
AUT202 - Dynamique et Contrôle des Systèmes

Travail Pratique

12 septembre 2024

Guilherme Nunes Trofino
2022-2024

Table des matières

1	Introduction	2
1.1	Information Matier	2
2	Partie 2 - Modélisation	5
2.1	Question 4	5
2.2	Question 5	6
2.3	Question 6	7
2.4	Question 7	8
2.5	Question 8	9
2.6	Question 9	10
2.7	Question 10	11
3	Partie 3 - Contrôleur	12
3.1	Question 11	12
3.2	Question 12	13
3.3	Question 13	14
3.4	Question 14	15
3.5	Question 15	16
3.6	Question 16	17
3.7	Question 17	18
4	Partie 4 - Observateur	19
4.1	Question 18	20
4.2	Question 19	21
4.3	Question 20	22
4.4	Question 21	23
4.5	Question 22	24
5	Partie 5 - Observateur - Contrôleur	25
5.1	Question 23	26
5.2	Question 24	27
5.3	Question 25	28

1. Introduction

Repository Hello! My name is Guilherme Nunes Trofino and this is my LaTeX notebook of AUT202 - Dynamique et Contrôle des Systèmes that can be found in my GitHub repository : https://github.com/tr0fin0/classes_ensta.

Disclaimer This notebook is made so it may help others in this subject and is not intend to be used to cheat on tests so use it by your on risk.

Suggestions If you may find something on this document that does not seam correct please reach me by e-mail : guitrofino@gmail.com.

1.1. Information Matier

Référence Dans cette matière le but sera de comprendre comment une Système d'Exploitation marche.

Partie 1 - Equation de Lagrange

Présentation Ce Travail Pratique évaluera la capacité d'entant donné une système le modéliser, le proposer un contrôleur et un observateur et, à la fin, les intégrées. Pendant ce development, des simulations MATLAB et Simulink seront proposées pour valider le development et permettre de visualiser les résultats.

Résultats Comme le but de ce travail est la modélisation du système le development des equations qui décrivent le système seront déjà connus. Sur ce Travail Pratique les Questions 1, 2 et 3 ont les résultats nécessaires pour le déroulement du projet et seront rappelées à la suite.

Problème On s'intéresse à la stabilisation d'une balle libre de rouler sur un plateau pivotant comme montrer dans la figure suivante :

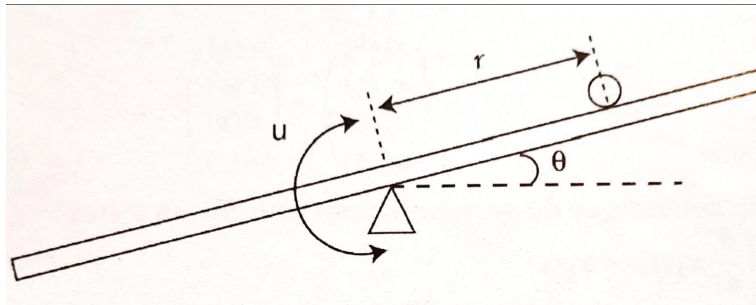


FIGURE 1.1 : Système d'Intéresse

Où :

1. J_p , **Moment d'Inertie du Plateau** par rapport à on axe de rotation ;
2. J_b , **Moment d'Inertie de la Balle** par rapport à son centre ;

$$J_b = \sigma \cdot m \cdot R^2, \quad 0 < \sigma \leq 1 \quad (1.1)$$

Où :

- (a) σ , **Densité** en fonction du rayon ;
3. R , **Rayon de la Balle** ;
4. m , **Masse de la Balle** ;
5. g , **Accélération Gravitationnelle** ;

Avec :

1. $r(t)$, **Distance de la Balle** par rapport au Centre du Plateau sur le temps ;
2. $\theta(t)$, **Angle du Plateau** par rapport son Centre de Rotation sur le temps ;
3. $u(t)$, **Commande du Système** ;

On souhaite maintenir la Balle sur la position r_{ref} à partir des données obtenus des capteurs $r(t)$ et $\theta(t)$ en utilisant la commande $u(t)$ qui contrôle l'angle du Plateau.

Remarque. Même que le document utilisé barre et balle pour nommer les partis du système, on appellera la barre comme plateau pour que les sous-écrits soit différent en facilitant la comprehension de la résolution.

Remarque. On considère que les angles de commande seront petits car la puissance disponibles pour la commande sont petits.

Question 1

Exercice 1.1. Montrer que l'énergie cinétique du système est donnée par :

$$E_c = \frac{1}{2} \cdot J_p \cdot \dot{\theta}^2(t) + \frac{1}{2} \cdot m \cdot (\dot{r}^2(t) + r^2(t) \cdot \dot{\theta}^2(t)) + \frac{1}{2} \cdot J_b \cdot \frac{\dot{r}^2}{R^2} \quad (1.2)$$

Remarque. On négligera dans l'énergie cinétique de rotation de la balle sur elle-même, c'est-à-dire : l'effet de la rotation de la barre $\theta(t)$.

Question 2

Exercice 1.2. Calculer l'énergie potentielle E_p en fonction de $r(t)$ et $\theta(t)$.

Remarque. On supposera que l'axe de rotation est au niveau du centre de gravité du plateau.

Résolution. L'Énergie Potentielle sera donnée par l'équation suivante :

$$E_p = m \cdot g \cdot r \cdot \sin(\theta(t)) \quad (1.3)$$

Question 3

Exercice 1.3. Dédurre de ce qui précède les équations de Lagrange suivantes :

$$(1 + \sigma) \cdot \frac{d}{dt} \dot{r}(t) = r(t) \cdot \dot{\theta}^2(t) - g \cdot \sin(\theta(t)) \quad (1.4)$$

$$\frac{d}{dt} (J_p + m \cdot r^2(t)) \cdot \dot{\theta}(t) = -m \cdot g \cdot r(t) \cdot \cos(\theta(t)) + u(t) \quad (1.5)$$

2. Partie 2 - Modélisation

États À la suite de cette exercice on utilisera la notation suivant pour les états du système utilisés pour résoudre le système :

$$X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} = \begin{bmatrix} r(t) \\ \dot{r}(t) \\ \theta(t) \\ \dot{\theta}(t) \end{bmatrix} \quad (2.1)$$

2.1. Question 4

Exercice 2.1. Montrer que la dynamique du système s'écrit sous forme d'état :

$$\begin{cases} \frac{d}{dt} x_1(t) = x_2(t) \\ \frac{d}{dt} x_2(t) = \frac{x_1(t) \cdot x_4^2(t) - g \cdot \sin(x_3(t))}{1 + \sigma} \\ \frac{d}{dt} x_3(t) = x_4(t) \\ \frac{d}{dt} x_4(t) = \frac{u(t) - m \cdot g \cdot x_1(t) \cos(x_3(t)) - 2m \cdot x_1(t) \cdot x_2(t) \cdot x_4(t)}{J_p + m \cdot x_1^2(t)} \end{cases} \quad (2.2)$$

Résolution. On note, par la définition des coordonnées d'état, que les états suivants sont triviales :

$$\boxed{\frac{d}{dt} x_1(t) = x_2(t)} \quad \boxed{\frac{d}{dt} x_3(t) = x_4(t)}$$

Ensuite, si on part de l'Équation (1.4) en faisant la substitution des variables dynamiques pour les variables d'état on aura :

$$(1 + \sigma) \cdot \frac{d}{dt} x_2(t) = x_1(t) \cdot x_4^2(t) - g \cdot \sin(x_3(t))$$

$$\boxed{\frac{d}{dt} x_2(t) = \frac{x_1(t) \cdot x_4^2(t) - g \cdot \sin(x_3(t))}{(1 + \sigma)}}$$

Finalement, si on part de l'Équation (1.5) en faisant la substitution des variables dynamiques pour les variables d'état on aura :

$$\frac{d}{dt} (J_p + m \cdot x_1^2(t)) \cdot x_4(t) = -m \cdot g \cdot x_1(t) \cdot \cos(x_3(t)) + u(t)$$

$$J_p \frac{d}{dt} x_4(t) + m \frac{d}{dt} (x_1^2(t) \cdot x_4(t)) =$$

$$J_p \frac{d}{dt} x_4(t) + 2m x_1(t) x_2(t) x_4(t) + m x_1^2(t) \frac{d}{dt} x_4(t) =$$

$$(J_p + m x_1^2(t)) \frac{d}{dt} x_4(t) = u(t) - m g x_1(t) \cos(x_3(t)) - 2m x_1(t) x_2(t) x_4(t)$$

$$\boxed{\frac{d}{dt} x_4(t) = \frac{u(t) - m g x_1(t) \cos(x_3(t)) - 2m x_1(t) x_2(t) x_4(t)}{(J_p + m x_1^2(t))}}$$

2.2. Question 5

Exercice 2.2. Modéliser le système dans le module Simulink de MATLAB. On prendra les valeurs suivantes pour les paramètres du système :

$$\begin{cases} J_p = 0.02 \text{ kg m}^2 \\ m = 0.6 \text{ kg} \\ \sigma = 0.8 \\ g = 9.81 \text{ ms}^{-2} \end{cases} \quad (2.3)$$

Résolution. Après obtenir les équations de la dynamique du système sous forme d'état le système suivant a été mise en place en Simulink pour modéliser les équations :

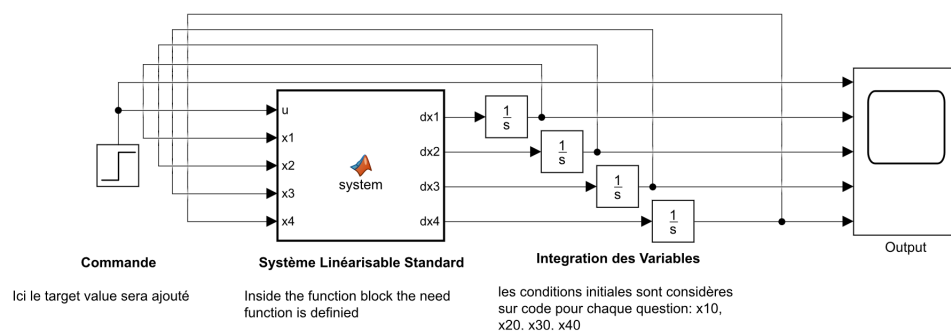


FIGURE 2.1 : Système d'Intéresse en Simulink, System

En utilisant le code suivant pour initialiser les variables nécessaires :

```
1 function [J, m, sig, g] = Q5()
2     J = 0.02; % inercia moment [kg.m2];
3     m = 0.6; % mass [kg]
4     sig = 0.8; % constant [-]
5     g = 9.81; % gravity [m/s2]
6 end
```

Dedans le bloque que represent `system` on ajoute le code suivant :

```
1 function [dx1, dx2, dx3, dx4] = system(u, x1, x2, x3, x4)
2     [J, m, sig, g] = Q5();
3
4     dx1 = x2;
5     dx2 = (x1*(x2)^2 - g*sin(x3))/(1+sig);
6     dx3 = x4;
7     dx4 = (u - m*g*x1*cos(x3) - 2*m*x1*x2*x4)/(J + m*x1^2);
```

Remarque. Comme les conditions du système ne changent pas les variables ajoutés sur chaque bloque de fonction seront données pour la fonction 5.

2.3. Question 6

Exercice 2.3. Simuler la dynamique du système avec une commande nulle pour différentes conditions initiales. Que constate-t-on ?

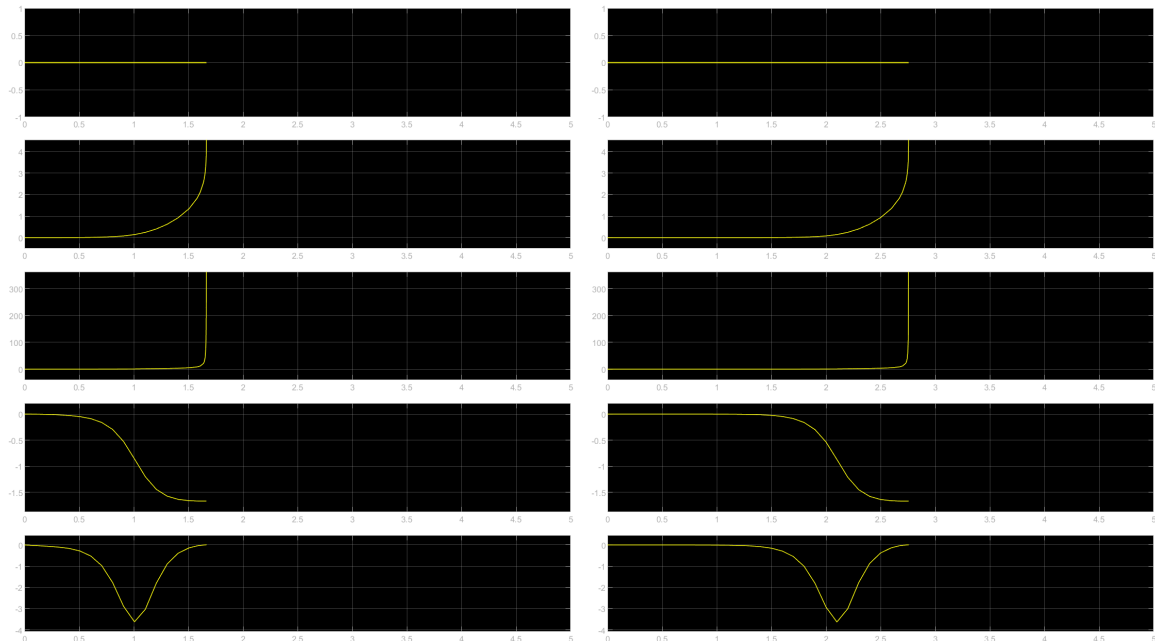
Résolution. En utilisant le code suivant on a une commande nulle, conditions initiales nulles, pour le système :

```
1 function [I1, I2] = P2(x10, x20, x30, x40)
2     I1 = [x10, x20, x30, x40]; % initial variables
3     I2 = [0, 0, 0, 0];
4 end
```

```
1 [J, m, sig, g] = Q5();
2 ref = 0.2;
3 [I1, I2] = P2(0.001, 0, 0, 0);
4 part2_simu
5 print('-spart2_simu', '-dpdf', 'model2.pdf')
```

```
1
2 elseif model == 4
```

Les variables I1 et I2 sont utilisés pour l'initialisation des blocs d'intégration. Finalement on a les résultats suivants :



(a) with $x_1(0) = 0.001$

(b) with $x_1(0) = 0.000001$

FIGURE 2.2 : Simulation Conditions Initiales, $x_2(0) = x_3(0) = x_4(0) = 0$

On note que, indépendamment des variables initiales choisîtes, le système diverge après quelques instants de simulation.

Phrase. Dans la suite de cette rapport sur les images de simulation on considère que le premier graphique en haut sera la commande $u(t)$ et à la suite sont les état $x_1(t)$, $x_2(t)$, $x_3(t)$ et $x_4(t)$.

2.4. Question 7

Exercice 2.4. Quels sont les points d'équilibre du système? Peut-on stabiliser la balle sur n'importe quelle position de r_{ref} ?

Exprimer l'état d'équilibre et la commande d'équilibre en fonction de la position r_{ref} sur laquelle on souhaite stabiliser la balle.

Résolution. Si on considère une système donne pour $X(t) = f(X(t), u(t))$ l'équilibre se fait quand $f(X_e(t), u_e(t)) = 0$ donc si on considère $x_1(t) = r_{\text{ref}}$ on a :

$$\begin{cases} \frac{d}{dt}x_1(t) = 0 = x_2(t) & \boxed{x_2(t) = 0} \\ \frac{d}{dt}x_2(t) = 0 = \frac{r_{\text{ref}} x_4^2(t) - g \sin(x_3(t))}{1 + \sigma} & \sin(x_3(t)) = 0 \therefore \boxed{x_3(t) = 0 = 0[2\pi]} \\ \frac{d}{dt}x_3(t) = 0 = x_4(t) & \boxed{x_4(t) = 0} \\ \frac{d}{dt}x_4(t) = 0 = \frac{u(t) - m g r_{\text{ref}} \cos(x_3(t)) - 2m r_{\text{ref}} x_2(t) x_4(t)}{J_p + m r_{\text{ref}}^2} & \boxed{u(t) = m g r_{\text{ref}}} \end{cases}$$

Finalement on note l'état d'équilibre $X_e(t)$ sera donne pour :

$$X_e(t) = \begin{bmatrix} x_{1_e} \\ x_{2_e} \\ x_{3_e} \\ x_{4_e} \end{bmatrix} = \begin{bmatrix} r_{\text{ref}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{e} \quad u_e(t) = m g r_{\text{ref}} \quad (2.4)$$

On note que tous les points d'équilibre du système seront données en fonction de r_{ref} si on a $X_e(t)$ et $u_e(t)$.

Remarque. Il faut noter que pour $x_3(t)$ il y avait deux solutions possibles :

$$\sin(x_3(t)) = 0 \quad \begin{cases} x_3(t) = 0[2\pi] \\ x_3(t) = \pi[2\pi] \end{cases}$$

Où $\theta[2\pi]$ représente tous les positions possibles d'obtenir l'angle θ dans le cercle trigonométrique.

La réponse $x_3(t) = \pi[2\pi]$ représente le plateau à l'envers, avec une angle de 180° , qui n'a pas de signification physique pour ce système et pour cette raison n'était pas considérée comme une réponse valable.

2.5. Question 8

Exercice 2.5. Ajouter dans le modèle Simulink un bloc permettant de calculer l'état d'équilibre X_{ref} et la commande d'équilibre u_{ref} en fonction de la position r_{ref} sur laquelle on souhaite stabiliser la Balle. Vérifier avec le modèle Simulink pour différentes valeurs de r_{ref} que si le système n'est pas initialisé exactement à l'équilibre, les trajectoires divergent.

Résolution. Après obtenir les équations de l'équilibre du système la fonction suivante a été mise en place en Simulink :

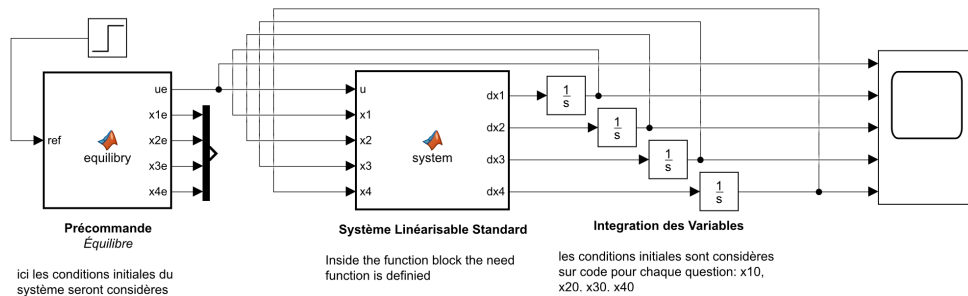
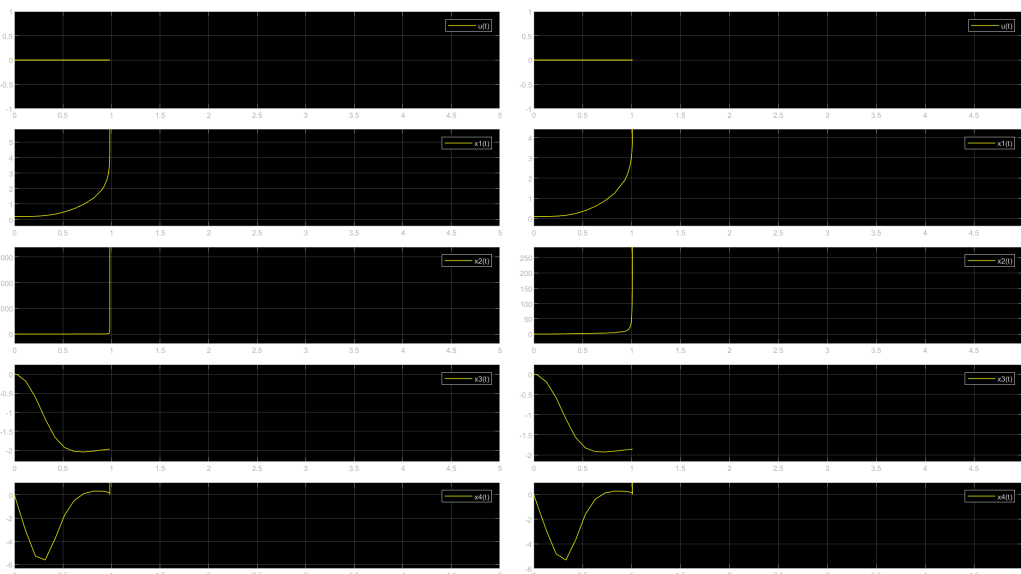


FIGURE 2.3 : Système d'Intérêt en Simulink, Équilibre

Dedans le bloque que represent `equilibry` on ajoute le code suivant :

```
1 function [ue, x1e, x2e, x3e, x4e] = equilibry(ref)
2     x1e = ref;
3     x2e = 0;
4     x3e = 0;
5     x4e = 0;
6     ue = m*g*x1e;
```

Qui donne des résultats suivants :



(a) with $x_1(0) = 0.2$ and $r_{\text{ref}} = 0.2$

(b) with $x_1(0) = 0.1$ and $r_{\text{ref}} = 0.2$

FIGURE 2.4 : Simulation Équilibre, $x_2(0) = x_3(0) = x_4(0) = 0$

On note que, hors de l'équilibre, le système diverge comme attendu.

2.6. Question 9

On note $X(t) = X_{\text{ref}} + \delta X(t)$ et $u(t) = u_{\text{ref}} + \delta u(t)$ sont les écarts à l'équilibre. On s'intéresse à l'équilibre associé à $r_{\text{ref}} = 0$.

Exercice 2.6. Montrer que le linéarisé tangent autour de la position d'équilibre $r_{\text{ref}} = 0$ s'écrit :

$$\underbrace{\frac{d}{dt} \begin{bmatrix} \delta x_1(t) \\ \delta x_2(t) \\ \delta x_3(t) \\ \delta x_4(t) \end{bmatrix}}_{\delta \dot{X}(t)} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{g}{1+\sigma} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{J_p} & 0 & 0 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} \delta x_1(t) \\ \delta x_2(t) \\ \delta x_3(t) \\ \delta x_4(t) \end{bmatrix}}_{\delta X(t)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_p} \end{bmatrix}}_B \delta u(t) \quad (2.5)$$

Résolution. On part de la solution obtenue sur la Question 7 :

$$\underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}}_{X(t)} \rightarrow \begin{cases} \frac{d}{dt} x_1(t) = x_2(t) \\ \frac{d}{dt} x_2(t) = \frac{x_1(t) x_4^2(t) - g \sin(x_3(t))}{1 + \sigma} \\ \frac{d}{dt} x_3(t) = x_4(t) \\ \frac{d}{dt} x_4(t) = \frac{u(t) - m g x_1(t) \cos(x_3(t)) - 2m x_1(t) x_2(t) x_4(t)}{J_p + m x_1^2(t)} \end{cases}$$

Pour la linéarisation les approximations suivantes seront considérées

$$\sin(\alpha) \approx \alpha \quad \cos(\alpha) \approx 1 \quad x^2(t) \approx 0 \quad (2.6)$$

En plus, les termes qui ont des multiplications des états seront considérés comme zéros car ils ne sont pas linéaires. C'est-à-dire que le système d'équations précédemment sera simplifié à :

$$\underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}}_{X(t)} \rightarrow \begin{cases} \frac{d}{dt} x_1(t) = x_2(t) \\ \frac{d}{dt} x_2(t) = \frac{-g x_3(t)}{1 + \sigma} \\ \frac{d}{dt} x_3(t) = x_4(t) \\ \frac{d}{dt} x_4(t) = \frac{u(t) - m g x_1(t)}{J_p} \end{cases}$$

Qui en notation matricielle donnera le système suivant :

$$\boxed{\underbrace{\frac{d}{dt} \begin{bmatrix} \delta x_1(t) \\ \delta x_2(t) \\ \delta x_3(t) \\ \delta x_4(t) \end{bmatrix}}_{\delta \dot{X}(t)} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{g}{1+\sigma} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{J_p} & 0 & 0 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} \delta x_1(t) \\ \delta x_2(t) \\ \delta x_3(t) \\ \delta x_4(t) \end{bmatrix}}_{\delta X(t)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_p} \end{bmatrix}}_B \delta u(t)}$$

2.7. Question 10

On pose :

$$\omega = \sqrt[4]{\frac{m \cdot g^2}{(1 + \sigma) \cdot J_p}} \quad (2.7)$$

Exercice 2.7. Calculer les valeurs propres du linéarisé tangent en boucle ouverte en fonction de ω . Le système en boucle ouverte est-il stable? Vérifier numériquement les valeurs propres en boucle ouverte avec MATLAB.

Résolution. On part de la matrice A découvert dans la Question 9 et on calcule les valeurs propres λ avec l'équation suivante :

$$\underbrace{\det(A - \lambda I) = 0}_{\text{valeurs propres définition}} \implies \begin{vmatrix} -\lambda & 1 & 0 & 0 \\ 0 & -\lambda & -\frac{g}{1+\sigma} & 0 \\ 0 & 0 & -\lambda & 1 \\ -\frac{mg}{J_p} & 0 & 0 & -\lambda \end{vmatrix} = 0 \xRightarrow{\text{Laplace Expansion}} \boxed{\lambda^4 = \frac{m \cdot g^2}{(1 + \sigma) \cdot J_p} = \omega^4}$$

Finalement les valeurs propres seront :

$$\begin{cases} \lambda_{1,2} = \pm\omega \\ \lambda_{3,4} = \pm\omega \cdot i \end{cases} \xRightarrow{\omega=6.3284} \begin{cases} \lambda_{1,2} = \pm 6.3284 \\ \lambda_{3,4} = \pm 6.3284 \cdot i \end{cases} \quad (2.8)$$

En utilisant le code suivant on calcule les valeurs propres du système :

```
1 function eigenValues = Q10(A)
2     eigenValues = eig(A);
3     for i = 1:size(eigenValues, 1)
4         if (real(eigenValues(i)) > 0)
5             disp(['warning: system diverge eig ', num2str(eigenValues(i))]);
6         end
7     end
8 end
```

Qui retourne les valeurs suivants :

```
1 >>>
2 eigenValues =
3
4     -6.3284 + 0.0000i
5     -0.0000 + 6.3284i
6     -0.0000 - 6.3284i
7     +6.3284 + 0.0000i
8
9 warning: system diverge eig 6.3284
```

On peut voir que les valeurs propres théoriques et numériques sont les mêmes.

On remarque qu'une valeur propre a la partie réelle plus grand que zéro qui fait que le système soit instable comme montrer pour les Questions 6 et 8.

3. Partie 3 - Contrôleur

3.1. Question 11

Exercice 3.1. Le linéarisé tangent est-il commandable ? Vérifier le rang de la matrice de commandabilité avec MATLAB. On pourra également s'assurer que la matrice de commandabilité est bien conditionnée.

Résolution. On note que le système sera Commandable si et seulement si la Matrice de Commandabilité $\mathcal{C}(A, B)$:

$$\mathcal{C}(A, B) = \begin{bmatrix} B & A \times B & \dots & A^{n-1} \times B \end{bmatrix} \quad (3.1)$$

est de rang $\dim(x)$, c'est-à-dire : si la matrice $\mathcal{C}(A, B)$ a une numéro de colognes linéairement indépendants égale à dimension du vecteur des états x , le système est commandable.

En utilisant le code suivant on calcule la matrice de Commandabilité :

```
1 function COM = Q11(A, B)
2     COM = B;
3     for i = 1:size(A, 1)-1
4         COM = [COM, (A^i) * B];
5     end
6
7     if rank(COM) ~= 4
8         disp(['warning: system not controable, rank ' num2str(rank(COM))]);
9     else
10        disp(['success: system controable, rank ' num2str(rank(COM))]);
11    end
12 end
```

```
1 >>>
2 success: system controable, rank 4
3
4 COM =
5          0          0          0 -272.5000
6          0          0 -272.5000          0
7          0  50.0000          0          0
8  50.0000          0          0          0
9
10 cond(COM) =
11  5.4500
```

Le système est défini avec 4 états et, comme on peut voir avec le résultat du code, le rank de la matrice de commandabilité est aussi 4, car elle a quatre 4 colognes linéairement indépendants, donc le système est commandable.

On note aussi que $\text{cond}(\text{COM}) = 5.45$ qui est proche de 1 et donc indique une matrice bien conditionnée.

Remarque. Le $\text{cond}(\text{COM})$ dépendra de la norme utilise comme donne par la définition : $\text{cond}(X, P) = \text{NORM}(X, P) * \text{NORM}(\text{INV}(X), P)$.

3.2. Question 12

On cherche tout d'abord à réaliser un bouclage d'état $\delta u(t) = -K \cdot \delta X(t)$ qui place les valeurs propres en boucle fermée en $-\omega$, -2ω et $-\omega \pm i\omega$.

Exercice 3.2. Calculer dans MATLAB le gain K qui place les valeurs propres en boucle fermée sur les valeurs voulues. On pourra utiliser la fonction `place`. Vérifier numériquement les valeurs propres en boucle fermée avec MATLAB.

Résolution. En utilisant le code suivant on boucle les valeurs propres sur les valeurs voulues :

```

1 function K = Q12(A, B, w)
2     P = [-w; -2*w; -w+1i*w; -w-1i*w];
3
4     K = place(A, B, P);
5
6     diff = sort(P) - sort(eig(A-B*K));
7     error = false;
8     for i = 1:size(diff,1)
9         if diff(i) > 1e-9
10             error = true;
11         end
12     end
13     if error
14         disp('error: fail to perform closed-loop pole assignment');
15     else
16         disp('success: closed-loop pole assignment');
17     end
18 end

```

```

1 >>>
2 success: closed-loop pole assignment
3
4 K =
5
6     -29.4300    -9.3009     8.0098     0.6328

```

Après on peut confirmer les valeurs propres sont en effet en boucle fermée avec l'équation suivante :

$$\det(A - BK - \lambda) = 0 \quad (3.2)$$

Comme on considère le système boucle fermée, c'est-à-dire un système avec feedback, donc le K n'est pas nulle, l'équation précédemment avec MATLAB on a :

```

1 >>> eig(A - B*K)
2
3 ans =
4     -12.6569 + 0.0000i
5     -6.3284 + 6.3284i
6     -6.3284 - 6.3284i
7     -6.3284 + 0.0000i

```

Comme $\omega = 6.3284$ et on a défini les valeurs propres comme $-\omega$, -2ω , $-\omega \pm i\omega$ le placement de pôles a bien marché.

3.3. Question 13

Exercice 3.3. Implémenter dans le modèle Simulink le retour d'état $\delta u(t) = -K \cdot \delta X(t)$ qui place les valeurs propres en boucle fermée sur les valeurs propres voulues.

Vérifier que le retour d'état permet bien de stabiliser la balle sur la position $r_{\text{ref}} = 0$ avec un état initial suffisamment proche de l'équilibre pour que le linéaire tangent reste une approximation valable.

Résolution. Après avoir défini K le système suivant a été mis en oeuvre sur Simulink :

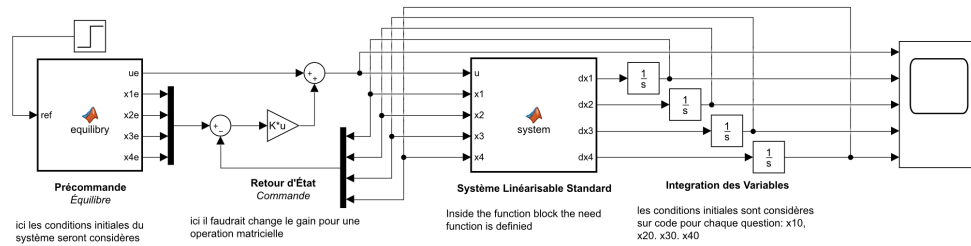
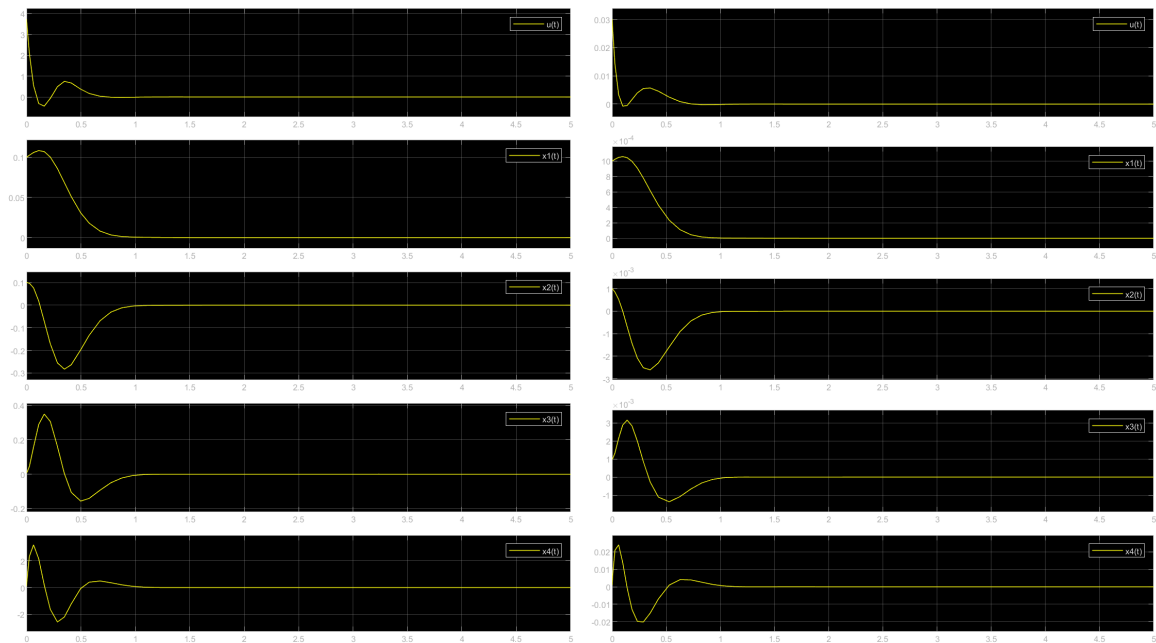


FIGURE 3.1 : Système d'Intérêt Simulink, Retour d'État

Qui donne les résultats suivants :



(a) $x_1(0) = x_2(0) = x_3(0) = x_4(0) = 0.1$

(b) $x_1(0) = x_2(0) = x_3(0) = x_4(0) = 0.001$

FIGURE 3.2 : Simulation Retour d'État, Gain Voulue

On note que, proche des valeurs d'équilibre, le système converge à $r_{\text{ref}} = 0$.

3.4. Question 14

On cherche maintenant à réaliser un bouclage d'état $\delta u(t) = -K \cdot X(t)$ qui minimise le critère :

$$J_{LQ} = \frac{1}{2} \int_0^\infty (\delta x_1^2(t) + \delta x_2^2(t) + \delta x_3^2(t) + \delta x_4^2(t) + \delta u^2(t)) dt \quad (3.3)$$

Exercice 3.4. Montrer que le critère J_{LQ} peut s'écrire sous la forme :

$$J_{LQ} = \frac{1}{2} \int_0^\infty (\delta X^T(t) R \delta X(t) + \delta u(t) Q \delta u(t)) dt \quad (3.4)$$

Identifier les matrices R et Q . Montrer que ce problème de commande optimale admet une solution. Rappeler comment est défini le gain optimal K .

Résolution. On rappelle que la matrice des états $\delta X(t)$ a des dimensions 4×1 et que la matrice d'entrée $\delta u(t)$ a des dimensions 1×1 donc le système précédemment pourrait être écrit comme :

$$\underbrace{(\delta X^T(t))}_{1 \times 4} \underbrace{R_{n \times n}}_{4 \times 4} \underbrace{\delta X(t)}_{4 \times 1} + \underbrace{\delta u(t)}_{1 \times 1} \underbrace{Q_{m \times m}}_{1 \times 1} \underbrace{\delta u(t)}_{1 \times 1}$$

On note que la seule façon d'avoir $\delta u^2(t)$ c'est si $Q_{1 \times 1} = 1$ et, par conséquence, la seule façon d'avoir $\delta x_1^2(t) + \delta x_2^2(t) + \delta x_3^2(t) + \delta x_4^2(t)$ c'est si $R_{4 \times 4} = I_{4 \times 4}$.

Ce système admet une solution si les hypothèses suivantes sont vérifiées :

1. **vrai**, (A, B) est Commandable : prove sur (3.1) Question 11 ;
2. **vrai**, R est Symétrique Positive : si elle est carré et tous ses valeurs propres sont positives ;

Preuve On note que $\det(R - \lambda I) \Rightarrow \lambda = 1$ donc R est Symétrique Positive. \square

3. **vrai**, Q est Symétrique Définie Positive : si, en plus d'être Symétrique Positive, elle est symétrique ;

Preuve On note que $Q = 1 = Q^T$ donc Q est Symétrique Définie Positive. \square

4. **vrai**, R est Symétrique Définie Positive :

Preuve On note que $R = I_{n \times n} = R^T$ donc R est Symétrique Définie Positive. \square

Remarque. Il existe une racine de R telle que (A, \sqrt{R}) est Observable, aussi valide ;

Comme tous les conditions sont attendus le problème de commande optimale propose est optimale e donc le Gain Optimale K_o sera donne pour :

$$K_o = Q^{-1} B^T S \quad (3.5)$$

Où S est l'unique solution symétrique définie positive de l'équation de Riccati algébrique suivante :

$$S A + A^T S - S B Q B^T S + R = 0 \quad (3.6)$$

3.5. Question 15

Exercice 3.5. Calculer dans MATLAB le gain optimal K qui minimise le critère J_{LQ} . On pourra utiliser la fonction `lqr`. Quels sont les valeurs propres en boucle fermée ?

Résolution. En utilisant le code suivant calcule le gain optimal qui minimise le critère J_{LQ} :

```
1 function [K, S, CLP] = Q15(A, B, Q, R)
2     [K, S, CLP] = lqr(A, B, Q, R);
3 end
```

En considérant $Q = I_{4 \times 4}$ et $R = 1$ on a :

```
1 >>>
2 K =
3
4     -11.8563     -4.0213     6.9244     1.1300
5
6
7 eig(A - B*K) =
8
9     -50.0029 + 0.0000i
10     -1.7369 + 2.7820i
11     -1.7369 - 2.7820i
12     -3.0249 + 0.0000i
```

Remarque. La nomenclature utilise pour la fonction sur MATLAB est différent de la nomenclature utilise sur le système.

3.6. Question 16

Exercice 3.6. Implémenter dans le modèle Simulink le retour d'état $\delta u(t) = -K \cdot \delta X(t)$ qui minimise le critère J_{LQ} . Vérifier que le retour d'état permet bien de stabiliser la Balle sur la position $r_{\text{ref}} = 0$ avec un état initial suffisamment proche de l'équilibre pour que le linéarisé tangent reste une approximation valable.

Résolution. Après avoir défini K_o le gain optimale, le système suivant a été mis en oeuvre sur Simulink :

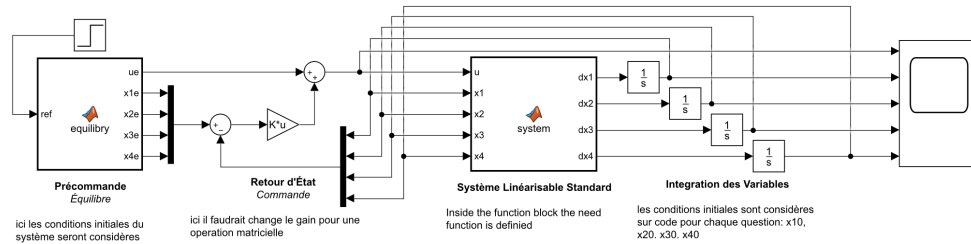
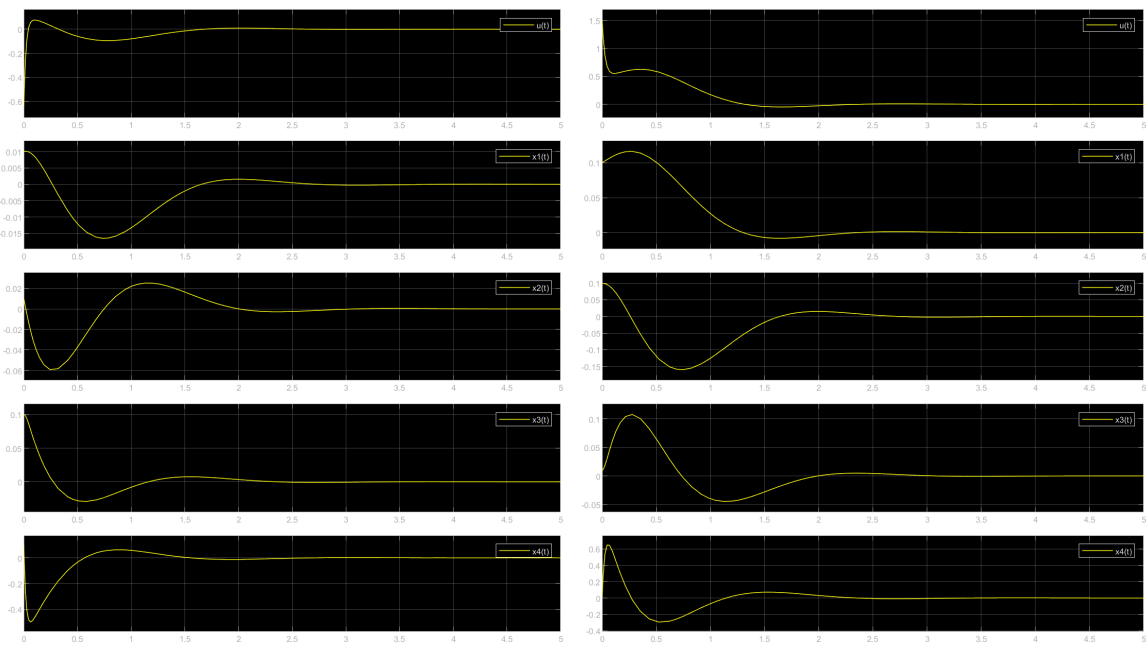


FIGURE 3.3 : Système d'Intérêt Simulink, Retour d'État

Qui donne les résultats suivants :



(a) $x_1(0) = x_2(0) = 0.01$ et $x_3(0) = x_4(0) = 0.1$

(b) $x_1(0) = x_2(0) = 0.1$ et $x_3(0) = x_4(0) = 0.01$

FIGURE 3.4 : Simulation Retour d'État, Gain Optimale

3.7. Question 17

Exercice 3.7. Comparer les résultats obtenus avec les deux méthodes.

Résolution. Après avoir défini K_ω , le gain voulue, K_o le gain optimale, le système suivant a été mis en oeuvre sur Simulink :

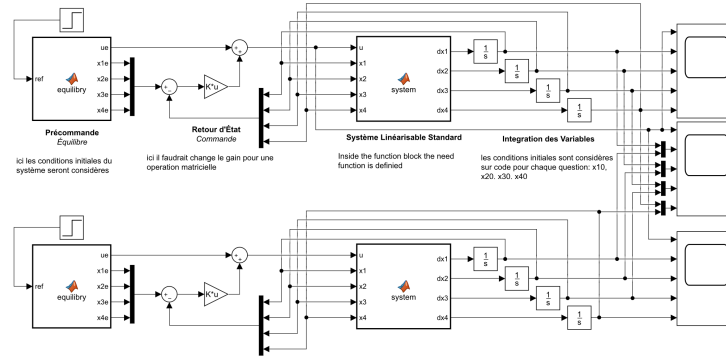
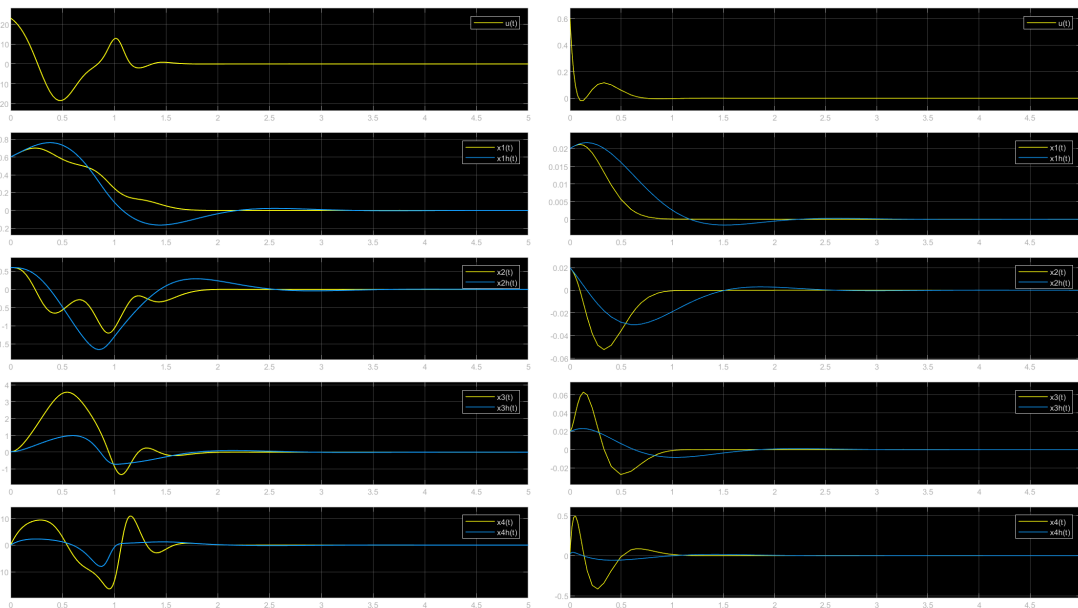


FIGURE 3.5 : Système d'Intérêt Simulink, Retour d'État

Qui donne les résultats suivants :



(a) $x_1(0) = x_2(0) = 0.6$ et $x_3(0) = x_4(0) = 0.02$

(b) $x_1(0) = x_2(0) = x_3(0) = x_4(0) = 0.02$

FIGURE 3.6 : Simulation Retour d'État, Comparaison Gain

On peut voir que le Gain Optimale K_o donne une curve plus stable, moins d'oscillations pour stabiliser, même que son amplitude soit plus importante pour les états $r(t)$ et $\theta(t)$ et moins importante pour les états $\dot{r}(t)$ et $\dot{\theta}(t)$.

En tout cas, le méthode **lqr** permette de découvrir les poles plus facilement que le méthode de placement de pôle car le premier utilise d'une critère plus facile à comprendre, bénéficier ou pénaliser la variation des paramètres dans le système.

4. Partie 4 - Observateur

On ne dispose que de capteurs capables de donner les mesures de position $r(t)$ et $\theta(t)$. On va donc devoir construire un observateur asymptotique pour estimer l'intégralité de l'état. On pose :

$$Y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} r(t) \\ \theta(t) \end{bmatrix} \quad (4.1)$$

On s'intéresse toujours à l'équilibre associé à r_{ref} , Balle stabilisée sur l'axe de rotation du Plateau.

4.1. Question 18

Exercice 4.1. Montrer que le linéarisé tangent du système, avec mesure, autour de la position d'équilibre $r_{\text{ref}} = 0$ s'écrit :

$$\left\{ \begin{array}{l} \frac{d}{dt} \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}}_{X(t)} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{g}{1+\sigma} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{J_p} & 0 & 0 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}}_{X(t)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_p} \end{bmatrix}}_B \delta u(t) \\ \\ \underbrace{\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}}_{Y(t)} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_C \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}}_{X(t)} \end{array} \right. \quad (4.2)$$

Résolution. On considère la résolution faite sur (2.6) Question 9 qui donne la première équation du système linéarisé. Après on peut voir que avec la définition des états présenter sur (2.1) on a l'équation suivante :

$$\underbrace{\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}}_{Y(t)} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_C \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}}_{X(t)} \quad (4.3)$$

4.2. Question 19

Exercice 4.2. Le linéarisé tangent est-il observable ? Vérifier le rang de la matrice d'observabilité avec MATLAB. On pourra également s'assurer que la matrice d'observabilité est bien conditionnée.

Résolution. On note que le système sera Observable si et seulement si la Matrice de Observabilité $\mathcal{O}(A, C)$:

$$\mathcal{O}(A, C) = \begin{bmatrix} C \\ C \times A \\ \vdots \\ C \times A^{n-1} \end{bmatrix} \quad (4.4)$$

est de rang $\dim(x)$, c'est-à-dire : si la matrice $\mathcal{O}(A, C)$ a une numéro de colognes linéairement indépendants égale à dimension du vecteur des états x , le système est observable.

En utilisant le code suivant on calcule la matrice de Observabilité :

```
1 function OBS = Q19(A, C)
2     OBS = [];
3     for i = 0:size(A, 1)-1
4         OBS = [OBS; (C * (A^i))];
5     end
6
7     if rank(OBS) ~= 4
8         disp(['warning: system not observable, rank ' num2str(rank(OBS))]);
9     else
10        disp(['success: system observable, rank ' num2str(rank(OBS))]);
11    end
12 end
```

```
1 >>>
2 success: system observable, rank 4
3
4 OBS =
5
6     1.0000         0         0         0
7         0         0     1.0000         0
8         0     1.0000         0         0
9         0         0         0     1.0000
10        0         0    -5.4500         0
11   -294.3000         0         0         0
12         0         0         0    -5.4500
13        0   -294.3000         0         0
```

Le système est défini avec 4 états et, comme on peut voir avec le résultat du code, le rank de la matrice de commandabilité est aussi 4, car elle a quatre 4 colognes linéairement indépendants, donc le système est commandable.

4.3. Question 20

Exercice 4.3. Écrire les équations de l'observateur asymptotique permettant d'estimer l'état $X(t)$ à partir de la mesure $Y(t)$ et la commande $u(t)$. On notera $\hat{X}(t)$ l'état estimé de L le gain de l'observateur.

Résolution. On considère l'équation suivante :

$$\dot{\hat{X}}(t) = A \cdot \hat{X}(t) + B \cdot u(t) + L \cdot (Y(t) - C \cdot \hat{X}(t)) \quad (4.5)$$

qui peut être implémenté sur MATLAB avec le diagramme que suit :

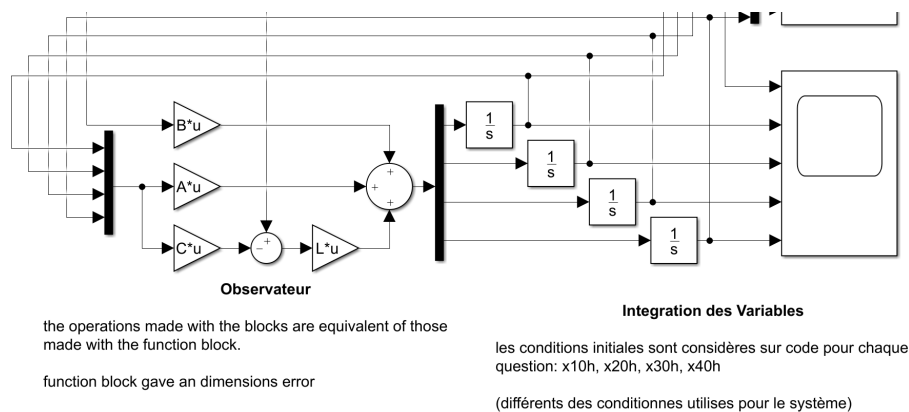


FIGURE 4.1 : Système D'Intérêt Simulink, Observateur

On note que l'Observateur est représenté avec des blocs basiques du Simulink. Une représentation avec un bloc `function` sera une alternative valable mais l'équation serait différente.

4.4. Question 21

On cherche à placer les valeurs de l'observateur en $-\omega$, -3ω et $-2\omega \pm i\omega$.

Exercice 4.4. Calculer dans MATLAB le gain L qui place les valeurs propres en boucle fermée sur les valeurs propres voulues. On pourra utiliser la fonction `place`. Vérifier numériquement les valeurs propres de l'observateur avec MATLAB.

Résolution. En utilisant le code suivant on boucle les valeurs propres sur les valeurs voulues :

```

1 function L = Q21(A, C, w)
2     P = [-w; -3*w; -2*w+1i*w; -2*w-1i*w];
3
4     L = place(A', C', P)';
5
6     diff = sort(P) - sort(eig(A-L*C));
7     error = false;
8     for i = 1:size(P,1)
9         if diff(i) > 1e-9
10             error = true;
11         end
12     end
13     if error
14         disp('error: fail to perform closed-loop pole assignment');
15     else
16         disp('success: closed-loop pole assignment');
17     end
18 end

```

```

1 >>>
2 success: closed-loop pole assignment
3
4 L =
5
6     23.9504     0.5651
7    129.2383    22.7820
8     -3.5660    26.6772
9    -272.1502   190.9987

```

Après on peut confirmer les valeurs propres sont en effet en boucle fermée avec l'équation suivante :

$$\det(A - LC - \lambda) = 0 \quad (4.6)$$

Comme on considère le système boucle fermée, c'est-à-dire un système avec feedback, donc le K n'est pas nul, l'équation précédemment avec MATLAB on a :

```

1 >>> eig(A - L*C)
2
3 ans =
4    -18.9853 + 0.0000i
5     -6.3284 + 0.0000i
6    -12.6569 + 6.3284i
7    -12.6569 - 6.3284i

```

Comme $\omega = 6.3284$ et on a défini les valeurs propres comme $-\omega$, -3ω , $-2\omega \pm i\omega$ le placement de pôles a bien marché.

4.5. Question 22

Exercice 4.5. Implémenter l'observateur dans le modèle Simulink. Vérifier qu'il permet bien d'estimer l'état au voisinage de la position $r_{\text{ref}} = 0$ avec un état suffisamment proche de l'équilibre pour que le linéarisé tangent reste une approximation valable.

Résolution. L'Observateur complet peut être implémenté avec le diagramme Simulink suivant :

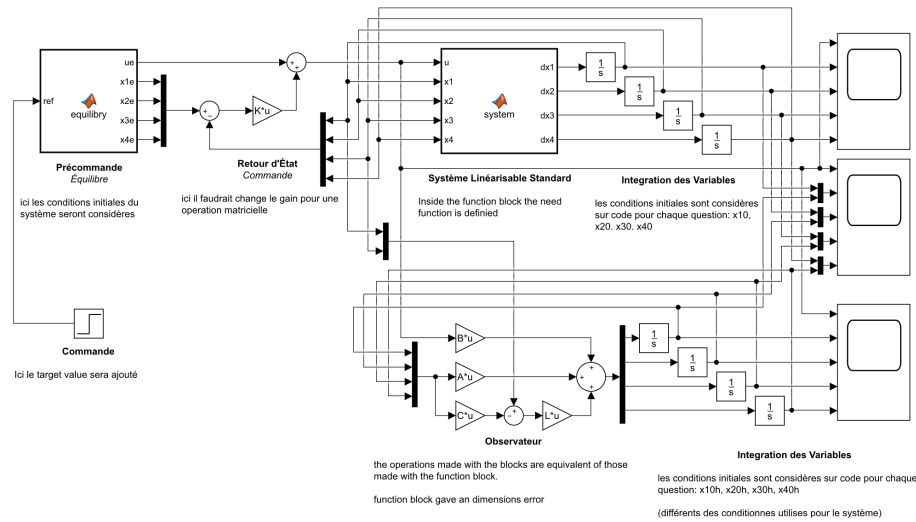
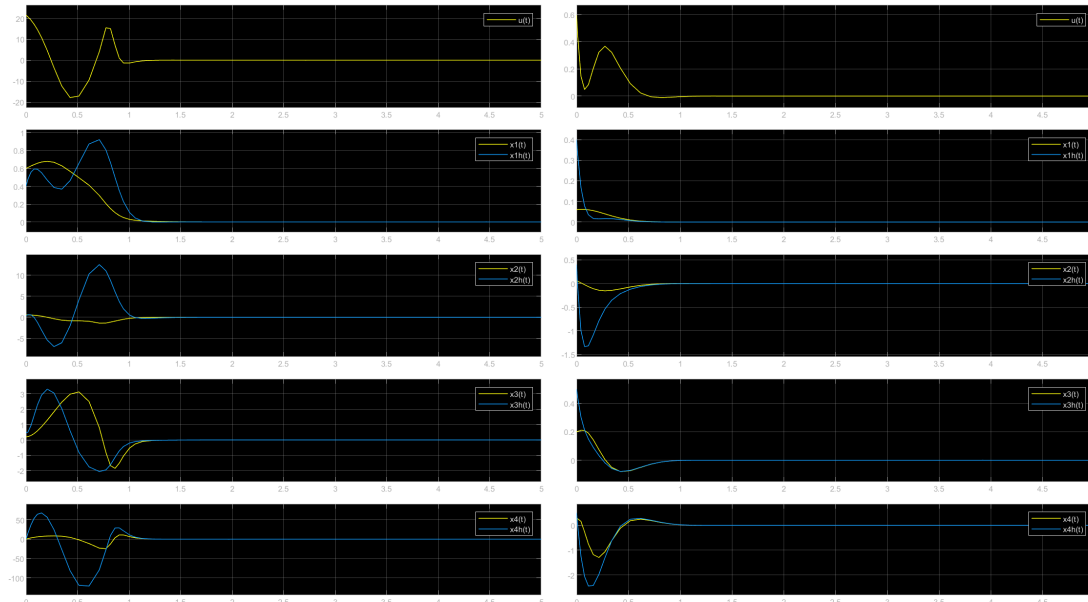


FIGURE 4.2 : Système d'Intérêt Simulink, Observateur Complet

Qui donne les résultats suivants :



(a) $x_1(0) = x_2(0) = 0.6$

(b) $x_1(0) = x_2(0) = 0.06$

FIGURE 4.3 : Simulation Observateur, autour de $r_{\text{ref}} = 0$

On considère $x_3(0) = x_4(0) = 0.2$, $\hat{x}_1(0) = \hat{x}_2(0) = 0.4$ et $\hat{x}_3(0) = \hat{x}_4(0) = 0.5$.

5. Partie 5 - Observateur - Contrôleur

On va maintenant utiliser l'état estimé par l'observateur dans le retour d'état. La loi de commande est donc de la forme :

$$u(t) = u_{\text{ref}} - K \underbrace{(\hat{X}(t) - X_{\text{ref}})}_{\delta \hat{X}(t)} \quad (5.1)$$

On pourra choisir le gain K calculé en plaçant les valeurs propres en boucle fermée ou en minimisant le critère J_{LQ} .

5.1. Question 23

Exercice 5.1. Implémenter l'observateur-contrôleur dans le modèle Simulink. Vérifier qu'il permet bien d'estimer l'état et de stabiliser la Balle sur la position $r_{\text{ref}} = 0$ avec un état initial suffisamment proche de l'équilibre pour que le linéarisé tangent reste une approximation valable.

Résolution. L'Observateur Contrôleur complet peut être implémenté avec le diagramme Simulink suivant :

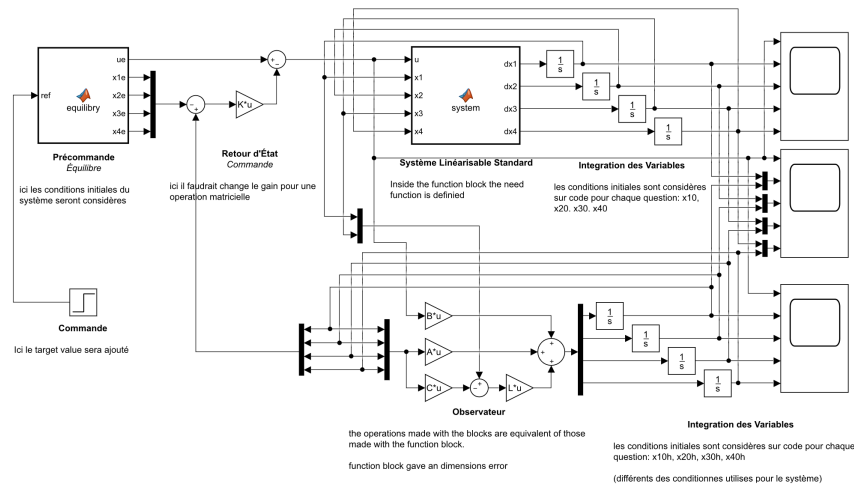
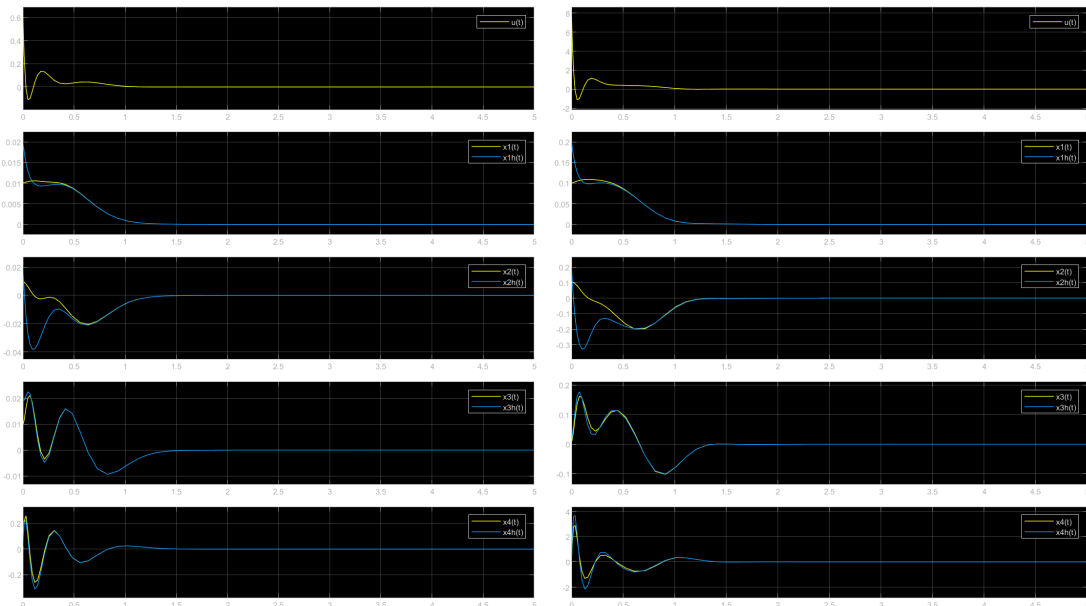


FIGURE 5.1 : Système d'intérêt Simulink, Observateur Contrôleur

Qui donne les résultats suivants :



(a) $x_1(0) = x_2(0) = 0.01$ et $\hat{x}_1(0) = \hat{x}_2(0) = 0.02$ (b) $x_1(0) = x_2(0) = 0.1$ et $\hat{x}_1(0) = \hat{x}_2(0) = 0.2$

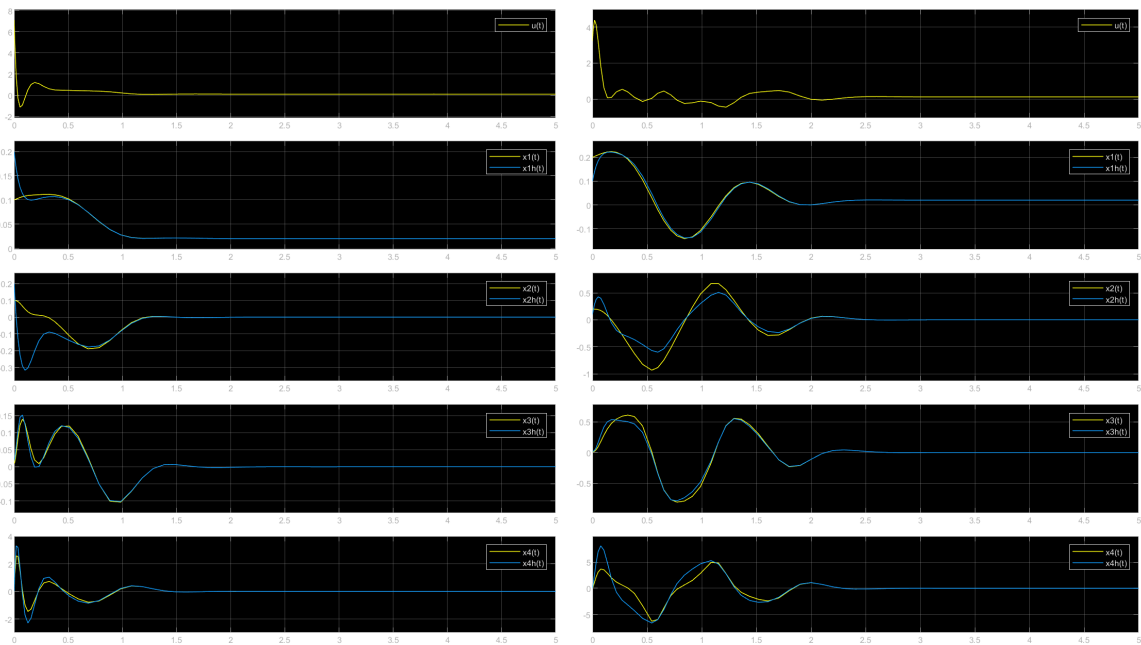
FIGURE 5.2 : Simulation Observateur-Contrôleur, autour de $r_{\text{ref}} = 0$

On considère $x_3(0) = x_4(0) = 0.01$ et $\hat{x}_3(0) = \hat{x}_4(0) = 0.02$, une configuration où est possible stabiliser la Balle sur la position désirée.

5.2. Question 24

On souhaite maintenant stabiliser la Balle sur une position $r_{\text{ref}} \neq 0$ mais supposée suffisamment proche de 0 pour que le linéarisé tangent reste une approximation valable.

Exercice 5.2. Compléter l'observateur-contrôleur dans le modèle Simulink afin de pouvoir estimer l'état et stabiliser la Balle sur n'importe quelle position $r_{\text{ref}} \neq 0$ supposée proche de 0. Vérifier qu'il permet bien d'estimer l'état et de stabiliser la Balle sur la position $r_{\text{ref}} \neq 0$ avec un état initial suffisamment proche de l'équilibre pour que le linéarisé tangent reste une approximation valable.



(a) $x_1(0) = x_2(0) = 0.1$ et $\hat{x}_1(0) = \hat{x}_2(0) = 0.2$

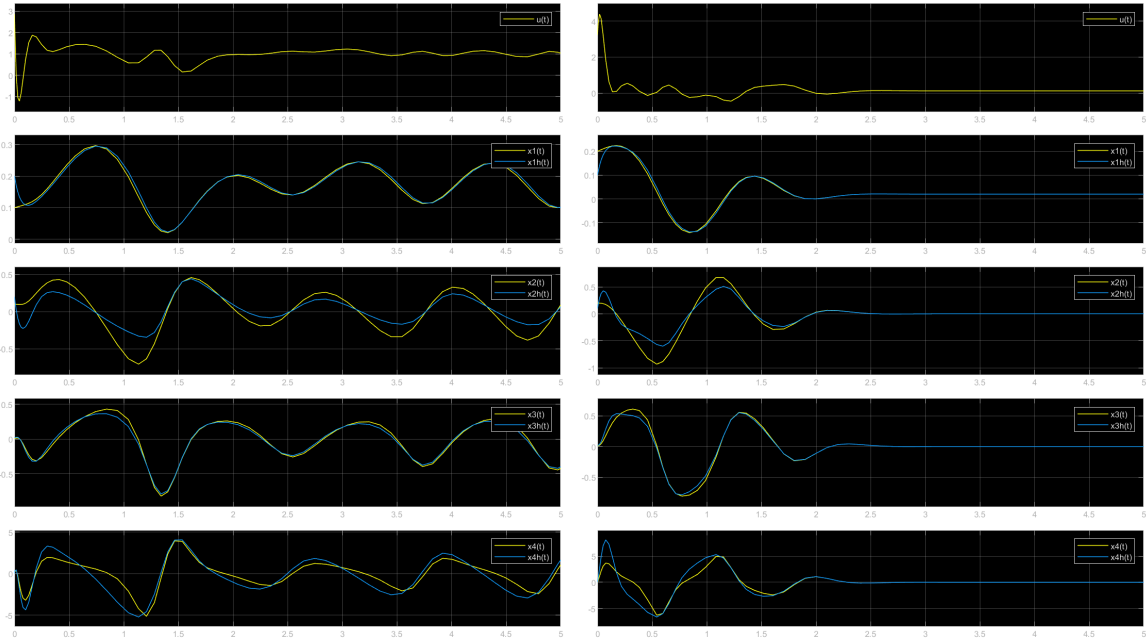
(b) $x_1(0) = x_2(0) = 0.2$ et $\hat{x}_1(0) = \hat{x}_2(0) = 0.1$

FIGURE 5.3 : Simulation Observateur-Contrôleur, autour de $r_{\text{ref}} = 0.02$

Résolution. On considère $x_3(0) = x_4(0) = 0.01$ et $\hat{x}_3(0) = \hat{x}_4(0) = 0.02$, une configuration où est possible stabiliser la Balle sur la position désirée donc la linéarisation est valable.

5.3. Question 25

Exercice 5.3. L'observateur-contrôleur est-il toujours performant lorsque la Balle s'éloigne significativement de la position $r_{\text{ref}} = 0$?



(a) $x_1(0) = x_2(0) = 0.1$ et $\hat{x}_1(0) = \hat{x}_2(0) = 0.2$

(b) $x_1(0) = x_2(0) = 0.2$ et $\hat{x}_1(0) = \hat{x}_2(0) = 0.1$

FIGURE 5.4 : Simulation Observateur-Contrôleur, autour de $r_{\text{ref}} = 0.2$

Résolution. On considère $x_3(0) = x_4(0) = 0.01$ et $\hat{x}_3(0) = \hat{x}_4(0) = 0.2$ qui fait que la Balle reste en oscillation autour du point d'équilibre désirée quand $x_1(0) \neq r_{\text{ref}}$ car la linéarisation n'est plus valable.

Le Observateur-Contrôleur arrive à compenser le mouvement en dehors de la région considérée mais c'est pas suffisamment, donc la Balle dépasse le point désirée et un comportement cyclique commence.

Par contre, si $x_1(0) = r_{\text{ref}}$ le Observateur-Contrôleur arrive à stabiliser le système car il est déjà sur la région d'équilibre et donc la stabilisation est possible.