matrice de commandabilité. De point de vue de physique c'est aussi logique comme dans notre cas où  $q_1 > q_2$  l'entrée  $q_1$  affecte sur la quantité d'eau dans les bacs 1 et 3, l'entrée  $q_2$  le fait pour le bac 2 respectivement.

Comme le système linéaire est obtenu grâce à l'hypothèse mentionnée précédemment nous sommes limités avec les commandes possibles de ce système. Par exemple, nous ne pouvons pas commander le système de sorte que la hauteur  $h_2$  augmente plus rapide que  $h_3$  en utilisant une entrée  $q_1$ . Il faut alors être précis avec les objectifs que nous pouvons nous poser.

Une commande par le retour d'état est possible dans notre cas, comme la matrice de commandabilité est de rang plein. Par contre, comme nous avons dit ci-dessus le choix de valeurs propres désirées doit être fait intelligemment et doit respecter les inégalités  $\lambda_1 \geq \lambda_3 \geq \lambda_2$  dans le cas où on a une entrée  $q_1$  ou deux entrées avec  $q_1 \geq q_2$ .

Si nous souhaitons commander le système de sorte que la sortie  $y = h_1$  arrive à la hauteur de 3 cm en moins de 5 secondes nous pouvons utiliser une commande suivante :

$$u(t) = B^T e^{-A^T t} * W^{-1}(0; t_f) \left( -x_0 + e^{-A t_f} x_f \right) \quad (2.1.2)$$

Si nous souhaitons appliquer cette commande pour les bacs qui sont vide au départ, elle va avoir la forme suivante :

$$u(t) = B^T e^{-A^T t} * W^{-1}(0; 5) * e^{-5A} * 0.03 \quad (2.1.3)$$

### 2.2 Étude de la fonction de transfert

Nous pouvons obtenir la matrice des fonctions de transfert avec Matlab :

$$G(s) = \begin{bmatrix} \frac{64.935(s+0.05533)(s+0.01941)}{(s+0.05582)(s+0.02719)(s+0.003715)} & \frac{0.009636}{(s+0.05582)(s+0.02719)(s+0.003715)} \end{bmatrix} \quad (2.2.1)$$

La forme de Smith-McMillan pour cette matrice de transfert nous obtenons facilement comme elle est de rang = 1 :

$$G_{SM}(s) = \begin{bmatrix} \frac{1}{(s+0.05582)(s+0.02719)(s+0.003715)} & 0 \end{bmatrix} \quad (2.2.2)$$

Cette matrice de transfert n'a pas de zéros, ses pôles nous pouvons voir clairement dans le dénumérateur de premier élément de la matrice.

Nous pouvons conclure que le système est stable BIBO comme les pôles de la forme Smith-McMillan sont réels et négatifs et égaux  $\pi_1 = -0.003715$ ,  $\pi_2 = -0.02719$ ,  $\pi_3 = -0.05582$

Le gain statique nous trouvons en utilisant la commande de Matlab `dcgain`.

Ce gain est égal  $10^4 * [1.2372 \quad 0.1709]$

Pour une entrée  $q_1(t) = 5 * 10^{-5}$  de type échelon nous allons regarder le comportement des hauteurs d'eau dans les bacs :

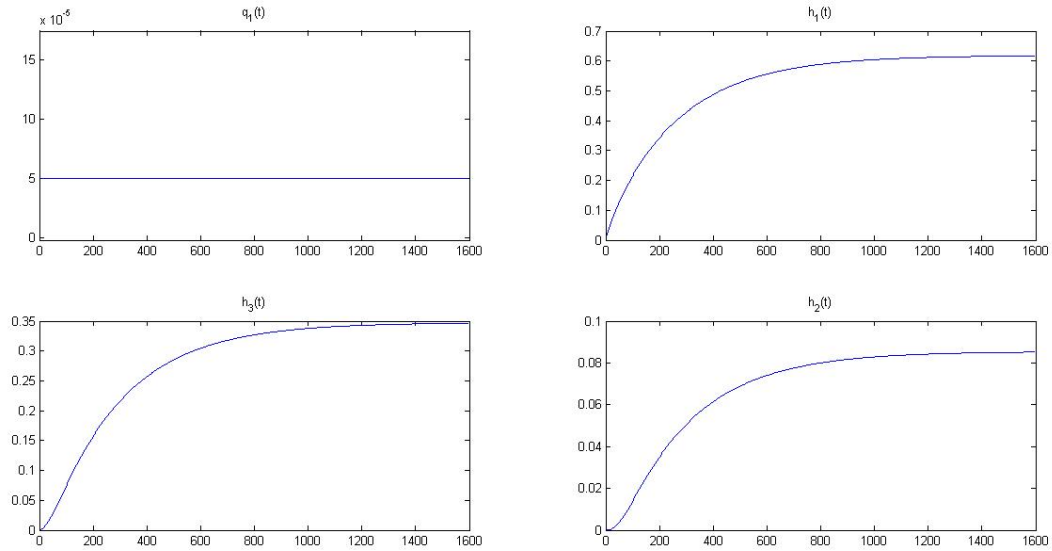


FIGURE 1 – Réponse du système à un échelon  $q_1(t) = 5 * 10^{-5}U(t)$

Nous voyons ici qu'on n'a pas de dépassement et que le transfert d'eau dans les bacs se fait consécutivement de bac 1 à bac 2 conformément à l'hypothèse  $H_1(t) \geq H_3(t) \geq H_2(t)$  de gauche à droite.

Les constantes de temps de 3 états sont suivantes :  $T_1 \approx 248$  s,  $T_3 \approx 303$  s,  $T_2 \approx 324$  s.

Le temps de montée de 3 états est suivant :  $t_{m1} = 2.2 * 248 = 545.6$  s,  $t_{m3} = 2.2 * 303 = 666.6$  s,  $t_{m2} = 2.2 * 324 = 712.8$  s.

Le temps de réponse de 3 états est suivant :  $t_{r1} = 3 * 248 = 744$  s,  $t_{r3} = 3 * 303 = 909$  s,  $t_{r2} = 3 * 324 = 972$  s.

Pour une entrée  $q_2(t) = 2 * 10^{-5}$  de type échelon nous allons regarder le comportement des hauteurs d'eau dans les bacs :

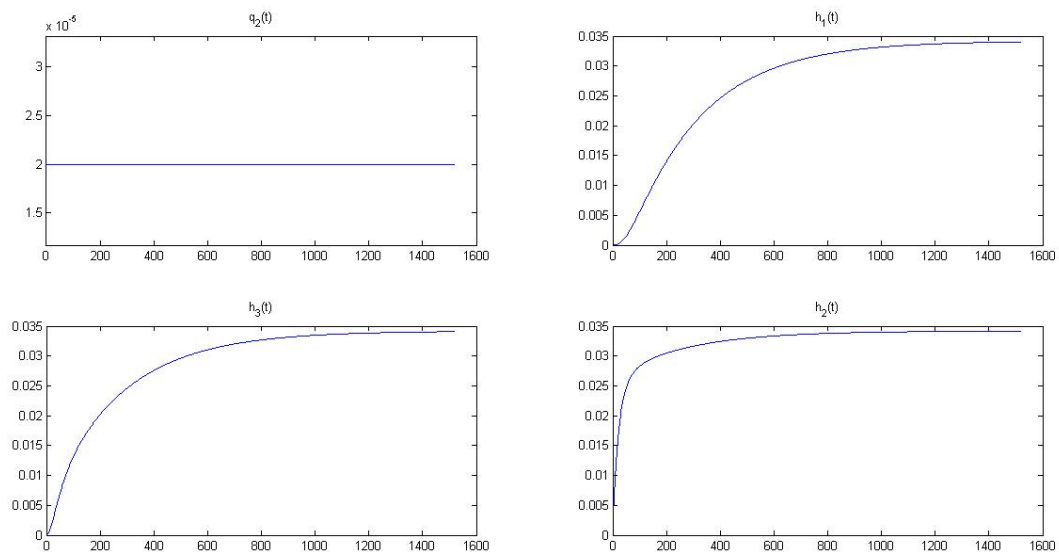


FIGURE 2 – Réponse du système à un échelon  $q_2(t) = 2 * 10^{-5}U(t)$

Nous voyons ici qu'on n'a pas non plus de dépassement et le changement des hauteurs d'eau dans les bacs se fait consécutivement de bac 2 à bac 1 conformément à l'hypothèse inverse  $H_1(t) \leq H_3(t) \leq H_2(t)$ . Si nous regardons dans Matlab les valeurs maximales des hauteurs

d'eau nous verrons que la valeur maximale de  $h_2$  est presque égale à celle de  $h_3$  et ainsi de suite. C'est normal, comme l'entrée  $q_2$  agit sur le bac 2 duquel l'eau s'écoule dans le réservoir, pourtant le transfert d'eau se fait de droite à gauche dans ce cas. Malgré cela, les caractéristiques temporelles ne sont pas les mêmes.

Les constantes de temps de 3 états sont suivantes :  $T_1 \approx 322$  s,  $T_3 \approx 224$  s,  $T_2$  ne se calcule pas comme la hauteur  $h_2$  ne se comporte pas de manière apériodique.

Le temps de montée de 3 états est suivant :  $t_{m1} = 2.2 * 322 = 708.4$  s,  $t_{m3} = 2.2 * 224 = 492.8$  s,  $t_{m2} = 285$  s.

Le temps de réponse de 3 états est suivant :  $t_{r1} = 3 * 322 = 966$  s,  $t_{r3} = 3 * 224 = 674$  s,  $t_{r2} = 387$  s.

Nous pouvons conclure que pour l'entrée  $q_2$  la hauteur dans le bac 2 ne se comporte pas pareil qu'avec l'entrée  $q_1$  mais logiquement elle se stabilise plus rapidement que les hauteurs dans les deux autres bacs.

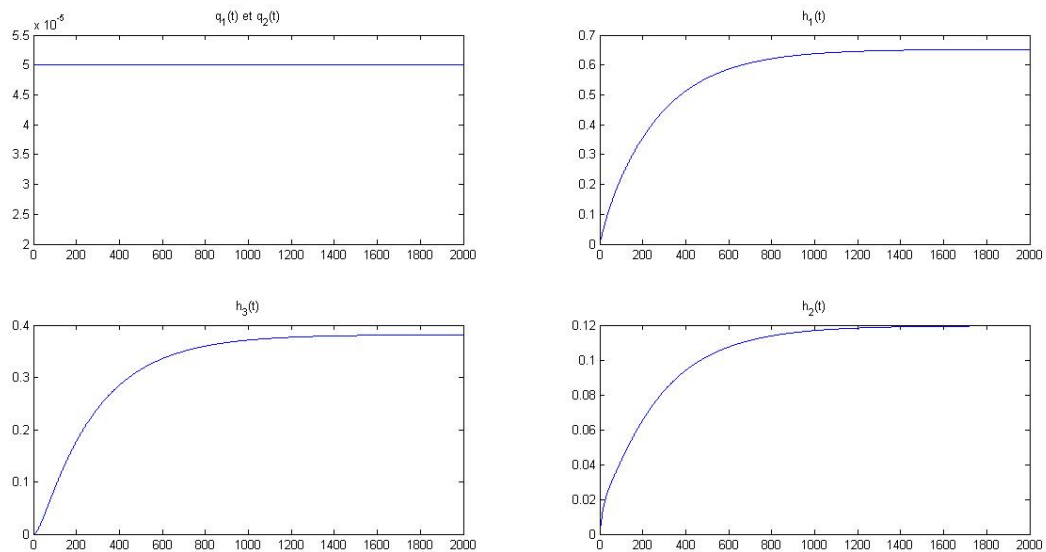


FIGURE 3 – Réponse du système à deux échelons  $q_1(t)$  et  $q_2(t)$

Finalement, pour deux entrées envoyées simultanément nous pouvons voir que comme le débit d'entrée  $q_1 > q_2$  le premier bac se remplit plus rapidement que les autres et le deuxième bac se remplit plus lentement que les autres comme il est pas affecté directement par une entrée.

Les constantes de temps de 3 états sont suivantes :  $T_1 \approx 253$  s,  $T_3 \approx 298$  s,  $T_2 \approx 255$  s.

Le temps de montée de 3 états est suivant :  $t_{m1} = 2.2 * 253 = 556.6$  s,  $t_{m3} = 2.2 * 298 = 655.6$  s,  $t_{m2} = 2.2 * 255 = 561$  s.

Le temps de réponse de 3 états est suivant :  $t_{r1} = 3 * 253 = 759$  s,  $t_{r3} = 3 * 298 = 894$  s,  $t_{r2} = 3 * 255 = 765$  s.

### 3 Asservissement du procédé

On s'intéresse à la commande du niveau d'eau dans le bac numéro 1

#### 3.1 Cahier des charges

Les spécifications désirées sur le système sont :

1. Le temps de réponse  $t_r$  de la sortie doit respecter  $t_r \leq 90$  s
2. L'erreur doit être nulle en régime permanent
3. Le système doit rester dans le cadre d'une étude linéaire, c'est à dire que les vannes ne doivent pas saturer :  $0 \leq Q_1(t); Q_2(t) \leq Q_{max}$
4. La consigne, notée  $w(t)$ , est un échelon de niveau exprimé en  $m$ . On pose  $w(t) = 0.05U(t)$ .

#### 3.2 Première approche par retour d'état - méthode de Bass-Gura

Selon le cahier des charges nous devons choisir les valeurs propres désirées pour que temps de réponse du système soit inférieure à 90 secondes. Autrement dit, nous voulons que chaque pôles du système soit  $< -\frac{1}{30}$ . De telle façon nous choisissons les pôles désirés suivants :  $[-0.05; -0.04; -0.035]$

Nous choisissons arbitrairement le vecteur  $q = [2 \ 1]^T$ . Nous avons plusieurs choix possibles comme le couple  $(A, Bq)$  est commandable pour plusieurs valeurs de  $q$ . Pour notre choix, la matrice de commandabilité est de rang plein alors nous avons droit de prendre ce vecteur.

Nous trouvons ensuite le gain  $k$  en utilisant la commande de Matlab *acker* pour placer nos valeurs propres désirés. On obtient  $k = 10^{-3} * [0.2615 \ 0.2663 \ 0.0664]$ . Pour utiliser le retour d'état il faut utiliser le gain  $K = kq$  qu'on calcule en utilisant Matlab.

Nous avons obtenu le système avec le retour d'état en prenant pour la matrice  $A$  une nouvelle matrice  $A - BK$ , pourtant il nous reste d'annuler l'erreur statique.

Pour ce faire, nous implémentons un pré-compensateur  $N$  qui nous assurera une erreur de position nulle dans tout les cas de figure.

Le pré-compensateur se présente comme étant une matrice  $2 \times 1$  telle que on ait la relation

$$\frac{y}{y_{ref}} = F_t N \quad avec \quad N = \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} \quad et \quad F_t = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \quad (3.2.1)$$

Ainsi, avoir une erreur de position nulle nous ramène à l'équation matricielle.

$$F_t(0)N = 1 \iff N_2 = \frac{1 - F_1(0)N_1}{F_2(0)} \quad (3.2.2)$$

Ainsi, on voit qu'il existe une infinité de solutions possibles pour assurer une erreur de position nulle. Il nous faut donc choisir un  $N_1$ . On prendra  $N_1$  égal à l'inverse du gain statique du à la 1<sup>re</sup> entrée afin d'avoir des valeurs de dépassement admissibles.

Avec le pré-compensateur alors déterminé, nous obtenons les résultats suivants :

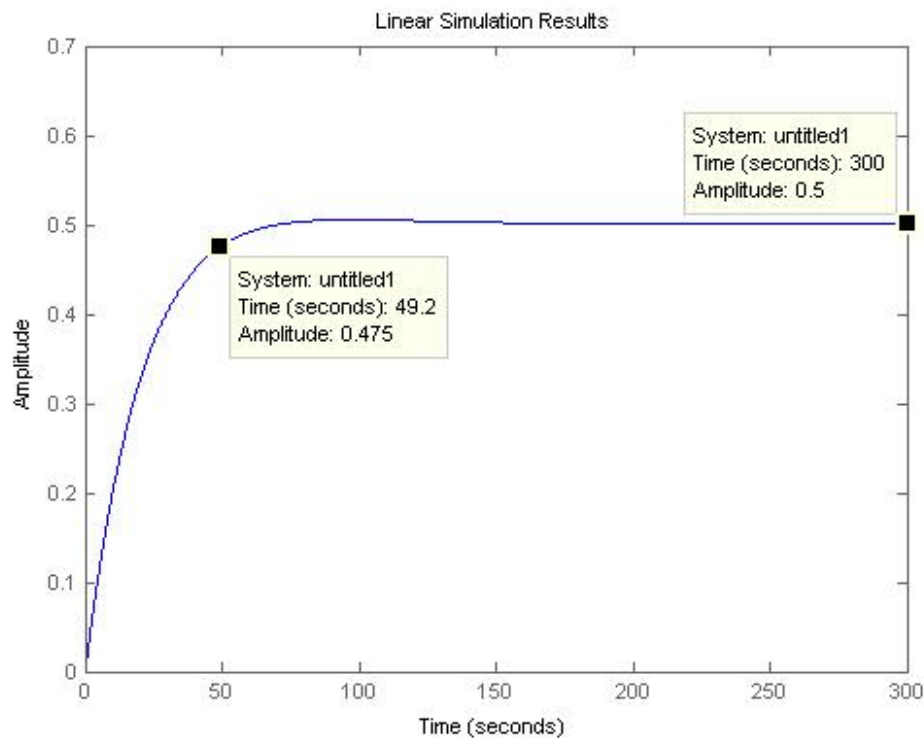
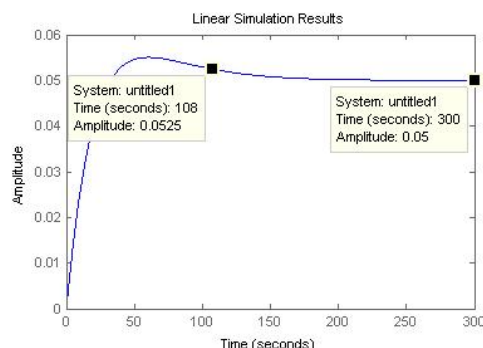
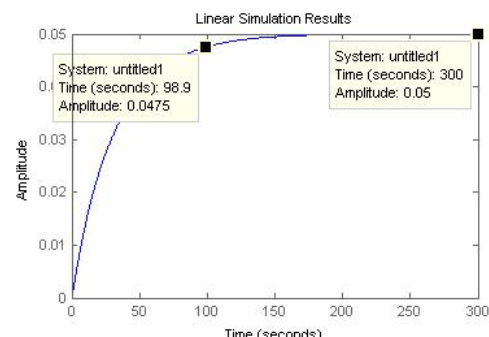


FIGURE 4 – Réponse de notre Système à une consigne de 0.05 m.

On voit bien que les différentes demandes du cahier des charges sont respectées, il n'y a pas d'erreur de position, et le temps de réponse à 5% est bien inférieur à 90 secondes.

On décide de regarder l'influence du choix de  $q$  sur la réponse du système, nous réalisons les tests suivant : Pour le faire nous pouvons regarder le gain statique de notre système qui est égal  $10^3 * [1.2925 \quad -0.4547]$ . Nous pouvons en déduire un pré-compensateur qui va prendre des valeurs inverses à ceux du gain statique.

Pour vérifier notre estimation nous allons simuler notre système en boucle fermée en utilisant des différents vecteurs  $q$ .

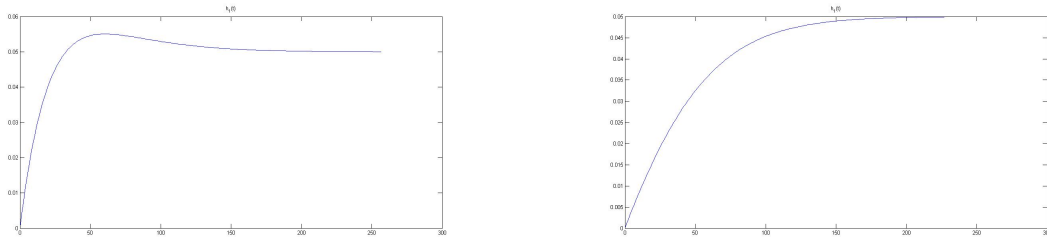
(a) Réponse du système avec  $q = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ (b) Réponse du système avec  $q = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$ FIGURE 5 – Comparaison des différentes dynamiques possibles pour différentes valeurs de  $q$ 

Comme nous pouvons voir, les caractéristiques temporelles varient si nous changeons le vecteur  $q$  ce qui veut dire que nous pouvons obtenir des meilleures performances du système en choisissant un vecteur  $q$  "spécial", mais il faut toujours vérifier que le système reste commandable avec nos choix.



Si on prend un vecteur  $q = [\alpha \ 0]^T$  où  $\alpha$  est un choix adéquat, l'asservissement dans ce cas va être fait que sur l'entrée  $q_1$ .

Également, si on souhaite utiliser uniquement la seconde pompe pour asservir le système il faut prendre un vecteur  $q = [0 \ \alpha]^T$ .



(a) Sortie  $h_1$  du système avec entrée  $q_1$  asservie. Le vecteur  $q = [2 \ 0]^T$  (b) Sortie  $h_1$  du système avec entrée  $q_2$  asservie. Le vecteur  $q = [0 \ 2]^T$

FIGURE 6 – Simulation du système avec deux entrées, en prenant comme la référence un échelon  $\omega(t) = 0.05U(t)$

Pour conclure sur la méthode Bass-Gura nous tenons à dire que nous pouvons asservir un système MIMO de la même manière qu'un système SISO grâce à cette méthode et technique de retour d'état. Le choix de vecteur  $q$  change les caractéristiques temporelles et surtout le fait d'asservir une ou plusieurs entrées. En effet, cela dépend de système réel, si l'on a la possibilité d'asservir toutes les entrées ou pas. Cette méthode nous permet de nous "débrouiller" dans ces situations en faisant des asservissements différents mais toujours en tenant compte de structure du système réel. Le principal avantage de cette méthode est donc que nous avons un large éventail de choix pour la commande que nous souhaitons appliquer.

### 3.3 Seconde approche par retour d'état - mise sous forme commandable par blocs

Cette méthode se base sur le calcul de la matrice de commandabilité. Matrice que nous avons déjà obtenu dans la partie 2.1.

A partir de cette matrice et à l'aide des indices de commandabilité, il nous est possible de former  $C_{0b}$  décrite par

$$C_{0b} = [b_1 \quad Ab_1 \quad b_2] \quad (3.3.1)$$

Nous obtenons :

$$C_{0b} = \begin{bmatrix} 64.9351 & -0.7779 & 0 \\ 0 & 0.7779 & 0 \\ 0 & 0 & 64.9351 \end{bmatrix}$$

En inversant cette matrice, on peut obtenir la matrice  $P$  qui sera la matrice de changement de base nous permettant de calculer la forme de commande compagne par blocs avec la relation

$$P = \begin{bmatrix} C_{0b}^{-1}(2,:) \\ C_{0b}^{-1}(2,:)A \\ C_{0b}^{-1}(3,:) \end{bmatrix} \quad (3.3.2)$$

Nous obtenons la matrice suivante :

$$P = \begin{bmatrix} 0 & 1.2855 & 0 \\ 0.0154 & -0.0313 & 0.0159 \\ 0 & 0 & 0.0154 \end{bmatrix}$$

On peut alors calculer notre forme de commande compagne par blocs

$$\begin{cases} A_c = PAP^{-1} \\ B_c = PB \end{cases} \quad (3.3.3)$$

On obtient

$$A_c = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.0363 & -0.0397 \\ 0.0001 & 0 & -0.0504 \end{bmatrix} \quad B_c = \begin{bmatrix} 0 & 0 \\ 1 & 1.0339 \\ 0 & 1 \end{bmatrix}$$

On voit que la matrice  $B$  n'est pas diagonale par blocs, on cherche donc un pré-compensateur  $R$  qui nous permettra de la rendre diagonale.  $R$  est telle que

$$B_{cc} = B_c R \quad (3.3.4)$$

avec

$$B_{cc} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

On trouve

$$R = \begin{bmatrix} 1 & -1.0339 \\ 0 & 1 \end{bmatrix}$$

On peut dès lors calculer un retour d'état  $\tilde{K}$  tel que l'on ait les valeurs propres désirées. On utilise la commande "place" sur le système  $(A_c, B_{cc})$  et en considérant les même valeurs propres que dans la partie 3.2. Nous obtenons la matrice  $\tilde{K}$  suivante

$$\tilde{K} = \begin{bmatrix} 0.0014 & 0.0387 & -0.0397 \\ 0.0001 & 0 & -0.0004 \end{bmatrix}$$

On peut alors déduire la valeur du gain de retour d'état dans la base initiale avec

$$K = K_c P \quad (3.3.5)$$

On obtient

$$K = 1 \times 10^{-3} \begin{bmatrix} 0.5953 & 0.5954 & 0.0041 \\ 0 & 0.1908 & -0.058 \end{bmatrix}$$

Ce gain nous permet alors de calculer le pré-compensateur nécessaire à assurer une erreur de position nulle de la même manière que dans la méthode de Bass-Gura.

Nous simulons donc notre système final obtenu, et nous obtenons la réponse suivante :

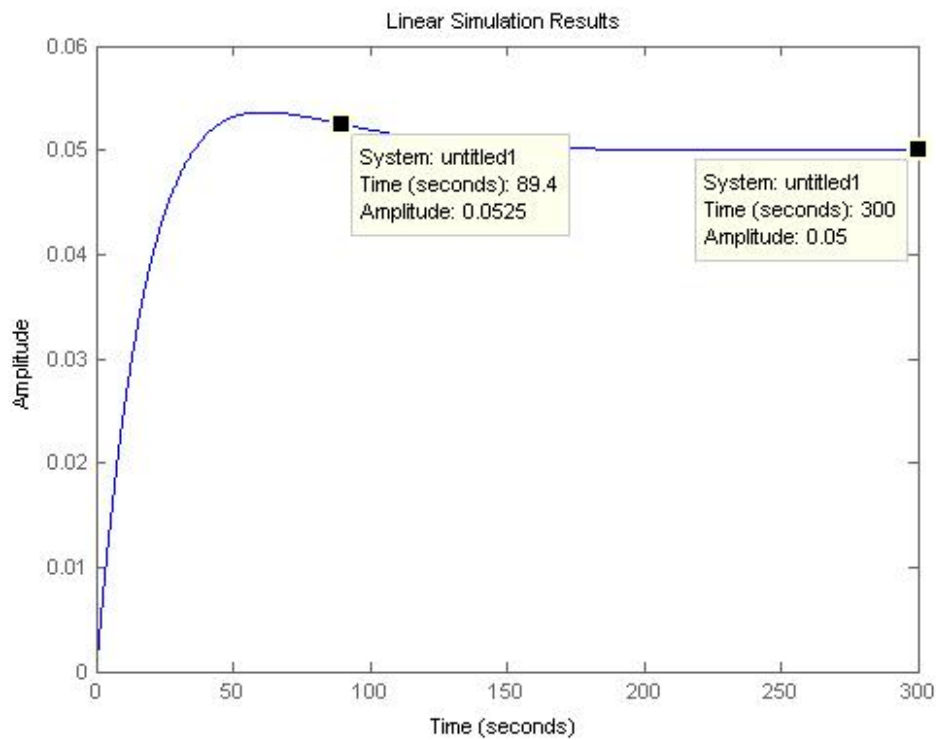


FIGURE 7 – Réponse de notre Système corrigé avec un retour d'état réalisé avec la méthode de mise en forme commandable par blocs à une consigne de 0.05 m.

On observe sur la figure que cette commande respecte le cahier des charges mais les résultats semblent moins bons que ceux d'une méthode de Bass-Gura où les paramètres ont bien été trouvés.

On teste une dernière méthode, on tente d'utiliser directement la méthode de implémentée dans Matlab en utilisant la commande "place". On obtient alors la réponse suivante :

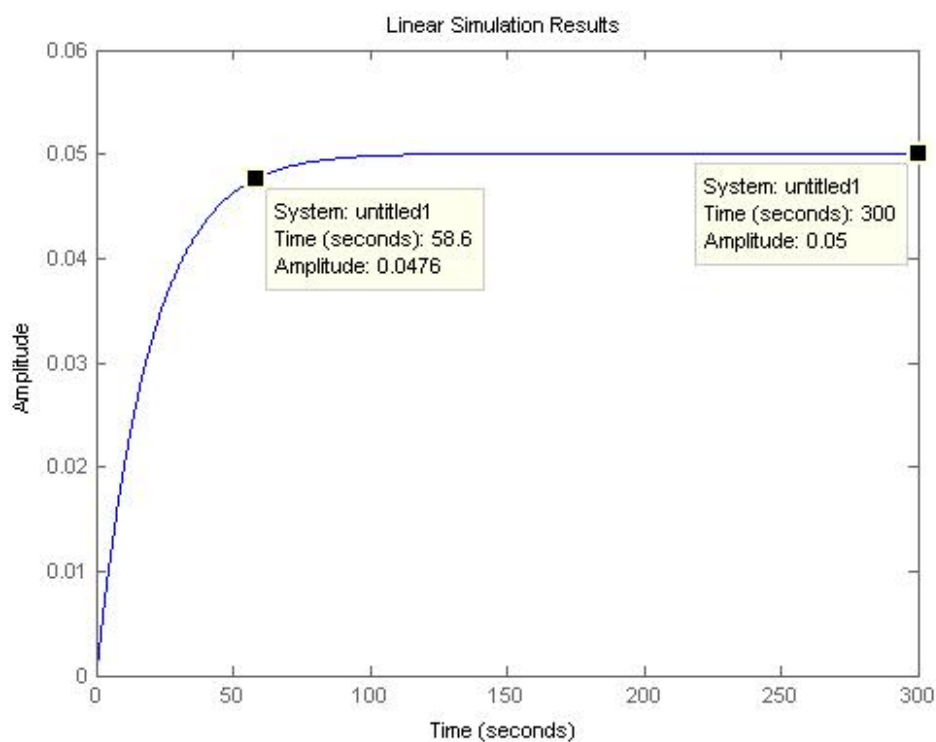


FIGURE 8 – Réponse de notre Système corrigé avec la commande "place" de Matlab à une consigne de 0.05 m.

Il semble que cette dernière méthode soit meilleure, en effet, elle respecte parfaitement le cahier des charges et est plus rapide. Il faut cependant faire attention à ce que les débits maximaux acceptables soient respectés. C'est aussi une méthode plus rapide à mettre en place si l'on possède matlab.

## 4 Conclusion

Nous pouvons maintenant conclure sur ce TP de commande avancée. Nous avons pu voir la mise en place de différentes méthodes de correction par retour d'état déjà vue en cours, et l'outil qu'est Matlab nous a permis de pouvoir les comparer et ainsi, identifier leurs points forts et leurs faiblesses.