

# Rapport BE

THEVENET Louis

MARTIN Nolann

## Table des matières

1. Introduction .....	1
2. Travaux pratiques .....	1
2.1. TP2 - Modèle continu structuré du pendule inversé .....	1
2.1.1. Présentation .....	1
2.1.2. Contrôle par retour d'état .....	2
2.1.3. Simulation d'un capteur et d'un prédicteur .....	5
2.2. TP3 - Modèle continu et discret du robot Lego .....	6
2.2.1. Présentation .....	6
2.2.2. Introduction du capteur et du prédicteur .....	10
2.2.3. Modèle hybride .....	14

## 1. Introduction

à toi de jouer nouloun

- Brève introduction présentant le contexte du travail réalisé.
- Mentionnez l'objectif principal du rapport.

## 2. Travaux pratiques

### 2.1. TP2 - Modèle continu structuré du pendule inversé

#### 2.1.1. Présentation

On simule le pendule inversé contrôlé par retour d'état, le système dynamique qui le décrit s'écrit

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

avec

- $g = 9.81$
- $l = 10$
- $t_0 = 0$
- $x_e = (0, 0)$
- $u_e = 0$
- $u(t) = u_e + K(x(t) - x_e)$
- $K = (k_1, k_2)$

### 2.1.2. 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.

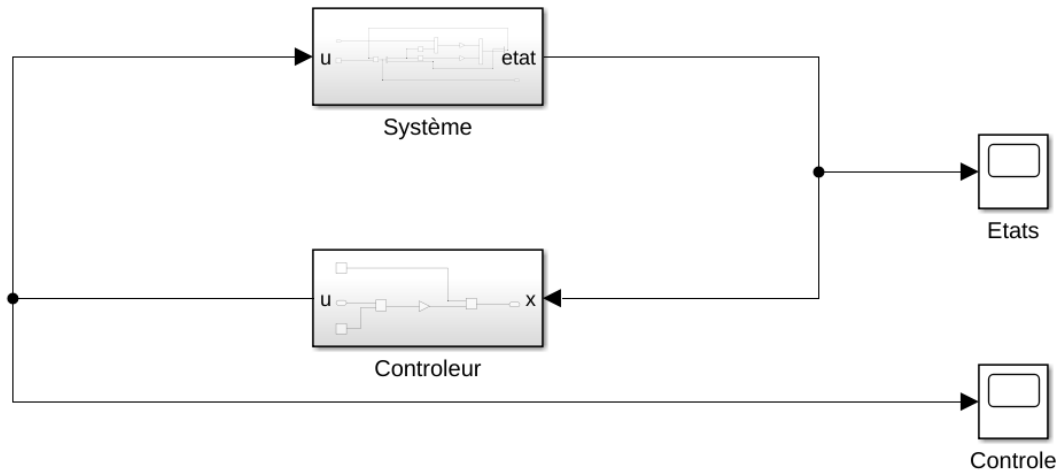


Figure 1: Schéma Simulink du système contrôlé par retour d'état

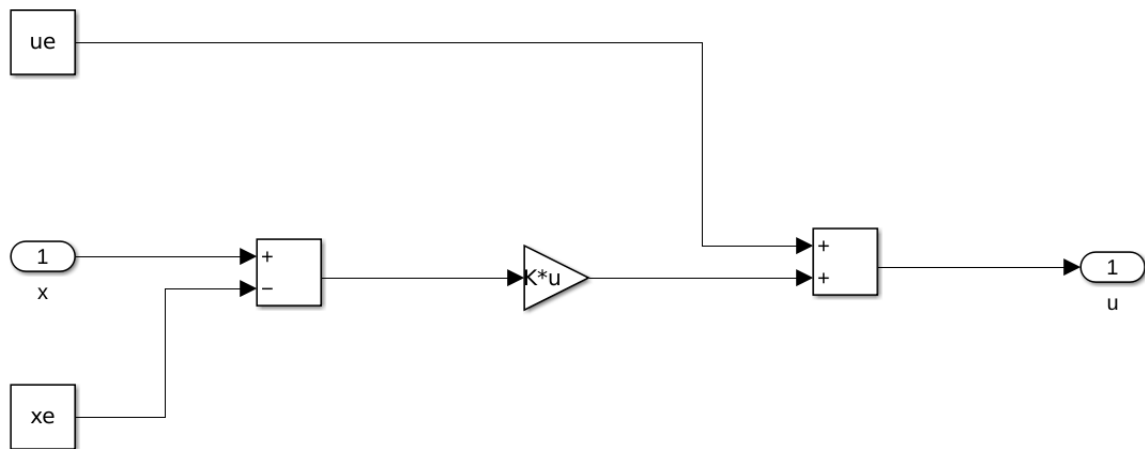


Figure 2: Schéma Simulink du contrôleur

### 2.1.2.1. Résultats pour le contrôle par retour d'état

On sait qu'il suffit d'avoir  $\begin{cases} k_1 > g \\ k_2 > 0 \end{cases}$  afin de contrôler asymptotiquement le système lorsque  $(\alpha_0, \dot{\alpha}_0)$  est suffisamment proche de  $x_e$ .

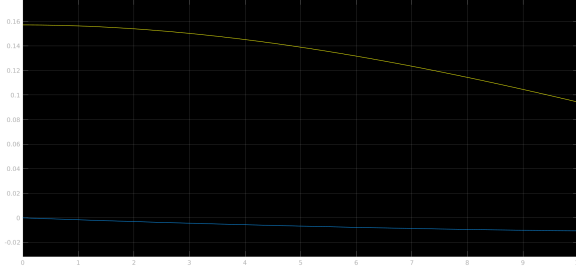


Figure 3: ■ :  $\alpha(t)$ , ■ :  $u(t)$

$x_0$	$t_f$	$K$	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.

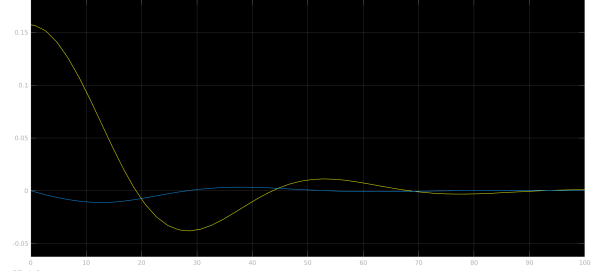


Figure 4: ■ :  $\alpha(t)$ , ■ :  $u(t)$

$x_0$	$t_f$	$K$	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  $x_e = (0, 0)$ .

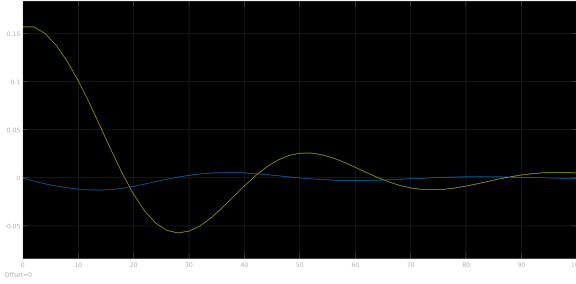


Figure 5: ■ :  $\alpha(t)$ , ■ :  $u(t)$

$x_0$	$t_f$	$K$	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.

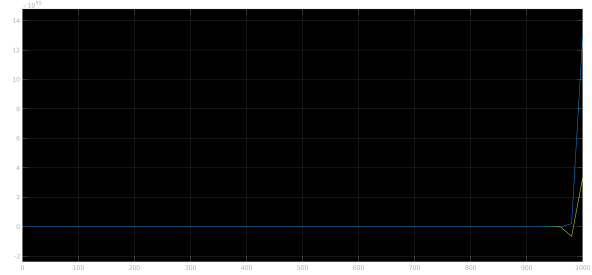


Figure 6: ■ :  $\alpha(t)$ , ■ :  $u(t)$

$x_0$	$t_f$	$K$	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.

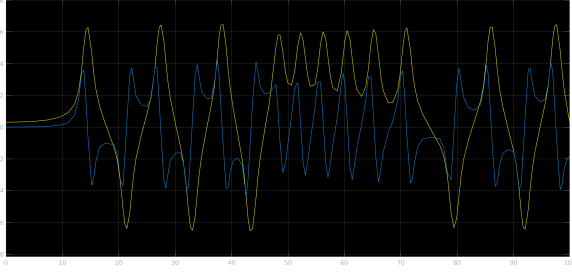


Figure 7: ■ :  $\alpha(t)$ , ■ :  $u(t)$

$x_0$	$t_f$	$K$	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.

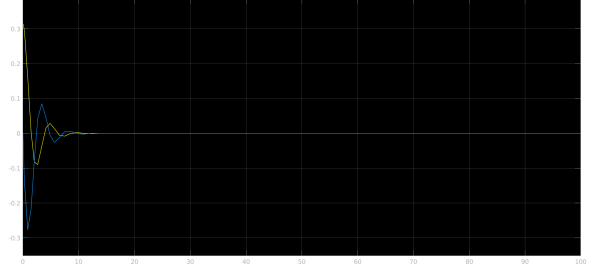


Figure 8: ■ :  $\alpha(t)$ , ■ :  $u(t)$

$x_0$	$t_f$	$K$	Intégrateur
$(\frac{\pi}{10}, 0)$	100	(30, 10)	ode45

En augmentant les valeurs de  $k_1$  et  $k_2$ , le système est de nouveau contrôlé asymptotiquement.

### 2.1.3. 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

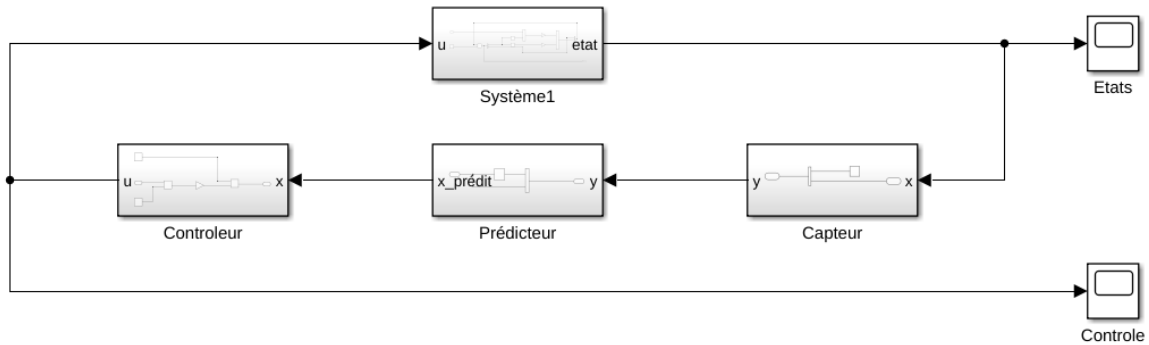


Figure 9: Schéma Simulink du système contrôlé par retour d'état avec capteur et prédicteur

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.

#### 2.1.3.1. Résultats pour le contrôle par retour d'état 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

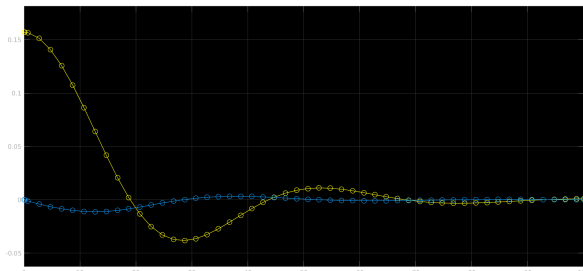


Figure 10: Pas = par défaut

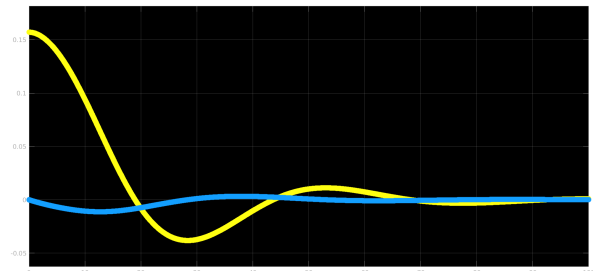


Figure 11: Pas = 0.001

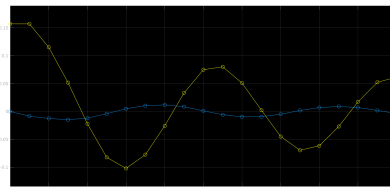


Figure 12: Pas = 5

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.

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

## 2.2. TP3 - Modèle continu et discret du robot Lego

### 2.2.1. Présentation

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

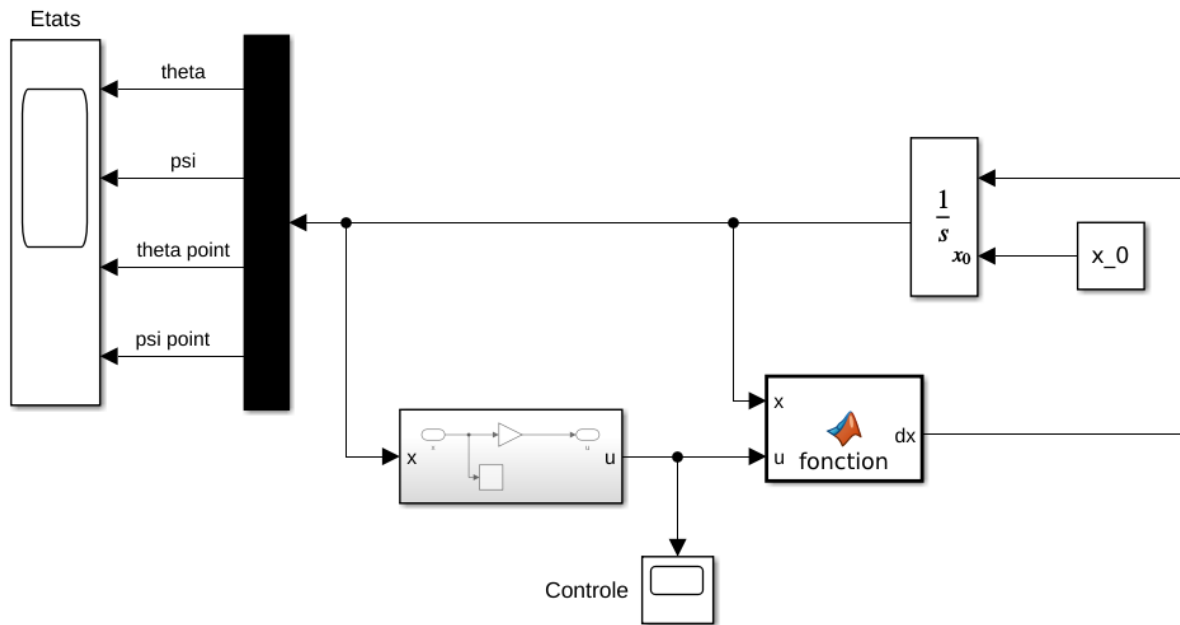


Figure 13: Schéma Simulink du système contrôlé par retour d'état

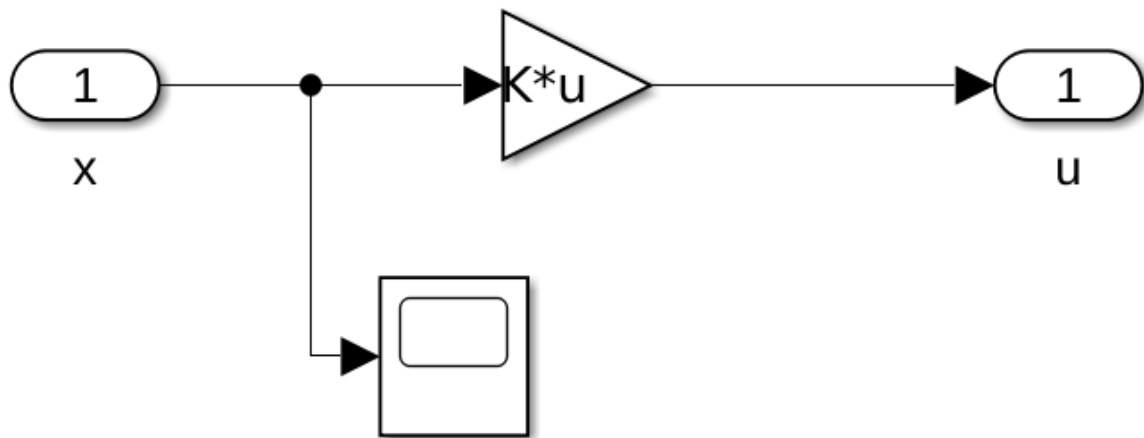


Figure 14: Schéma Simulink du contrôle par retour d'état

Le vecteur  $K$  est calculé à part 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);
```

#### 2.2.1.1. Résultats pour le contrôle par retour d'état

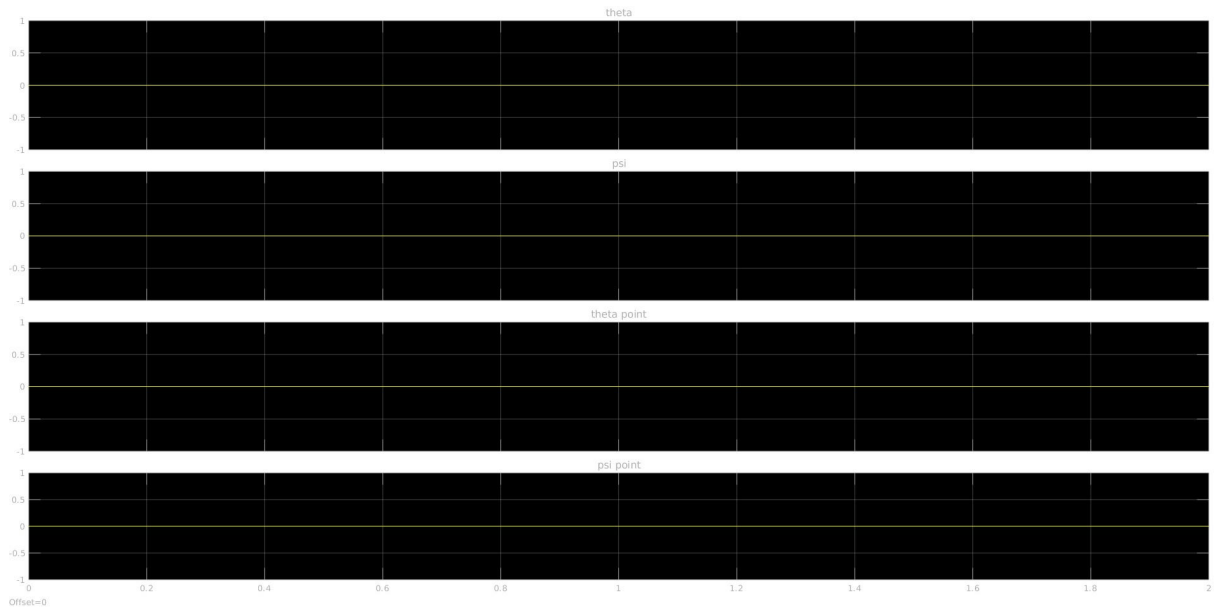


Figure 15:  $x(t)$

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

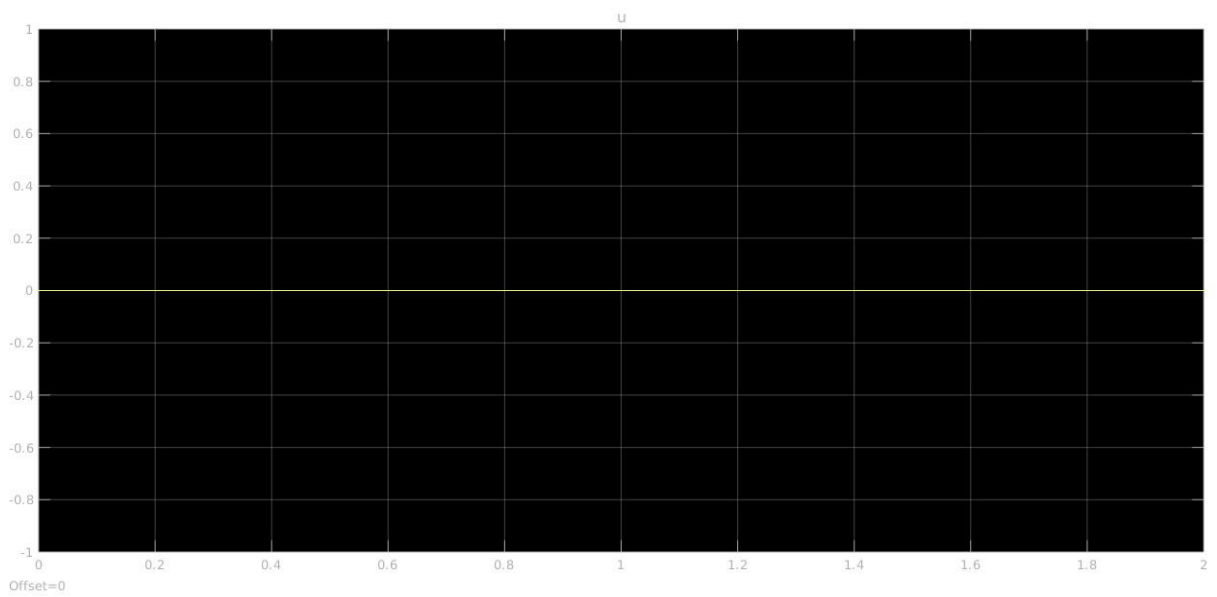


Figure 16:  $u(t)$

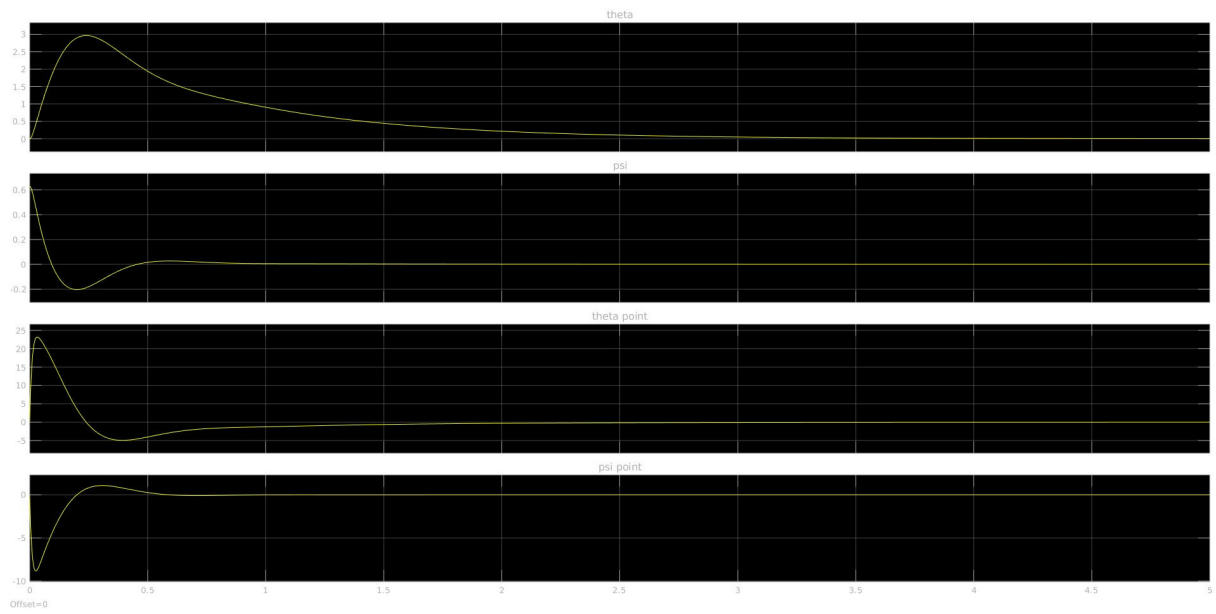


Figure 17:  $x(t)$

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

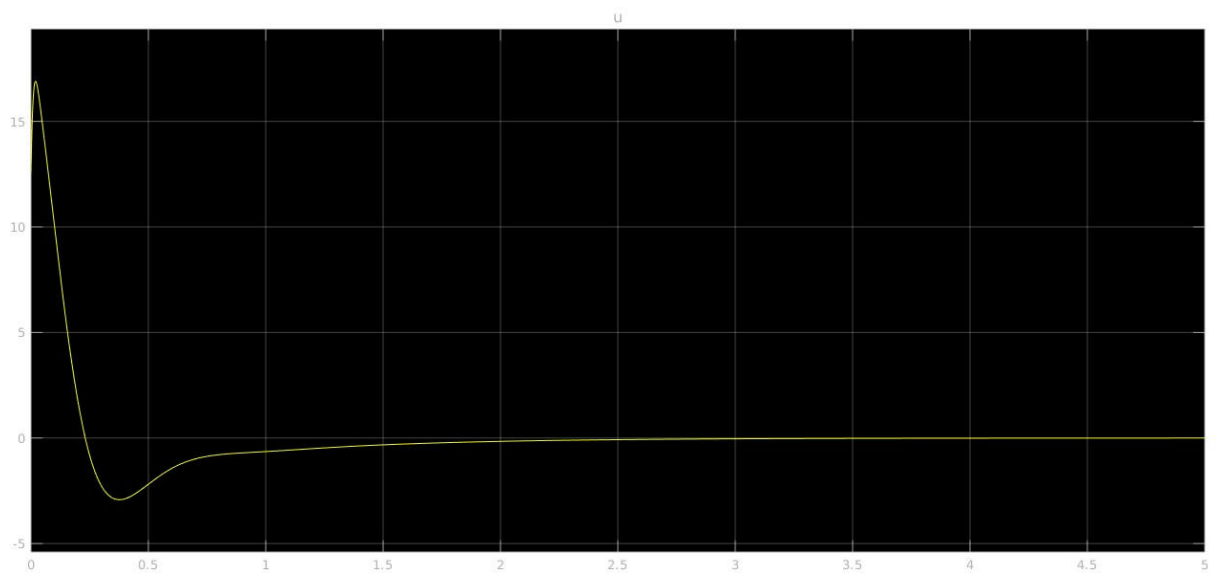


Figure 18:  $u(t)$



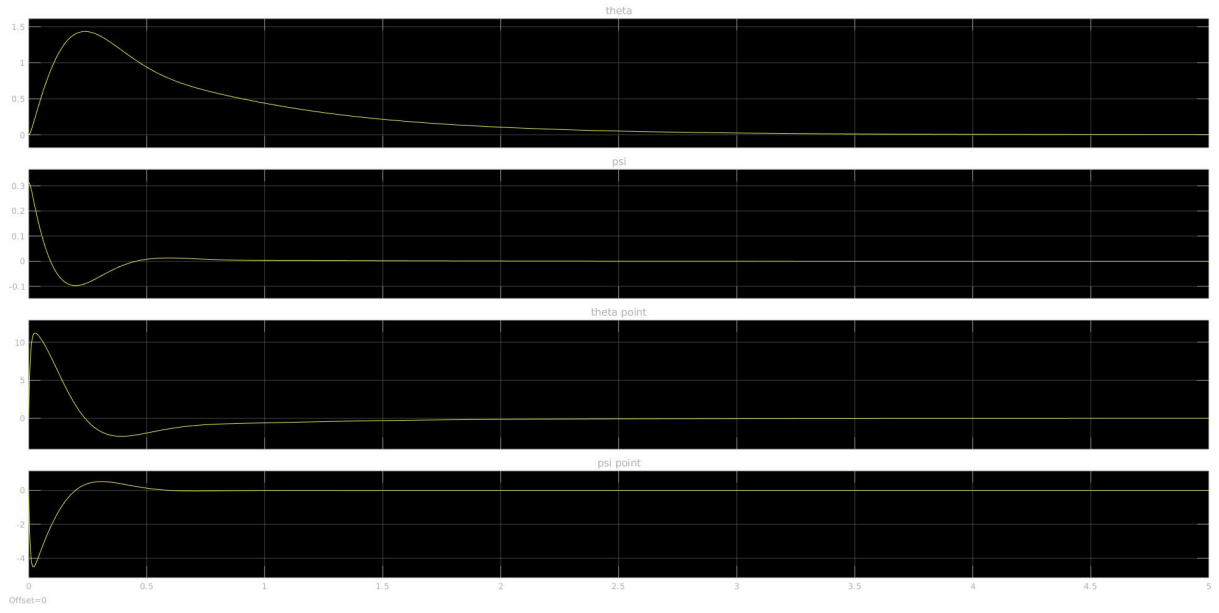


Figure 19:  $x(t)$

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

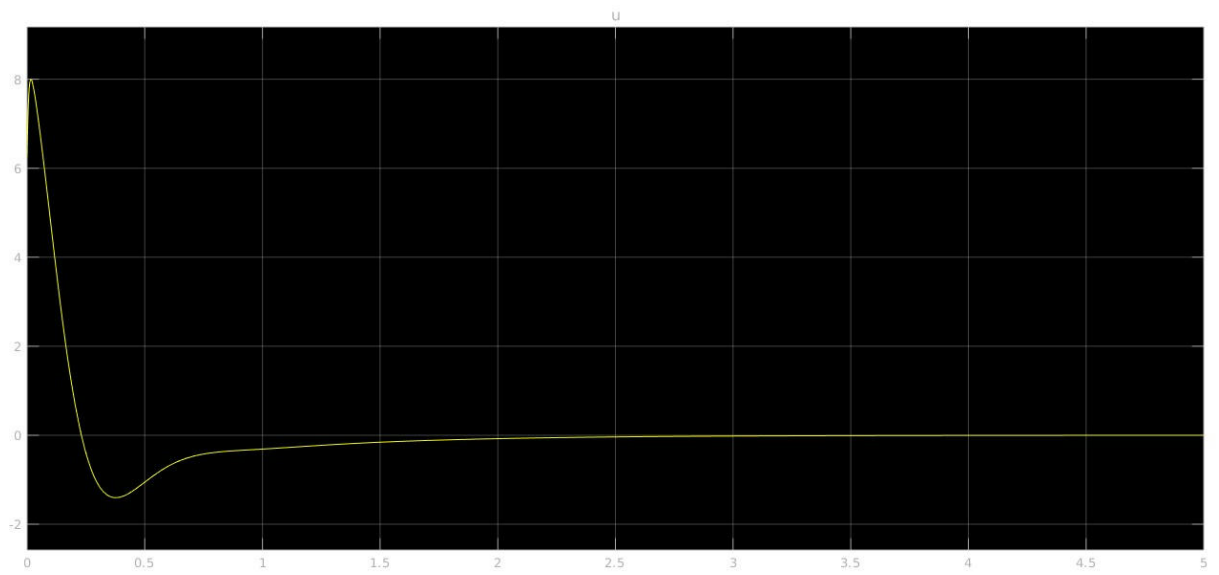


Figure 20:  $u(t)$

### 2.2.2. Introduction du capteur et du prédicteur

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

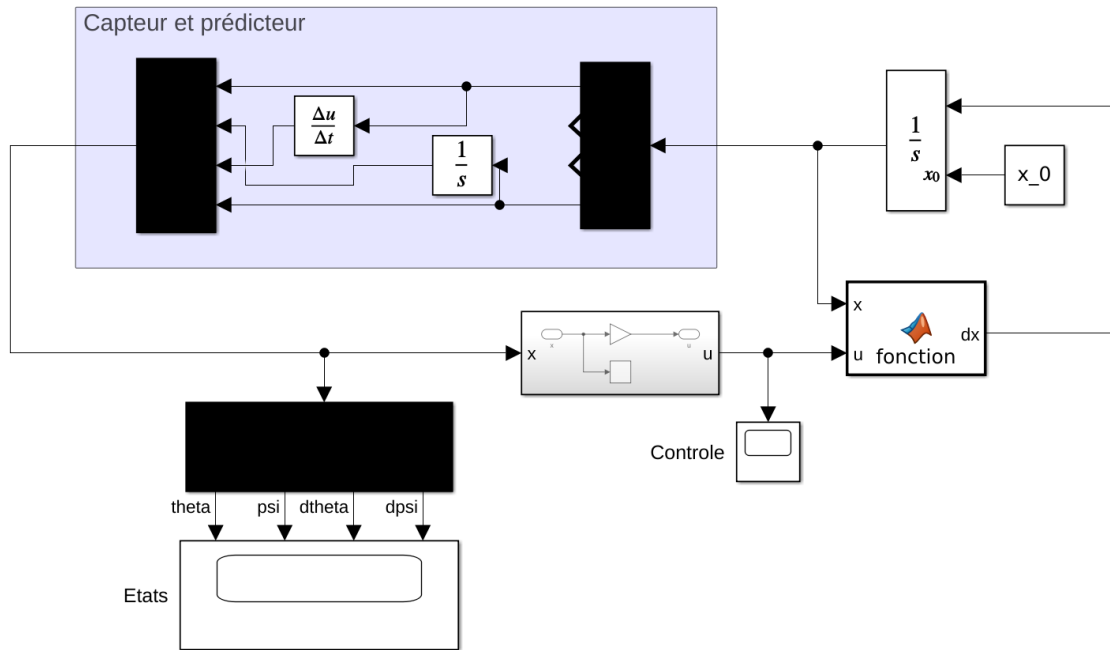


Figure 21: Schéma Simulink du système contrôlé par retour d'état avec capteur et prédicteur

Le capteur est modélisé par le bloc **Demux** qui permet de sélectionner les composantes  $\dot{\varphi}(t)$  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.

### 2.2.2.1. Résultats pour le contrôle par retour d'état avec capteur et prédicteur

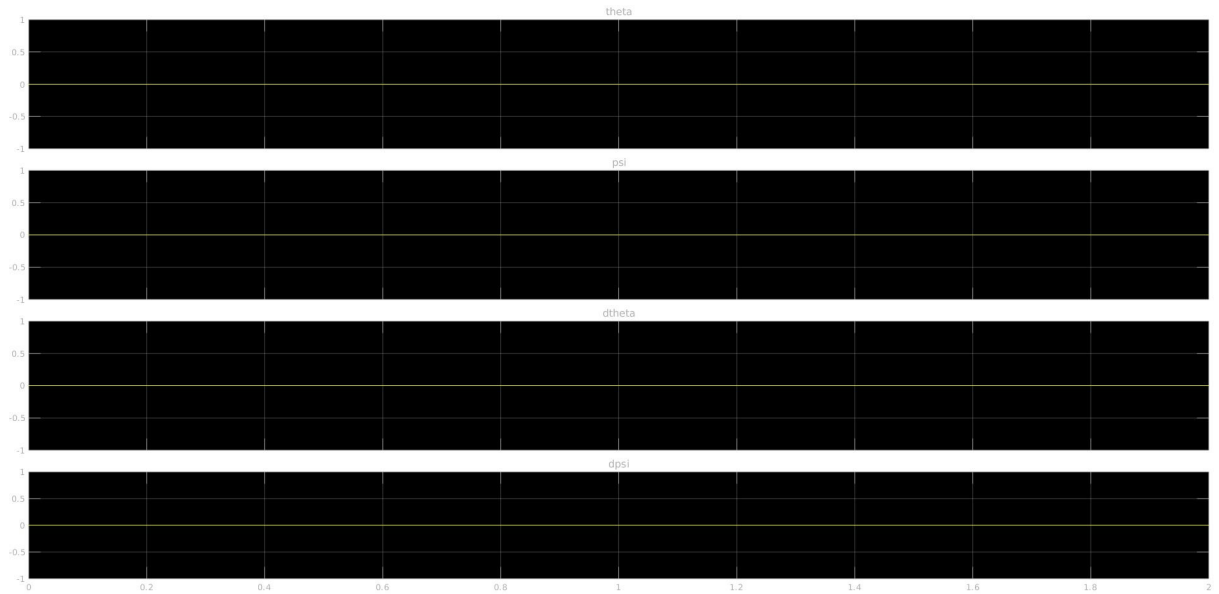


Figure 22:  $x(t)$

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

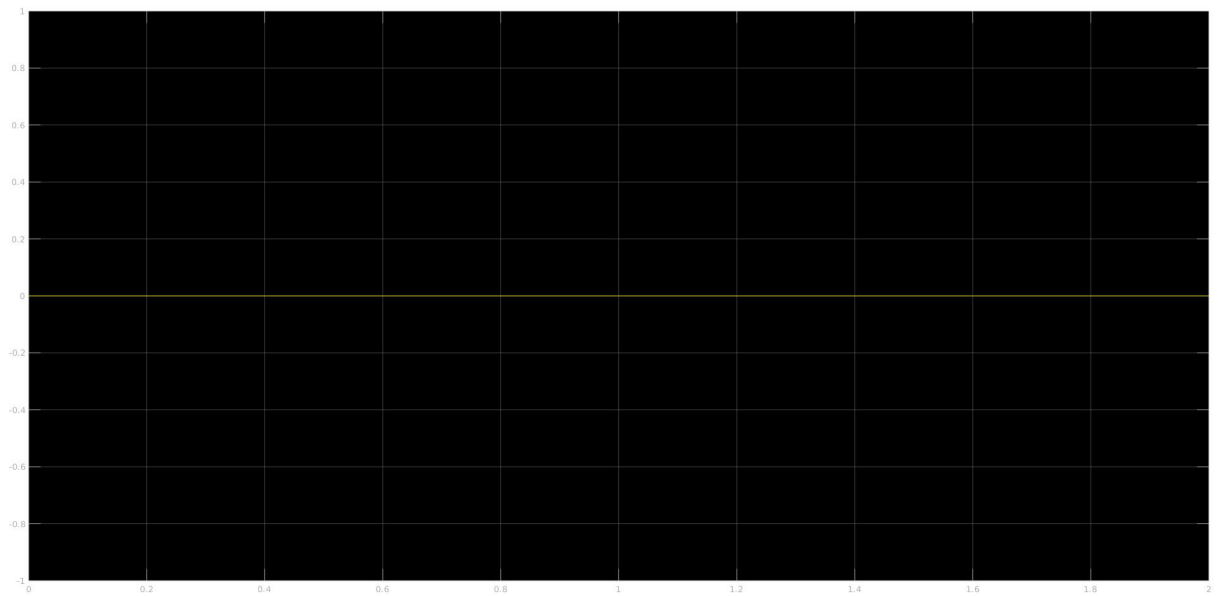


Figure 23:  $u(t)$

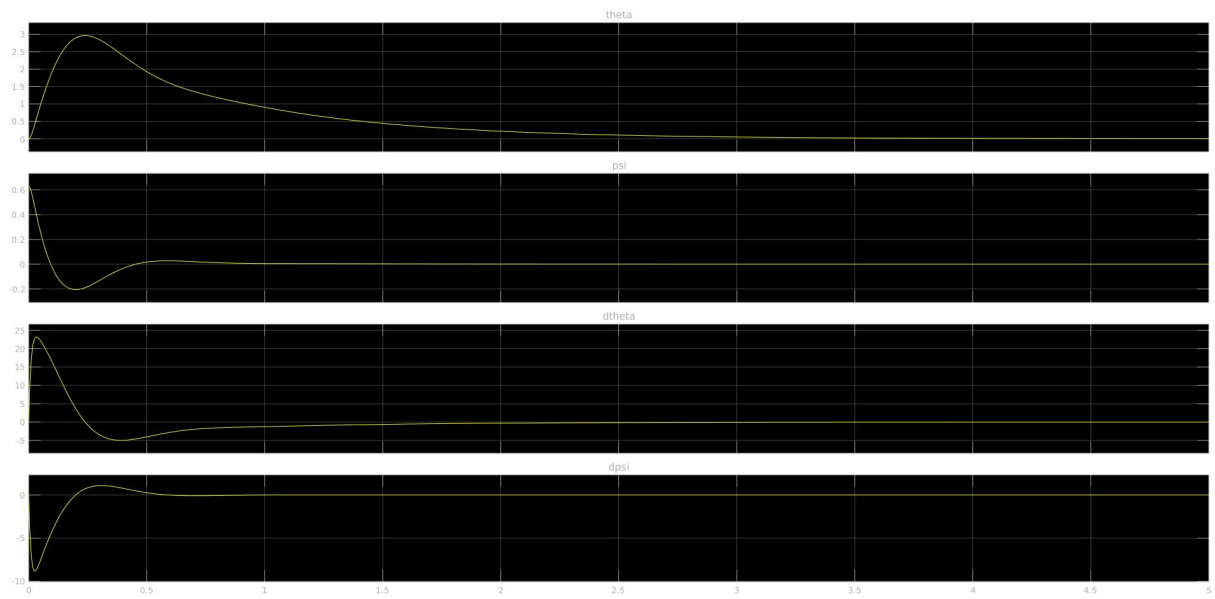


Figure 24:  $x(t)$

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

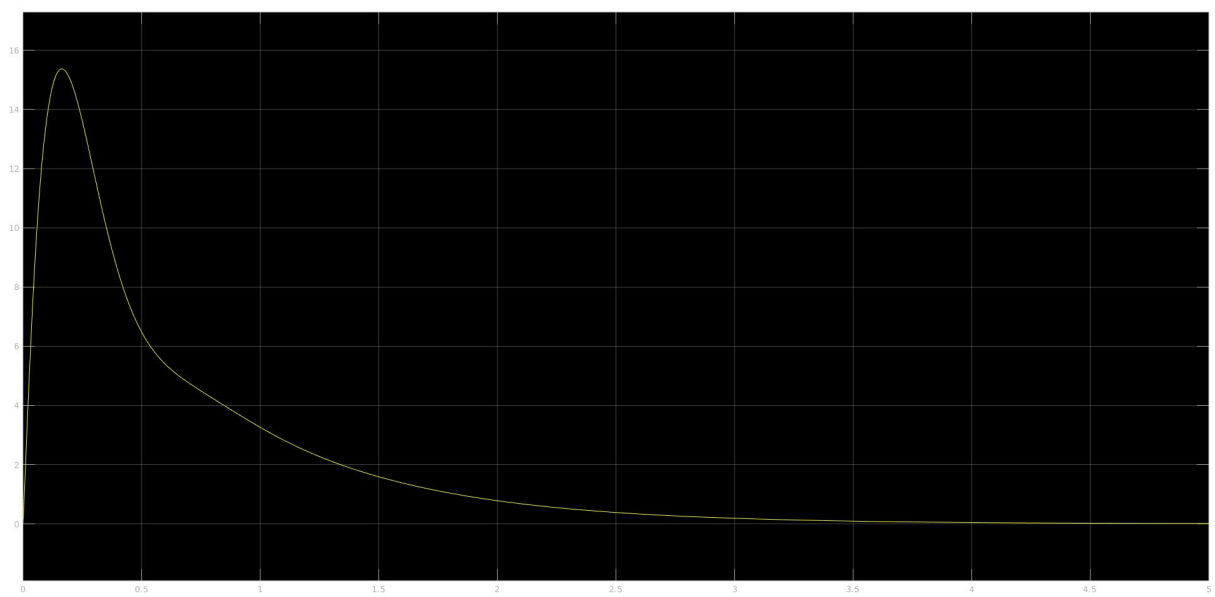


Figure 25:  $u(t)$

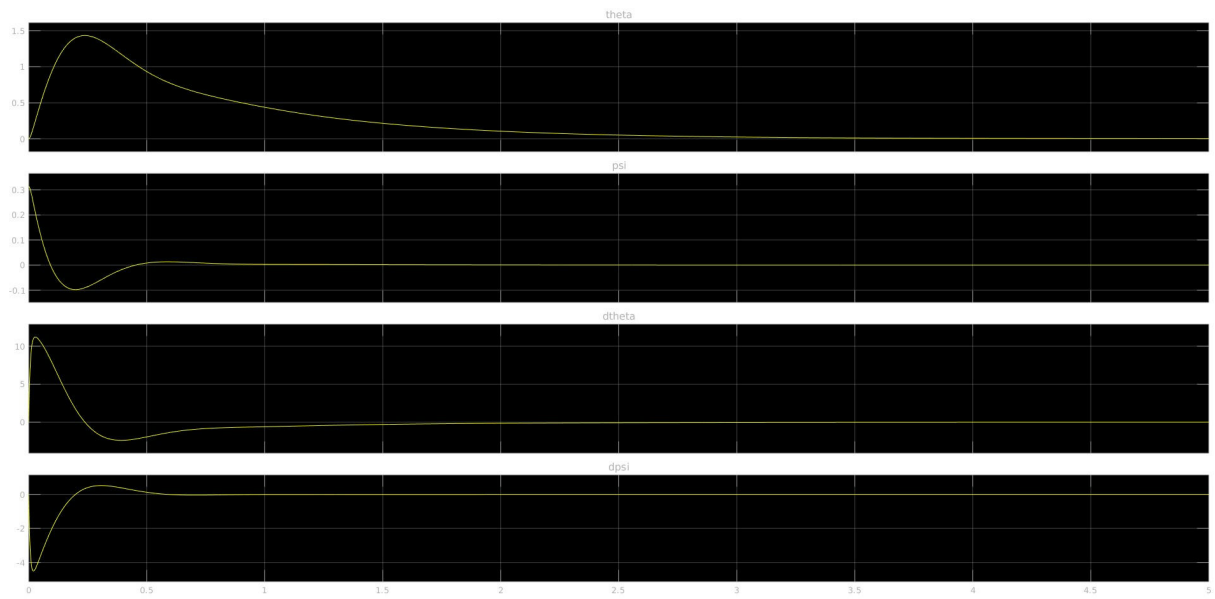


Figure 26:  $x(t)$

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

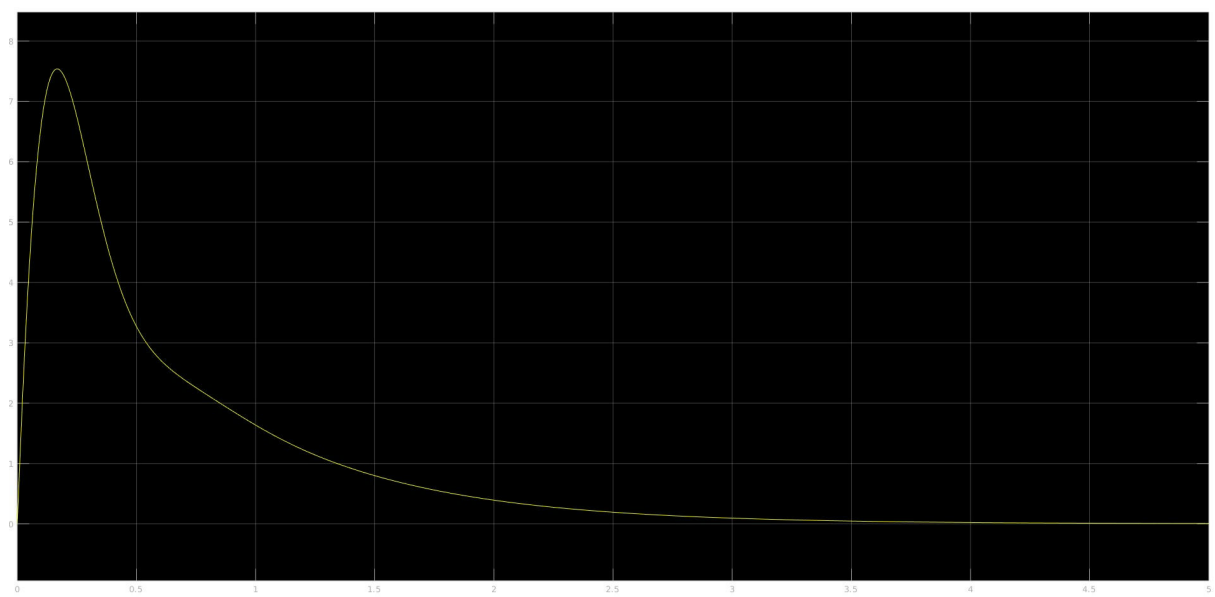


Figure 27:  $u(t)$

On modifie notre schéma Simulink :

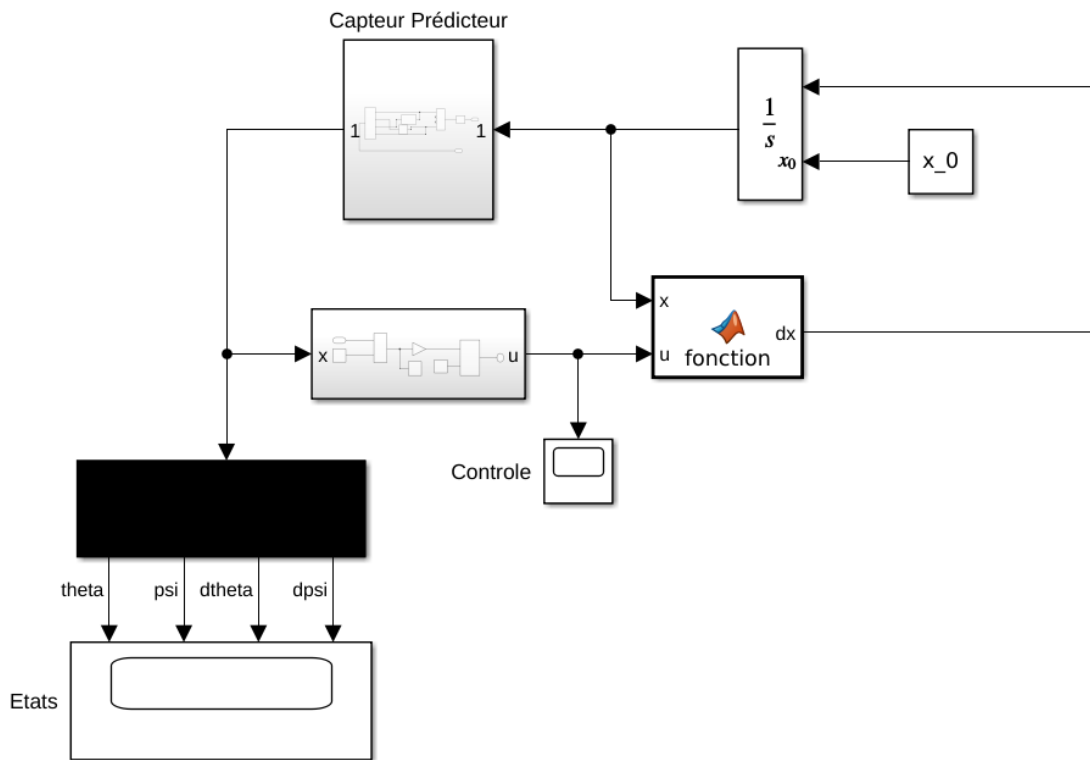


Figure 28: Schéma Simulink du système hybride contrôlé par retour d'état avec capteur et prédicteur

On introduit dans le capteur un bloc **Zero-Order Hold** et on utilise des opérateurs discrets :

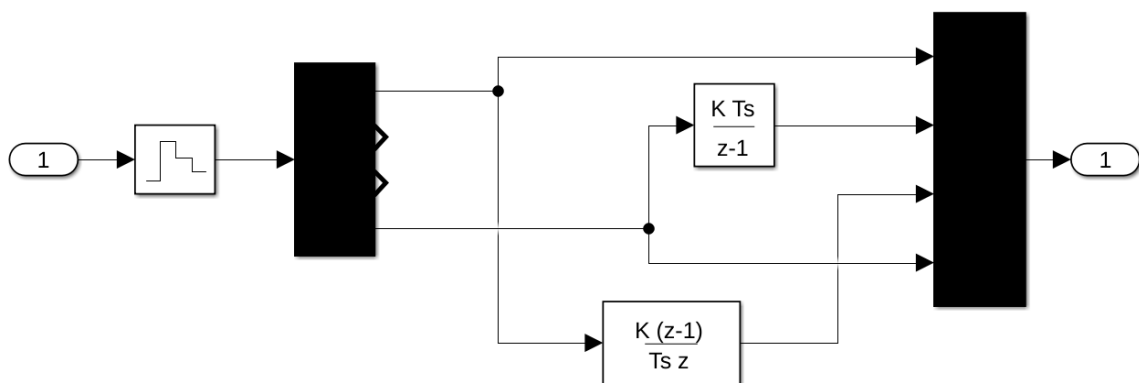


Figure 29: Schéma Simulink du capteur et prédicteur discrets

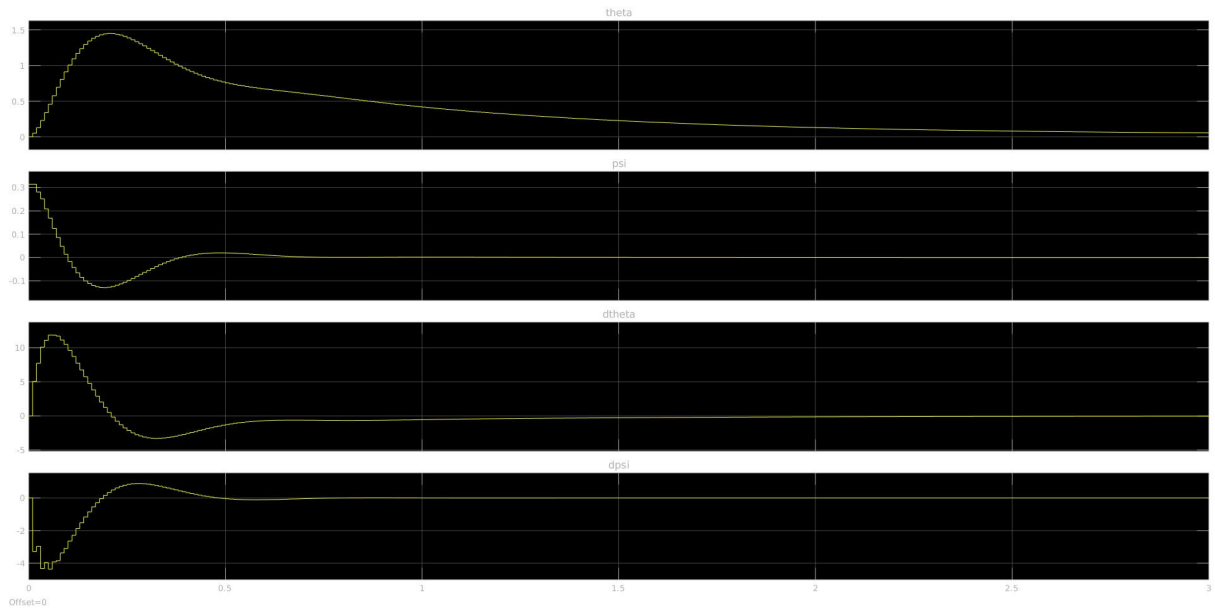


Figure 30:  $x(t)$

$x_0$	$t_f$	$K$	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 la solution discrète est plus bruitée que la continue.