

# Rapport BE

THEVENET Louis

MARTIN Nolann

## Table des matières

I Introduction .....	2
II Travaux pratiques .....	4
II - 1 TP2 - Modèle continu structuré du pendule inversé .....	4
II - 1.1 Présentation .....	4
II - 1.1.1 Résultats pour le contrôle par retour d'état .....	5
II - 1.2 Simulation d'un capteur et d'un prédicteur .....	6
II - 1.2.1 Résultats avec capteur et prédicteur .....	6
II - 2 TP3 - Modèle continu et discret du robot Lego .....	8
II - 2.1 Présentation .....	8
II - 2.1.1 Résultats pour le contrôle par retour d'état .....	9
II - 2.2 Introduction du capteur et du prédicteur .....	11
II - 2.2.1 Résultats avec capteur et prédicteur .....	12
II - 2.3 Modèle hybride .....	15
III Conclusion .....	16

## I Introduction

On souhaite simuler un pendule inversé, à l'aide du principe fondamental du moment cinétique, on obtient le système dynamique qui le régit :

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

où  $g = 9.81$  et  $l = 10$

On cherche maintenant à contrôler le système par retour d'état afin de le stabiliser au point d'équilibre  $x_e = (0, 0)$ .

Pour cela on va linéariser notre système, ce qui est possible autour du point d'équilibre  $x_e$  car :

$$\sin(x_1) \underset{0}{\approx} x_1$$

$$\cos(x_1) \underset{0}{\approx} 1$$

Ainsi, le système approximé linéairement autour de  $x_e$  est

$$(S') : \begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = \frac{g}{l} x_1(t) - \frac{u(t)}{l} \\ x_1(0) = x_{0,1} = \alpha_0 \\ x_2(0) = x_{0,2} = \dot{\alpha}_0 \end{cases}$$

Et se réécrit

$$(S') : \begin{cases} \dot{Ax} + Bu \\ x_0 = \begin{pmatrix} \alpha_0 \\ \dot{\alpha}_0 \end{pmatrix} \end{cases}$$

avec

$$A = \begin{pmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{pmatrix}, B = \begin{pmatrix} 0 \\ -\frac{1}{l} \end{pmatrix}$$

On peut alors calculer les valeurs propres de  $A$ .

$$\begin{cases} \text{tr}(A) = 0 \\ \det(A) = -\frac{g}{l} \end{cases} \Rightarrow \text{Sp}(A) = \left\{ -\sqrt{\frac{g}{l}}, \sqrt{\frac{g}{l}} \right\}$$

Ainsi, pour  $K = (k_1, k_2) \in \mathbb{R}^2$ ,

$$A - BK = \begin{pmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{pmatrix} - \begin{pmatrix} 0 \\ -\frac{1}{l} \end{pmatrix} (k_1 \ k_2) = \begin{pmatrix} 0 & 1 \\ \frac{g}{l} - k_1 & -\frac{k_2}{l} \end{pmatrix}$$

Les paramètres  $k_1$  et  $k_2$  du contrôle par retour d'état sont choisis tels que les valeurs propres de  $A - BK$  soient à **parties réelles strictement négatives** afin que le système soit asymptotiquement contrôlé.

Après calcul des valeurs propres de  $A - BK$  on obtient donc la condition suivante :

$$k_1 > g \text{ et } k_2 > 0$$

Ainsi, lors de nos simulations, il suffira de choisir  $k_1$  et  $k_2$  tels que  $k_1 > g$  et  $k_2 > 0$  pour que le système soit asymptotiquement contrôlé.

## II Travaux pratiques

### II - 1 TP2 - Modèle continu structuré du pendule inversé

#### II - 1.1 Présentation

Lors de ce TP, on réalise le schéma Simulink du pendule inversé que l'on contrôle par retour d'état. Dans une seconde partie, on simule l'existence de capteurs qui nous obligent à reconstruire numériquement certaines variables d'état.=== Contrôle par retour d'état Le **contrôle par retour d'état**  $u(t) = u_e + K(x(t) - x_e)$  évalue l'écart entre  $x(t)$  et le point d'équilibre recherché  $x_e$ ,  $u_e$  représente la consigne au point d'équilibre et  $K = (k_1, k_2)$  sont les paramètres du contrôle.

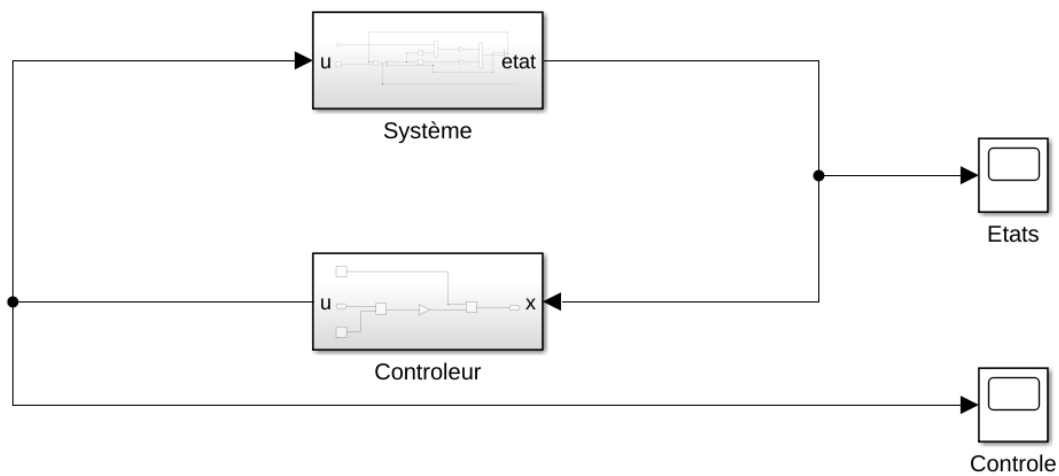


Figure 1: Schéma Simulink du système contrôlé par retour d'état

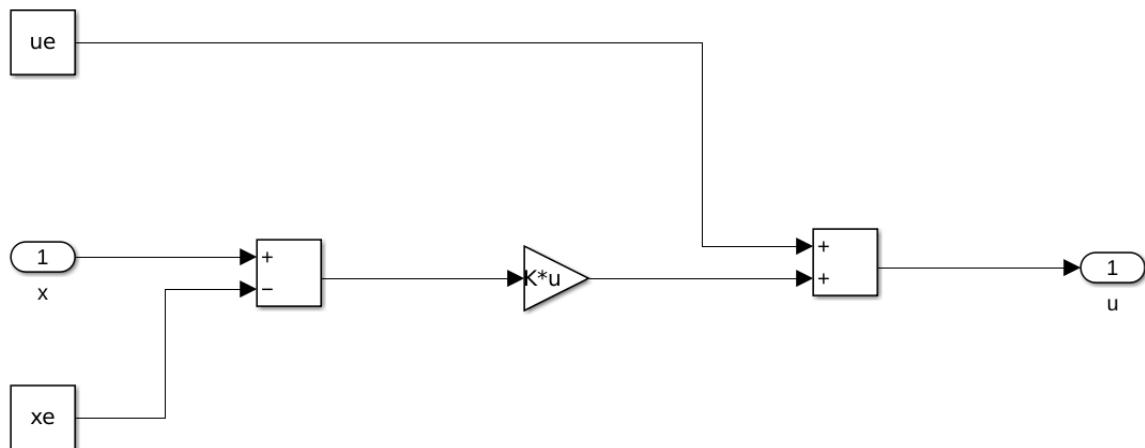


Figure 2: Schéma Simulink du contrôleur

## II - 1.1.1 Résultats pour le contrôle par retour d'état

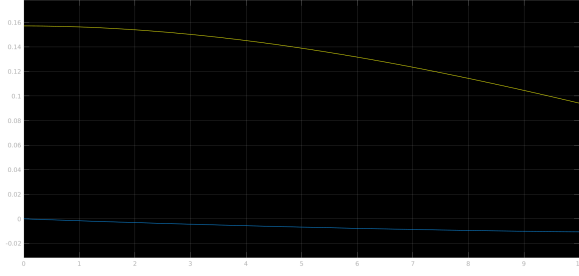


Figure 3:  $\alpha(t)$  (orange),  $u(t)$  (blue)

$x_0$	$t_f$	$K$	Intégrateur
$(\frac{\pi}{20}, 0)$	10	(30, 10)	ode45

Sur cette première image, on constate que le temps de simulation est trop court pour que le système converge.

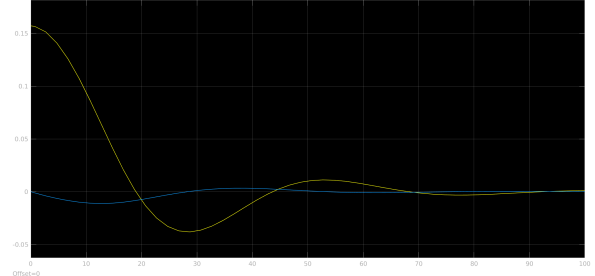


Figure 4:  $\alpha(t)$  (orange),  $u(t)$  (blue)

$x_0$	$t_f$	$K$	Intégrateur
$(\frac{\pi}{20}, 0)$	100	(10, 1)	ode45

Avec un temps de simulation plus long, et des valeurs de  $k_1$  et  $k_2$  inférieures, le système converge vers le point d'équilibre  $x_e = (0, 0)$ .

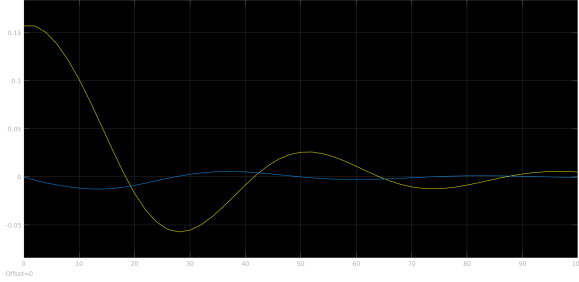


Figure 5:  $\alpha(t)$  (orange),  $u(t)$  (blue)

$x_0$	$t_f$	$K$	Intégrateur
$(\frac{\pi}{20}, 0)$	100	(10, 1)	Euler, ode1

L'intégrateur **ode1** (Euler) trouve une solution proche de celle de l'intégrateur **ode45** pour les mêmes  $x_0$  et  $K$  (voir Figure 4). Cependant, on constate que l'intégrateur **ode1** est moins précis que **ode45** et il faut un temps de simulation plus long pour qu'il soit asymptotiquement contrôlé.

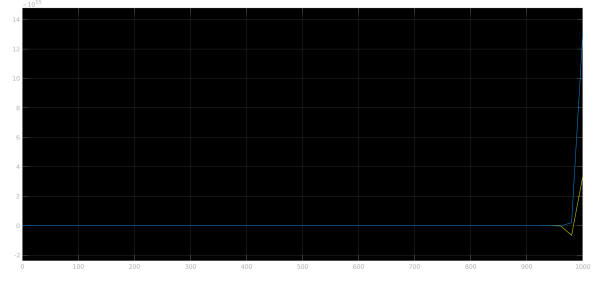


Figure 6:  $\alpha(t)$  (orange),  $u(t)$  (blue)

$x_0$	$t_f$	$K$	Intégrateur
$(\frac{\pi}{20}, 0)$	1000	(10, 1)	Euler, ode1

Pour un temps de simulation plus long  $t_f = 1000$ , on constate que l'intégrateur **ode1** produit une erreur numérique en fin de simulation, le système diverge.

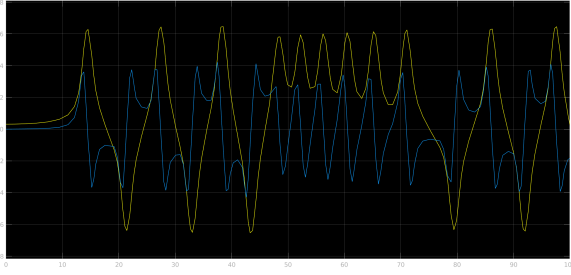


Figure 7: ■ :  $\alpha(t)$ , ■ :  $u(t)$

$x_0$	$t_f$	$K$	Intégrateur
$(\frac{\pi}{10}, 0)$	100	(10, 1)	ode45

Pour cette condition initiale, on constate une nouvelle fois une erreur numérique dans la solution, le pendule peut tourner sur lui-même dans ce cas et le système n'est jamais contrôlé asymptotiquement.

Ces courbes confirment les résultats théoriques obtenus précédemment. En effet, on constate que lorsque  $k_1 > g$  et  $k_2 > 0$ , le système est asymptotiquement contrôlé. On a aussi pu explorer les limites du modèle continu structuré en rencontrant quelques cas d'erreurs numériques.

## II - 1.2 Simulation d'un capteur et d'un prédicteur

Ici on suppose que l'on a accès qu'à  $\dot{\alpha}(t)$  et on reconstruit  $\alpha(t)$  grâce à des sous-systèmes capteur et prédicteur

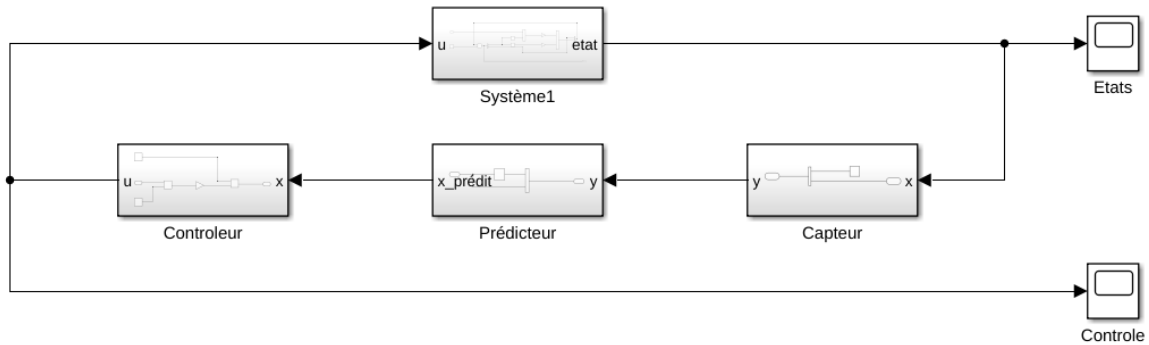


Figure 9: Schéma Simulink du système contrôlé par retour d'état avec capteur et prédicteur

Le bloc Capteur ne garde que la composante  $\dot{\alpha}(t)$  de  $x(t)$  et le bloc Prédicteur reconstruit  $\alpha(t)$  à partir de  $\dot{\alpha}(t)$ . On peut alors utiliser le même contrôle par retour d'état que précédemment.

### II - 1.2.1 Résultats avec capteur et prédicteur

■ :  $\alpha(t)$ , ■ :  $u(t)$

$x_0$	$t_f$	$K$	Pas	Intégrateur
$(\frac{\pi}{20}, 0)$	100	(10, 1)	Pas	Euler, ode1

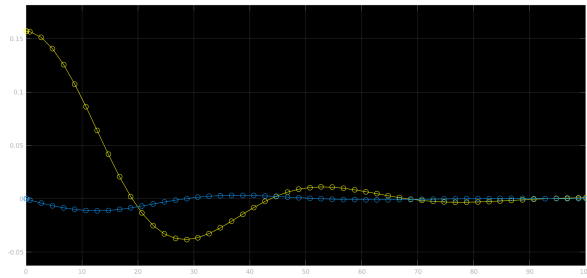


Figure 10: Pas = par défaut

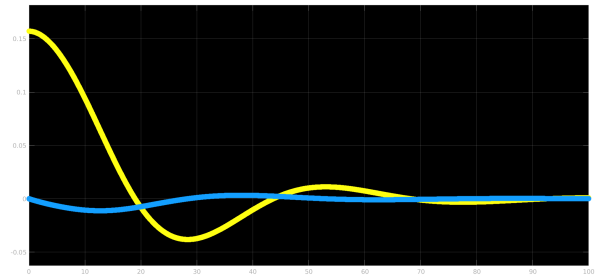


Figure 11: Pas = 0.001

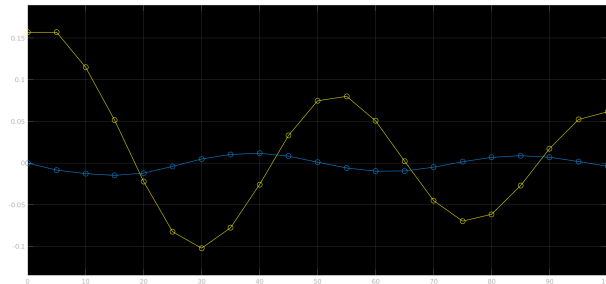


Figure 12: Pas = 5

Dans la Figure 10, on retrouve une solution semblable à celle obtenue à sur Figure 4 avec un nombre de points limité.

Dans la Figure 11, le pas est très faible, l'intégrateur `ode1` (Euler) renvoie une solution visuellement proche de la solution continue

Dans la Figure 12, le nombre de points qui forment la solution est très faible. On perd en précision mais on gagne en temps de calcul. L'allure de la solution est toujours reconnaissable.

Ces résultats permettent d'illustrer l'importance du pas de discrétisation dans la solution calculée.

Le cas Figure 10 semble le plus pertinent pour une simulation numérique, il permet d'avoir une solution précise et rapide à calculer.

## II - 2 TP3 - Modèle continu et discret du robot Lego

### II - 2.1 Présentation

On reprend le modèle continu du robot Lego pendule inversé avec un contrôleur par retour d'état.

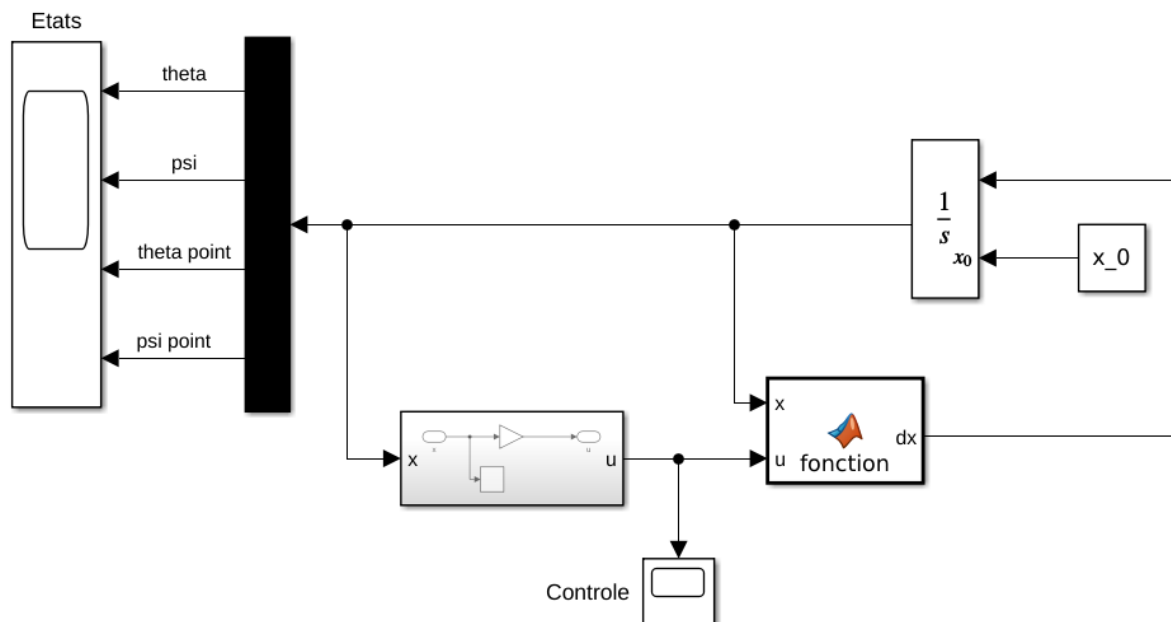


Figure 13: Schéma Simulink du système contrôlé par retour d'état

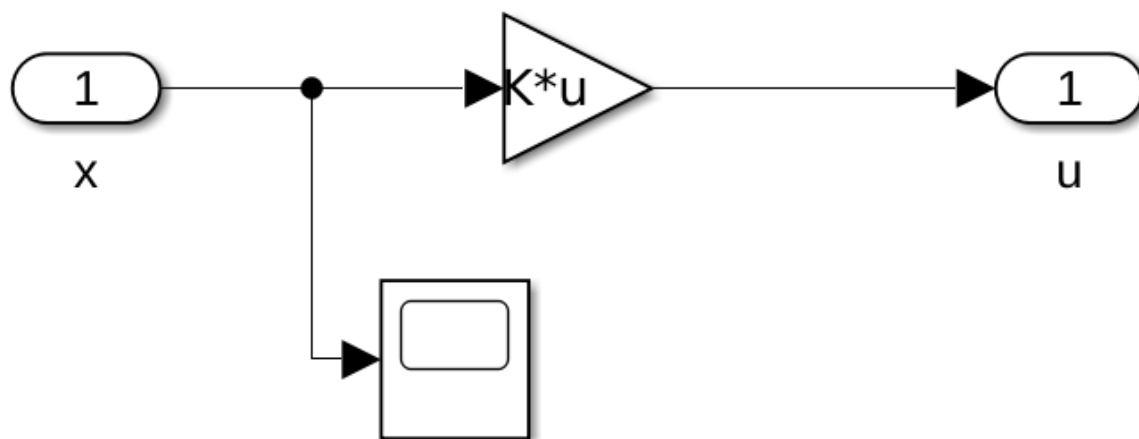


Figure 14: Schéma Simulink du contrôle par retour d'état

Le vecteur  $K$  est calculé à partir des valeurs propres souhaitées  $V$  et des matrices  $A$  et  $B$  de la manière suivante :

```
1 V = [-136.5905, -2.6555, -3.5026, -5.9946];
2 K = -place(A,B,V);
```



## II - 2.1.1 Résultats pour le contrôle par retour d'état

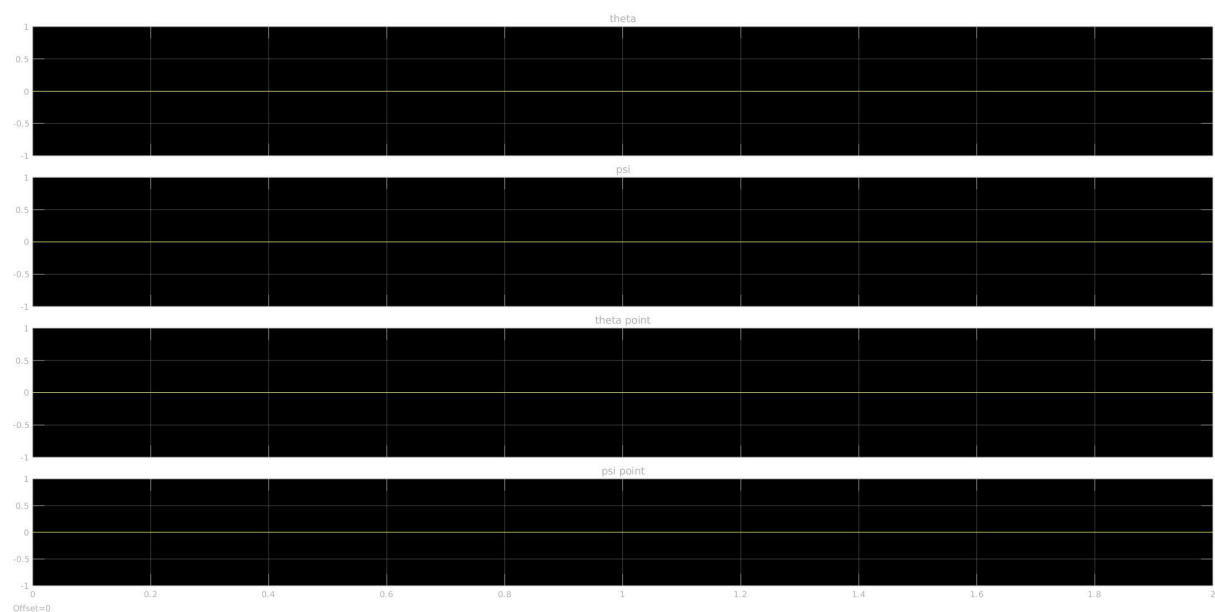


Figure 15:  $x(t)$

$x_0$	$t_f$	$K$	Intégrateur
$(0, 0, 0, 0)$	2	$K$	ode45

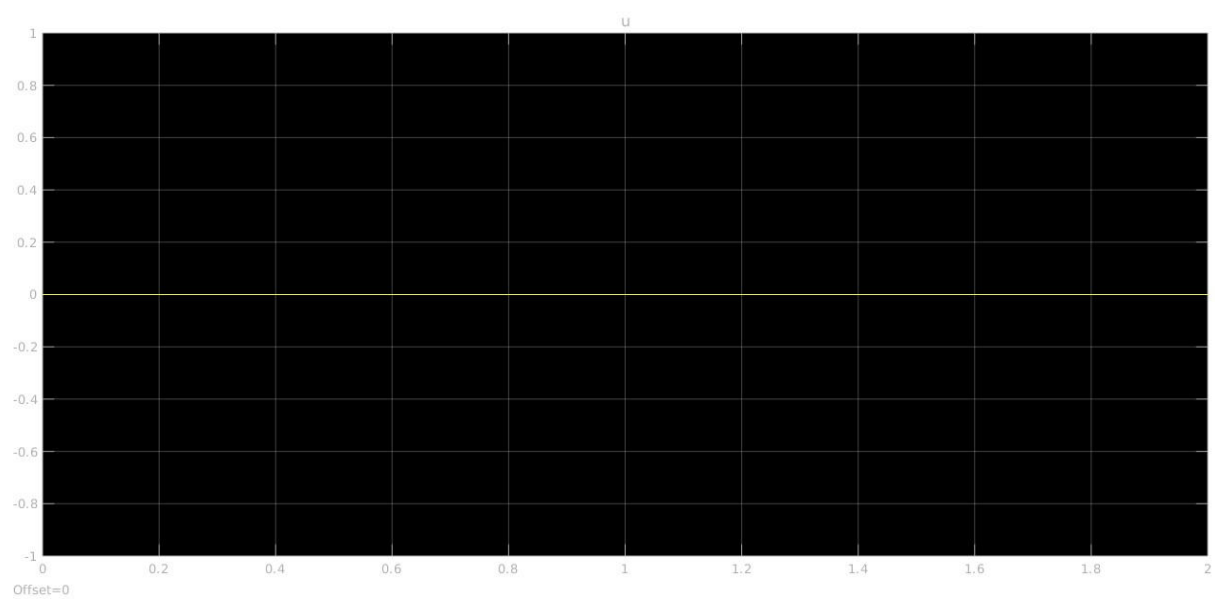


Figure 16:  $u(t)$

A l'origine, le contrôle est nul et le système est à l'équilibre.

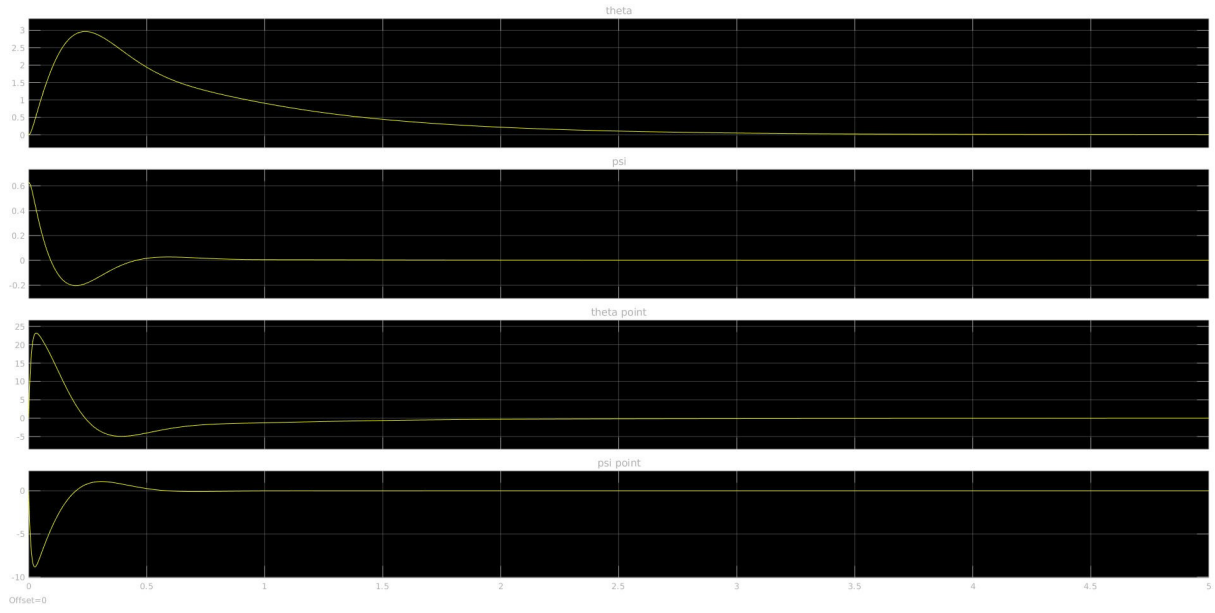


Figure 17:  $x(t)$

$x_0$	$t_f$	$K$	Intégrateur
$(0, \frac{\pi}{5}, 0, 0)$	5	$K$	ode45

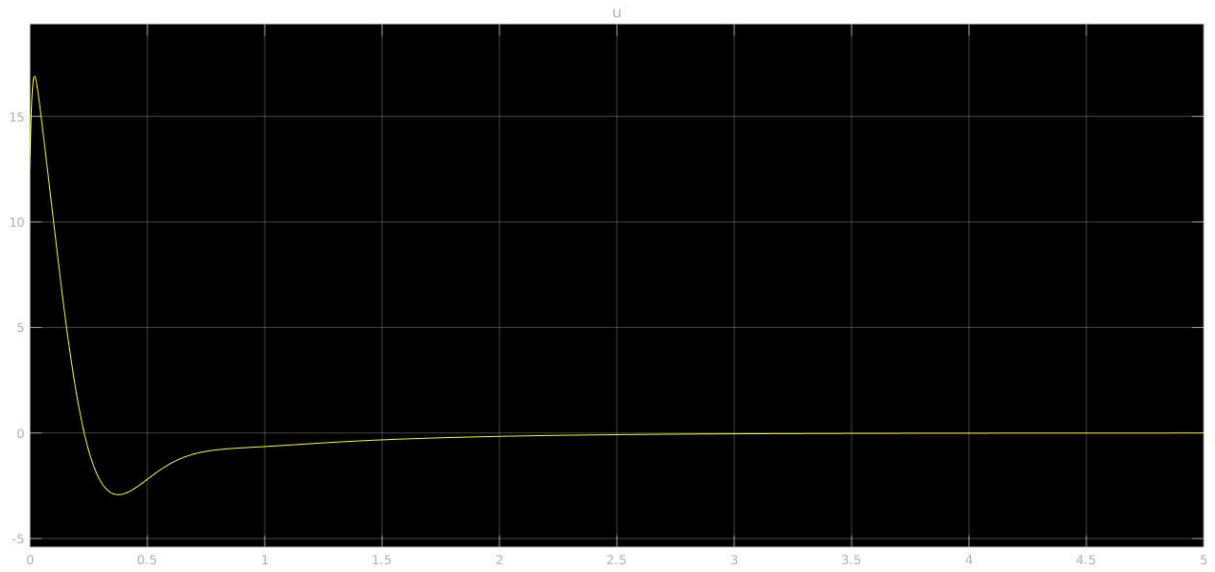


Figure 18:  $u(t)$

Lorsque  $\dot{\theta}(0) = \frac{\pi}{5}$ , le système converge rapidement vers l'équilibre.

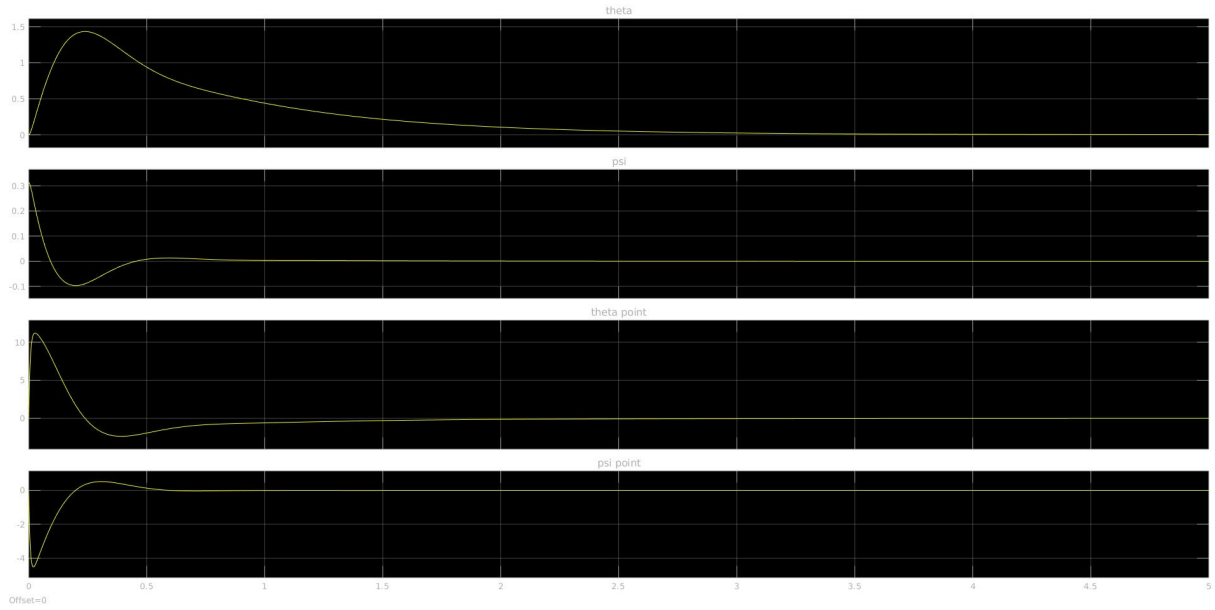


Figure 19:  $x(t)$

$x_0$	$t_f$	$K$	Intégrateur
$(0, \frac{\pi}{10}, 0, 0)$	5	$K$	ode45

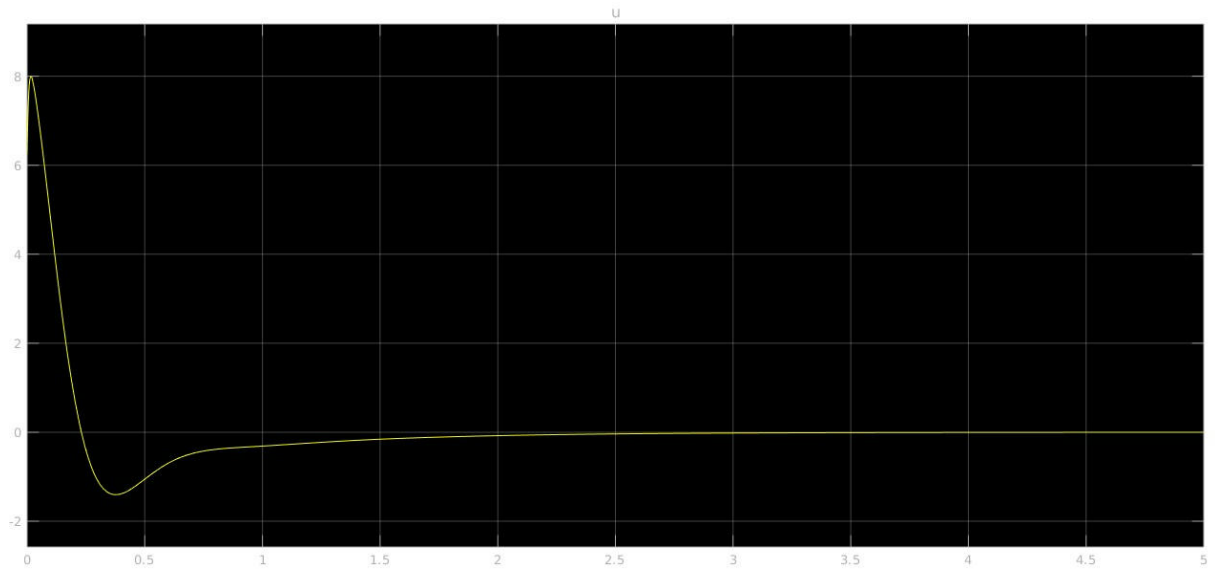


Figure 20:  $u(t)$

Lorsque  $\dot{\theta}(0) = \frac{\pi}{10}$ , l'allure de la solution reste la même et le système est toujours contrôlé asymptotiquement.

## II - 2.2 Introduction du capteur et du prédicteur

On suppose que l'on a accès qu'à  $\dot{\psi}(t)$  et  $\theta(t)$  et on reconstruit  $x(t) = (\theta(t), \dot{\theta}(t), \psi(t), \dot{\psi}(t))$  par un sous-système prédicteur.

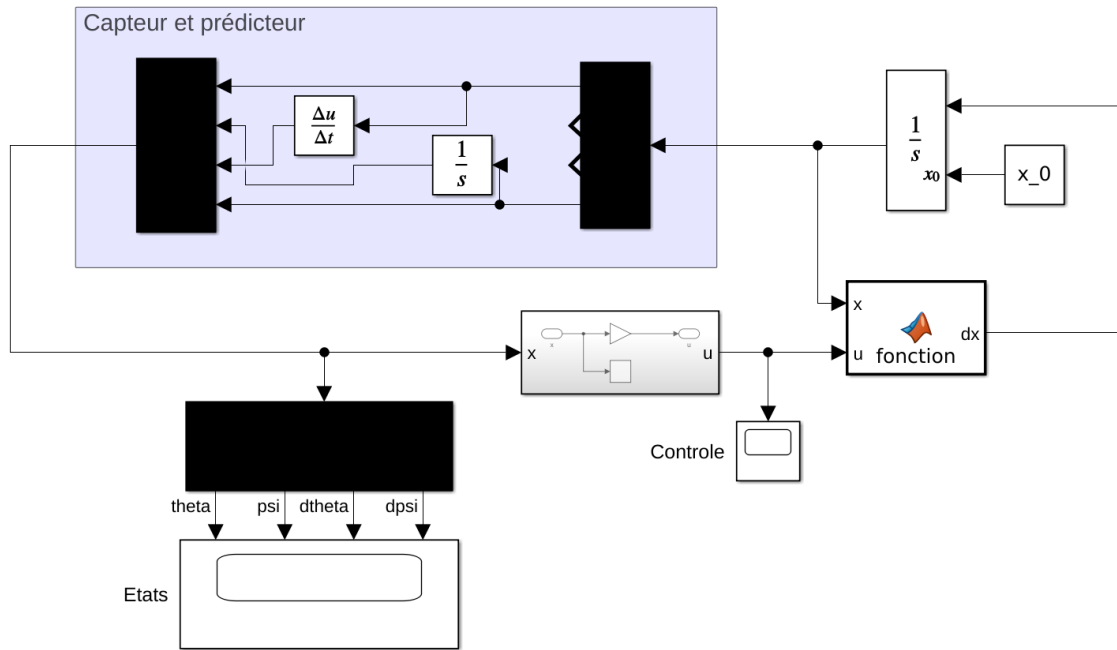


Figure 21: Schéma Simulink du système contrôlé par retour d'état avec capteur et prédicteur

Le capteur est modélisé par le bloc **Demux** qui permet de sélectionner les composantes  $\dot{\psi}(t)$  et  $\theta(t)$  de  $x(t)$ . Le bloc **Mux** permet de reconstruire  $x(t)$  avec les composantes déduites par intégration et dérivation. On peut alors utiliser le même contrôle par retour d'état que précédemment.

## II - 2.2.1 Résultats avec capteur et prédicteur

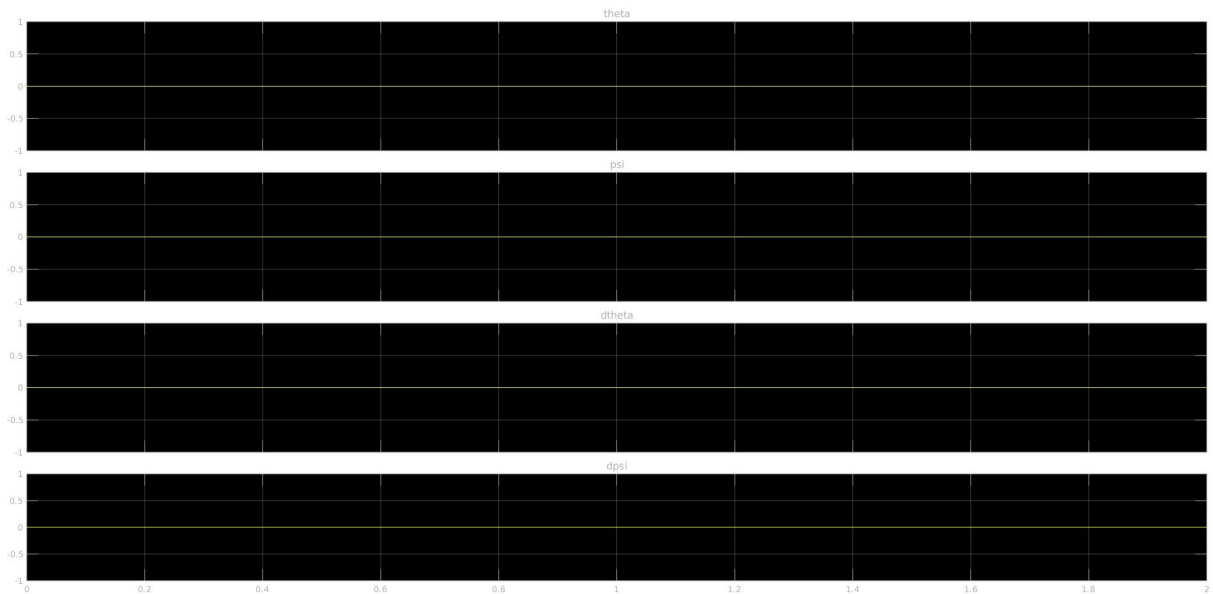


Figure 22:  $x(t)$

$x_0$	$t_f$	$K$	Intégrateur
(0, 0, 0, 0)	2	$K$	ode45

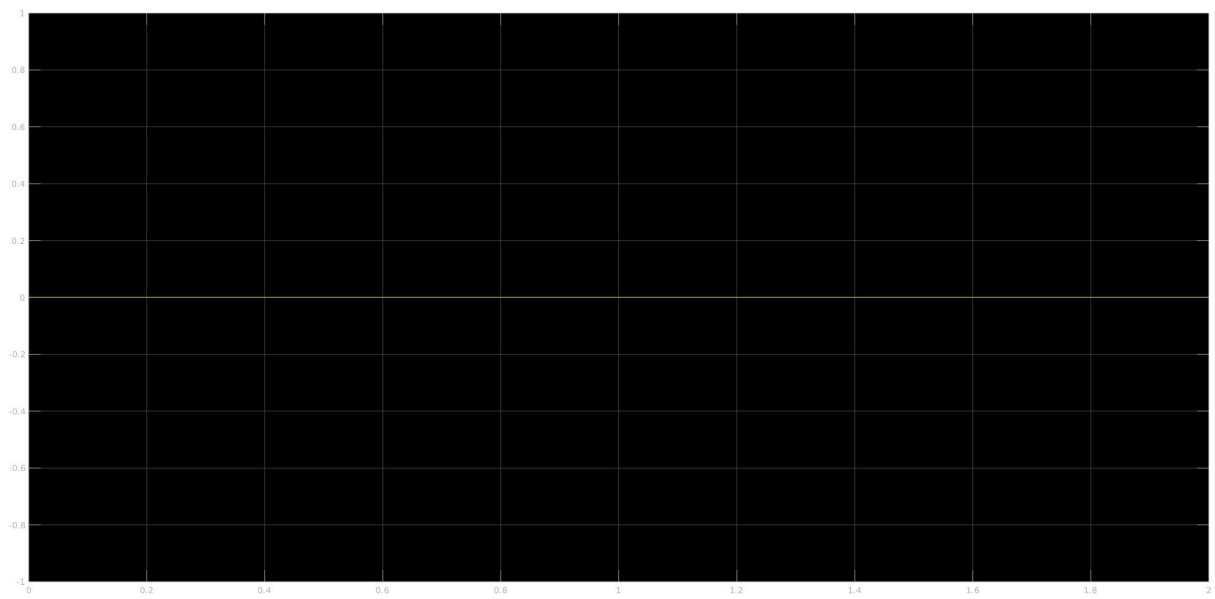


Figure 23:  $u(t)$

L'origine est un point d'équilibre du système.

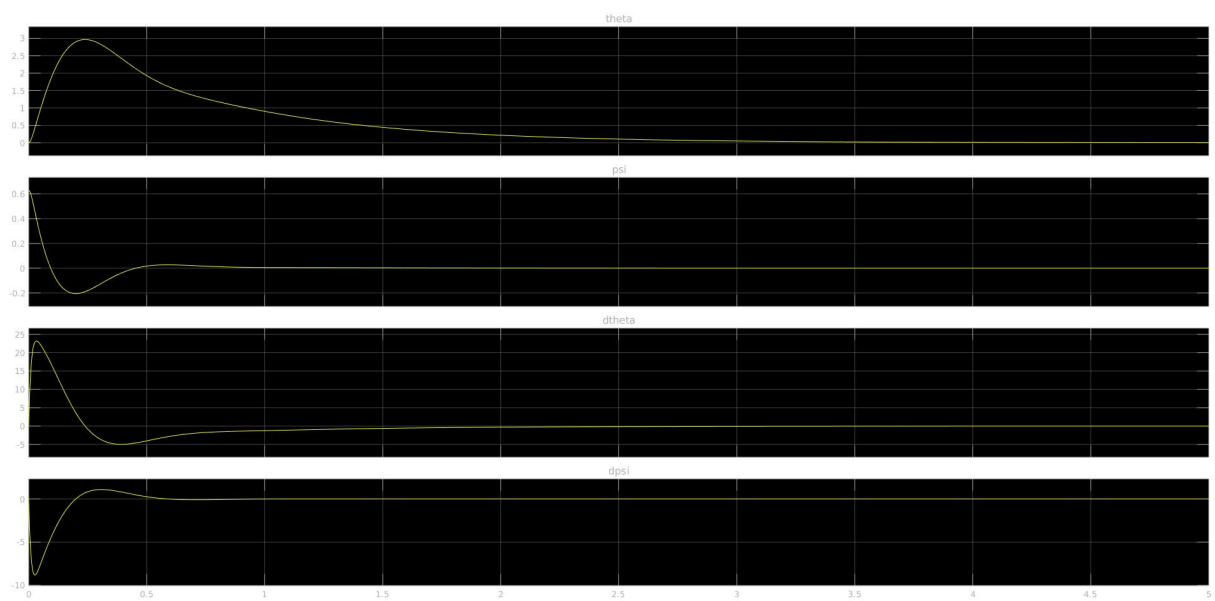


Figure 24:  $x(t)$

$x_0$	$t_f$	$K$	Intégrateur
$(0, \frac{\pi}{5}, 0, 0)$	5	$K$	ode45

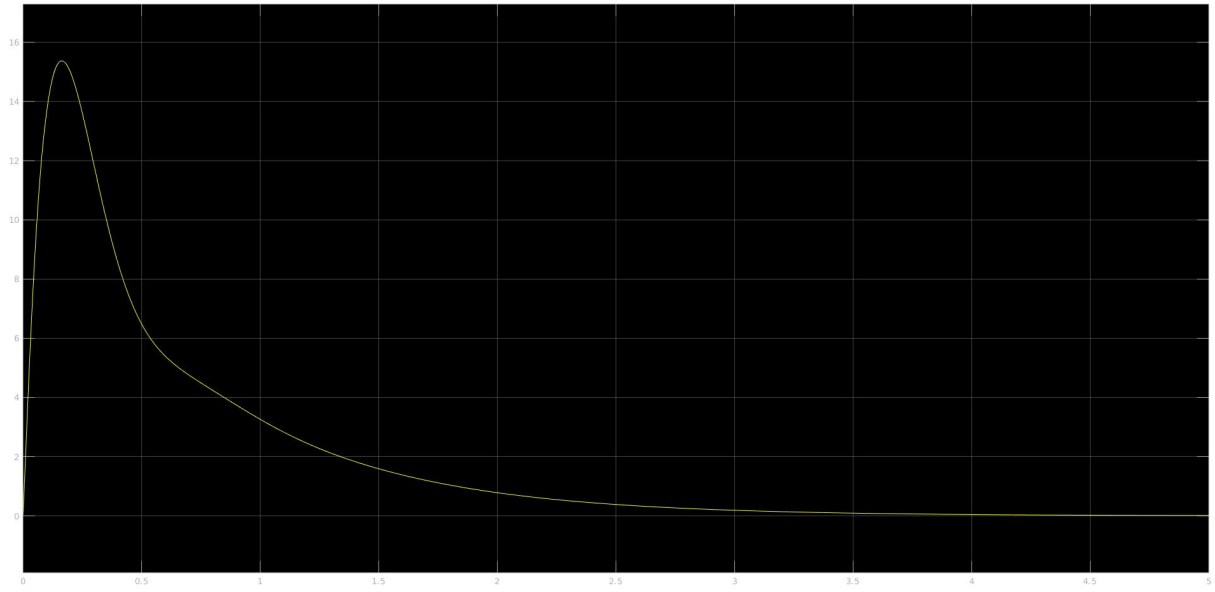


Figure 25:  $u(t)$

On remarque une différence avec la Figure 18,  $u(t)$  reste positif et converge de manière plus « douce », ce qui se traduit par un contrôle moins brusque.

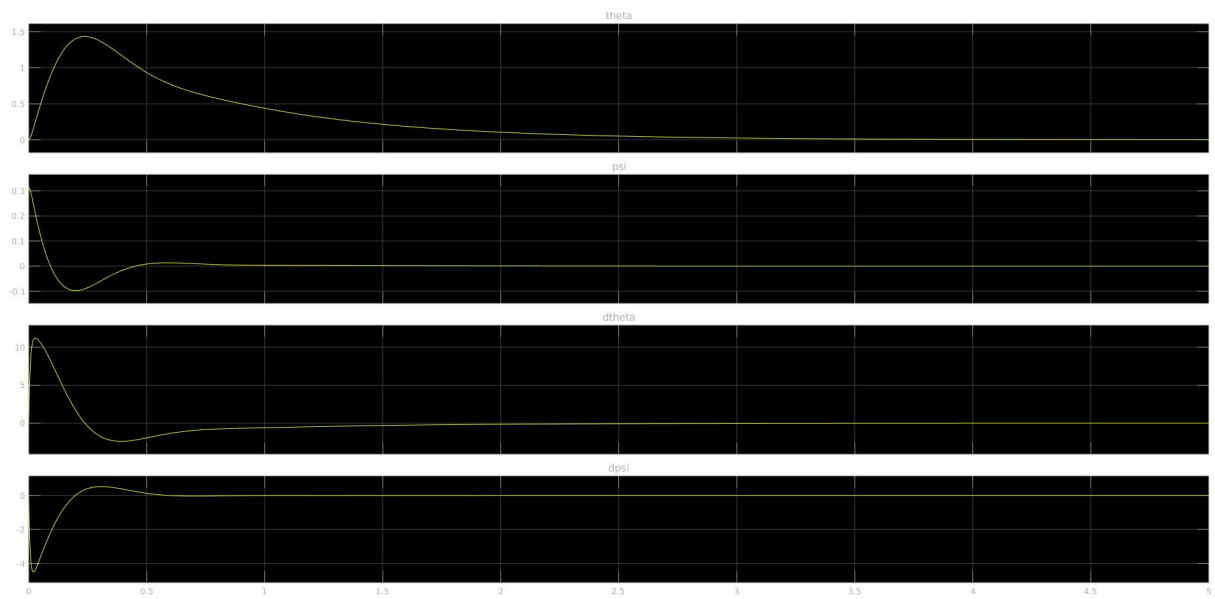


Figure 26:  $x(t)$

$x_0$	$t_f$	$K$	Intégrateur
$(0, \frac{\pi}{10}, 0, 0)$	5	$K$	ode45

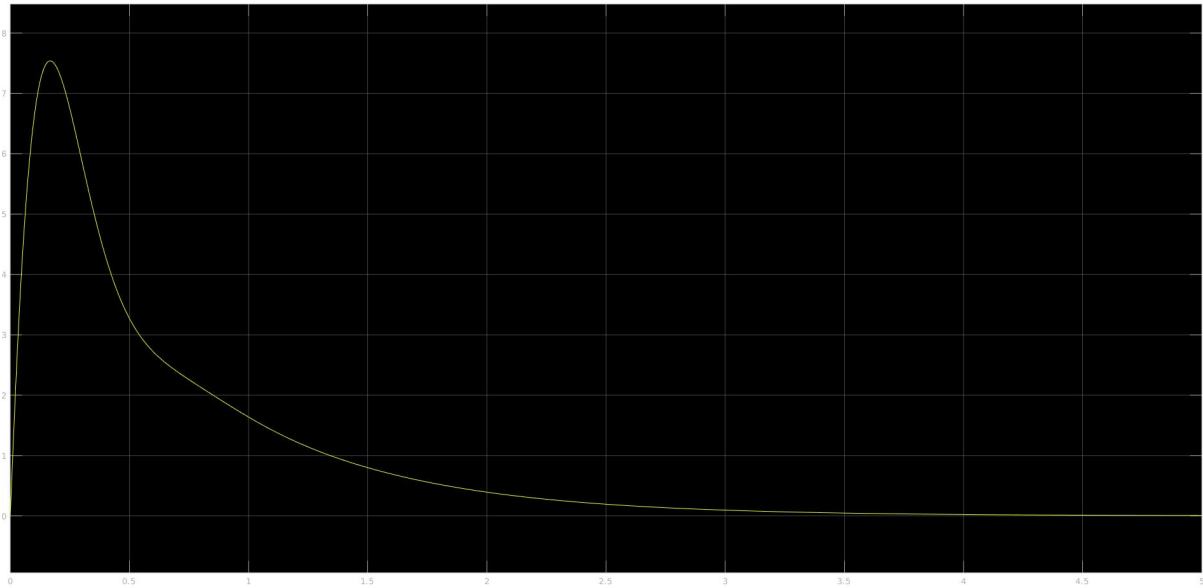


Figure 27:  $u(t)$

On retrouve un comportement similaire comparativement à la Figure 20.

## II - 2.3 Modèle hybride

Le modèle hybride est obtenu en recréant les variables  $\theta$  et  $\psi$  de manière discrète. On modifie notre schéma Simulink :

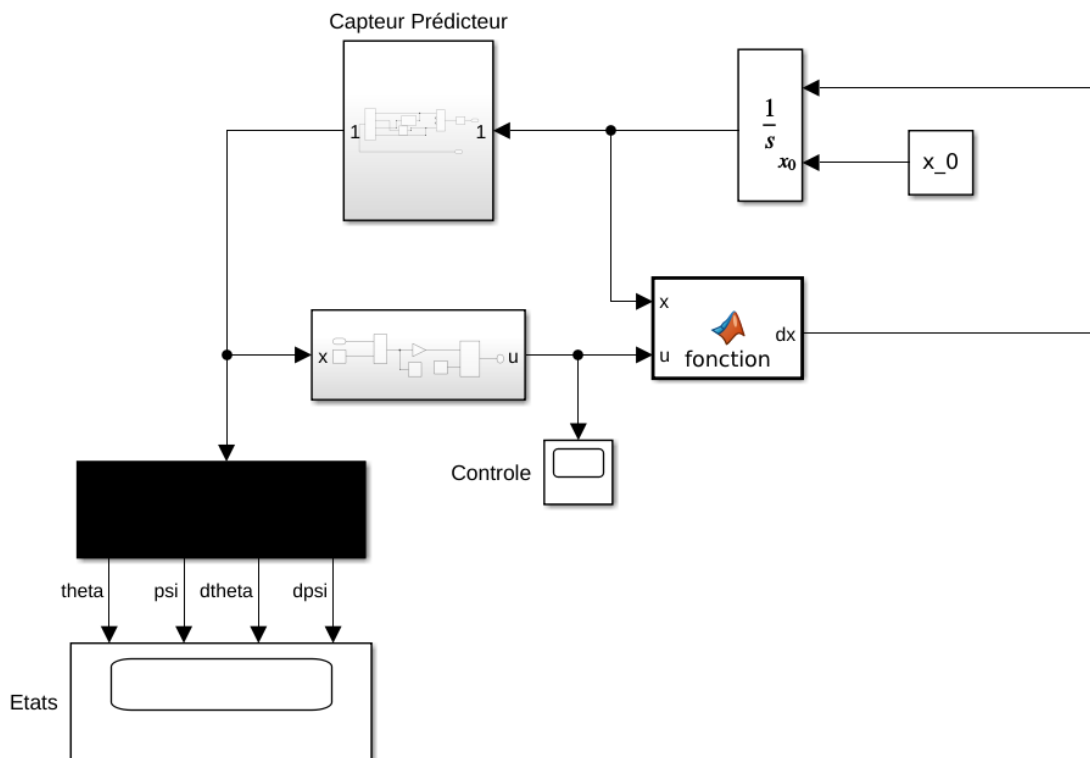


Figure 28: Schéma Simulink du système hybride contrôlé par retour d'état avec capteur et prédicteur

On introduit dans le capteur un bloc Zero-Order Hold et on utilise des opérateurs d'intégration et de dérivation discrets :

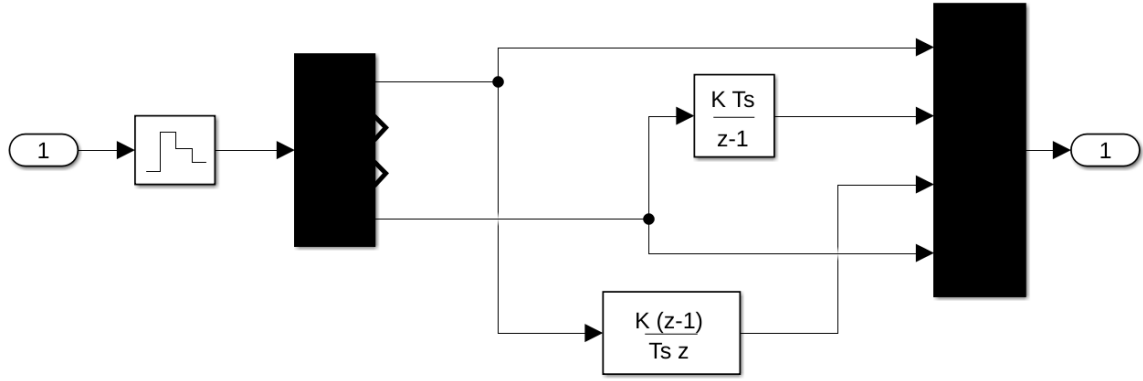


Figure 29: Schéma Simulink du capteur et prédicteur discrets

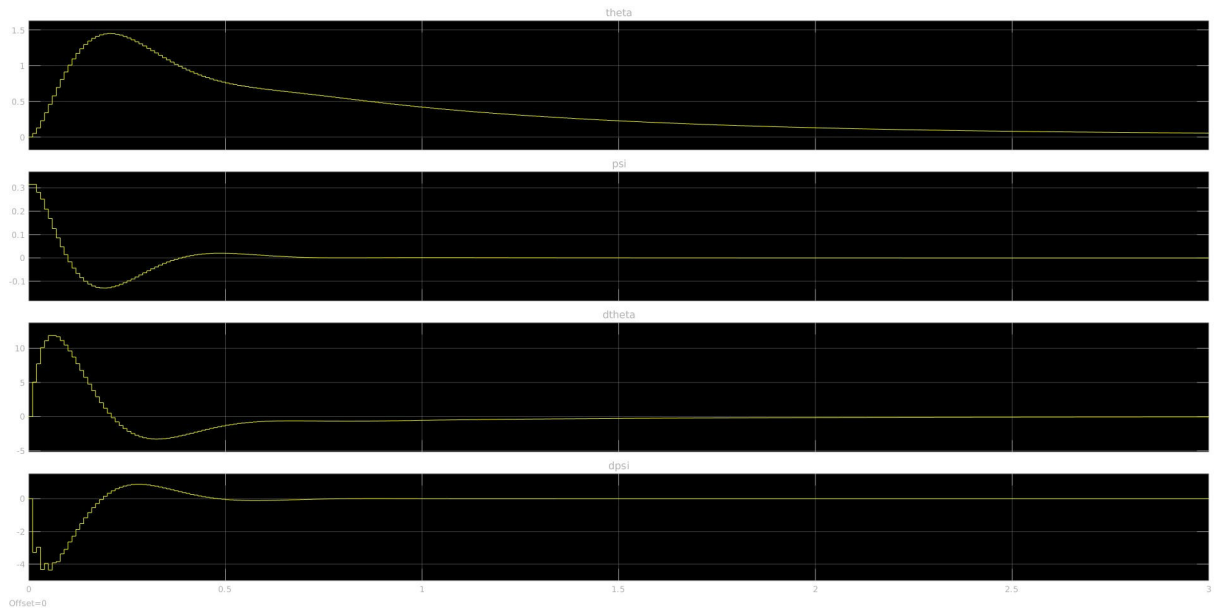


Figure 30:  $x(t)$

$x_0$	$t_f$	$K$	Intégrateur
$(0, \frac{\pi}{10}, 0, 0)$	3	$K$	ode45

La solution continue est ainsi construite à partir de la solution discrète.

On remarque que l'utilisation des signaux discrets a introduit du bruit dans la solution. Ceci est dû à leur approximation par le *solver* ode45.

### III Conclusion

Ces différents exercices de TP ont permis de mettre en pratique les notions de contrôle par retour d'état sur un système concret et de comparer les résultats selon les valeurs de la matrice  $K$  et les conditions initiales. On a ainsi pu confirmer le résultat théorique qui stipule que le pendule inversé est contrôlable par retour d'état lorsque  $k_1 > g$  et  $k_2 > 0$ .

L'introduction des capteurs et du prédicteur a permis d'ajouter du « réalisme » au modèle en nous rapprochant des conditions réelles d'implémentation du contrôle par retour d'état. On a ainsi pu implémenter une solution à cette perte d'information et en constater les conséquences sur le contrôle.



Finalement, l'utilisation de signaux discrets a permis de mettre en évidence les problèmes d'approximation numérique et de bruit qui peuvent en découler.