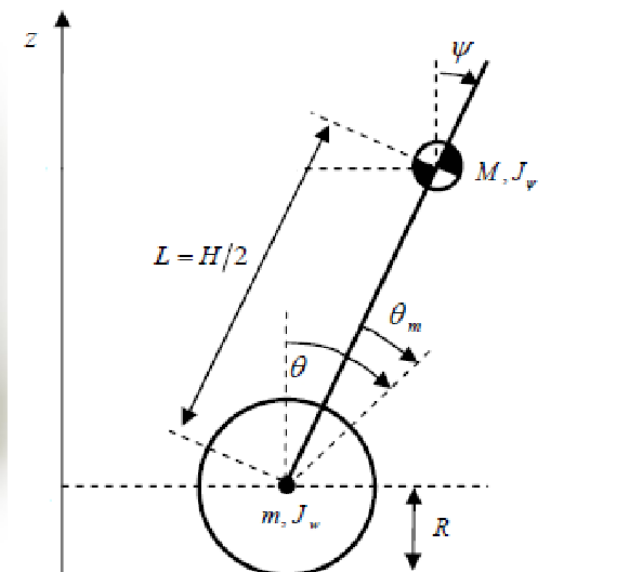


Rapport de bureau d'études Automatique Systèmes Cyber Physique

RAGOT Cyrian



Contents

1	Introduction	3
2	Modèle du pendule inversé	4
2.1	Contrôle par retour d'état	4
2.1.1	Analyse du modèle théorique	4
2.1.2	Simulation du modèle sur Simulink	5
2.2	Capteurs	6
3	Modèle du robot Lego	7
4	Robot Lego NXT	8
5	Conclusion	9

1 Introduction

2 Modèle du pendule inversé

2.1 Contrôle par retour d'état

Dans cette partie, nous étudierons un modèle simple d'un pendule inversé contrôlé par retour d'état (figure 1) pour lequel on a accès aux variables de sortie. Le système contrôlé issu des équations physiques de la dynamique est

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{g}{l} \sin(x_1(t)) - \frac{\cos(x_1(t))u(t)}{l} \\ \dot{x}_1(0) = x_{0,1} = \alpha_0 \\ \dot{x}_2(0) = x_{0,2} = \dot{\alpha}_0, \end{cases} \quad (1)$$

avec

- $\mathbf{g} = 9.81\mathbf{m/s^2}$ constante de gravité
- $l = 10\mathbf{m}$ longueur du pendule
- $\mathbf{t}_0 = 0\mathbf{s}$ instant initial
- $\mathbf{x}(t) = (x_1(t), x_2(t))^T = (\alpha(t), \dot{\alpha}(t))^T$ variable d'état
- $(\mathbf{x}_e, \mathbf{u}_e)^T = (0, 0, 0)^T$ point de fonctionnement
- $\mathbf{u}(t)$ contrôle par retour d'état

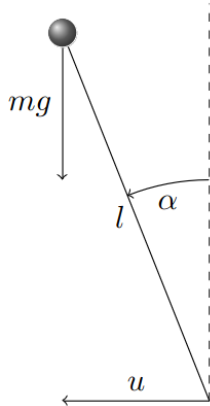


Figure 1: Schéma du pendule inversé

2.1.1 Analyse du modèle théorique

L'équation d'état est alors

$$\dot{x}(t) = f(x(t), u(t)), \quad (2)$$

où

$$f(x, u) = \begin{pmatrix} x_2 \\ \frac{g}{l} \sin(x_1) - \frac{\cos(x_1)u}{l} \end{pmatrix} \quad (3)$$

On souhaite stabiliser le système à l'origine (la position verticale du pendule inversé), cependant le système non contrôlé n'est pas stable à l'origine. En effet, lorsque $u = 0$,

en passant par la jacobienne de g au point de fonctionnement puis son polynôme caractéristique, on trouve la valeur propre $\sqrt{\frac{g}{l}}$ qui est à partie réelle strictement positive.

On choisit alors un contrôle en boucle fermée par retour d'état linéaire de la forme $u(t) = u_e + K(x(t) - x_e)$ avec $K = (k_1, k_2)$. Cherchons alors K de manière à avoir un contrôle qui stabilise le système asymptotiquement en x_e .

On boucle le système :

$$\dot{x}(t) = f(x(t), u_e + K(x(t) - x_e)) := g(x(t)) \quad (4)$$

Alors $g(x_e) = f(x_e, u_e) = 0$ donc x_e est un point d'équilibre de $\dot{x} = g(x)$.

La matrice jacobienne associée est alors

$$J_g(x) = \begin{pmatrix} 0 & 1 \\ \frac{\cos(x_1)}{l}(g - k_1) + \frac{u_e}{l}\sin(x_1) & -\frac{\cos(x_1)}{l}k_2 \end{pmatrix} \quad (5)$$

Au point de fonctionnement $(x_e, u_e) = (0, 0, 0)$ on a

$$J_g(x_e) = \begin{pmatrix} 0 & 1 \\ \frac{g-k_1}{l} & -\frac{k_2}{l} \end{pmatrix} \quad (6)$$

De plus, le contrôle stabilise asymptotiquement le système en (x_e, u_e) si et seulement si les valeurs propres sont à parties réelles strictement négatives donc si et seulement si

$$\begin{cases} \text{tr}(J_g(x_e)) < 0 \\ \det(J_g(x_e)) > 0 \end{cases} \quad (7)$$

Finalement,

$$\begin{cases} k_1 > g \\ k_2 > 0 \end{cases} \quad (8)$$

2.1.2 Simulation du modèle sur Simulink

Maintenant que l'on a compris comment le système contrôlé devrait réagir, nous allons effectuer des simulations sur Simulink où nous pourrions comparer le comportement du système pour différents cas d'étude.

Les schémas blocs construits sur Simulink lors des séances de TP sont représentés sur les figures 2, 3 et 4. Nous étudierons les simulations avec les différents cas de la figure ???. On rappelle que t_f est le temps de simulation en secondes.

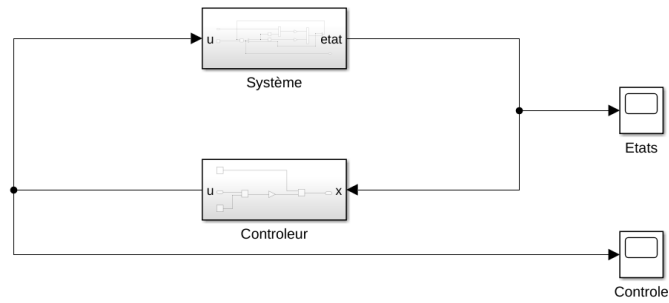


Figure 2: Schéma bloc du système contrôlé sur Simulink

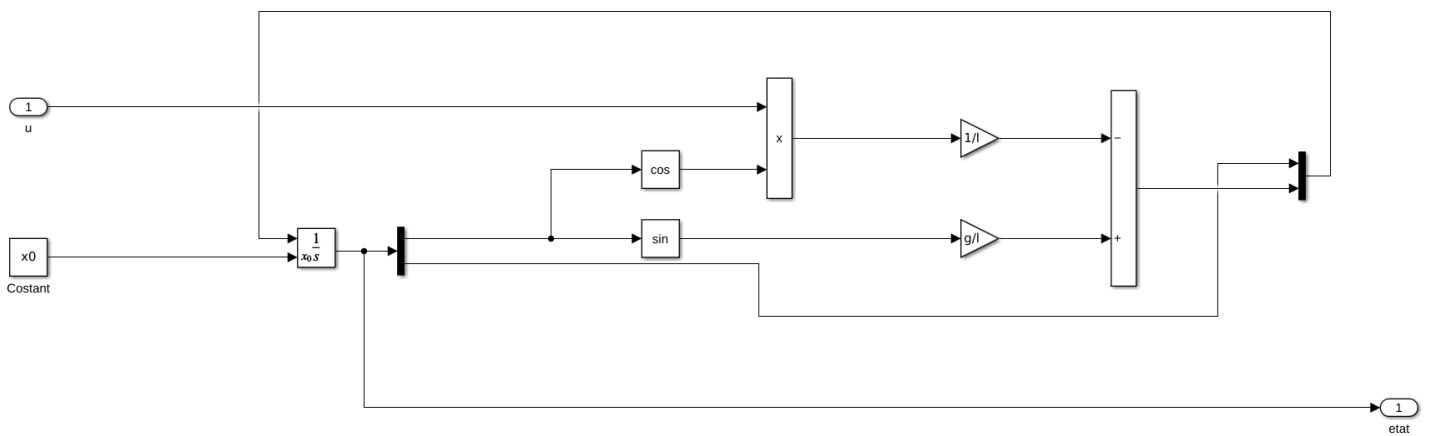


Figure 3: Schéma bloc du système

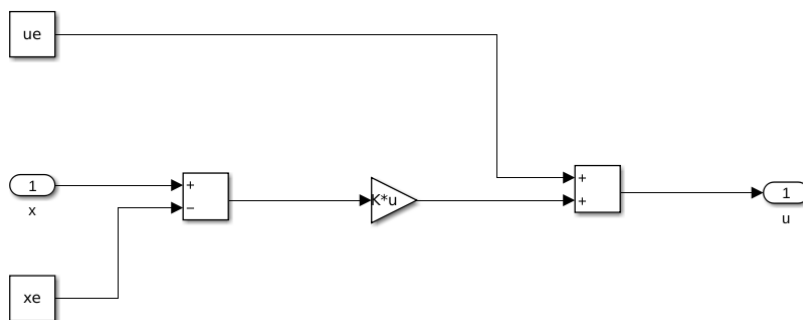


Figure 4: Schéma bloc du contrôle

Commentons maintenant les courbes obtenues.

2.2 Capteurs

Nous avons pu contrôler correctement le système à l'aide des variables de sorties de position angulaire et de vitesse, cependant, sur un système réel, on n'aura pas toujours accès à toutes les variables de sorties. On va alors maintenant supposer que l'on a accès qu'à la valeur α à l'aide d'un capteur sur le système. Il faut alors reconstruire α à l'aide d'un prédicteur. Pour cela, on utilise un intégrateur continu.

3 Modèle du robot Lego

4 Robot Lego NXT

5 Conclusion