Bontinck Laérian, CM1-F SN 1A Fallot Ysabel, CM1-E SN 1A

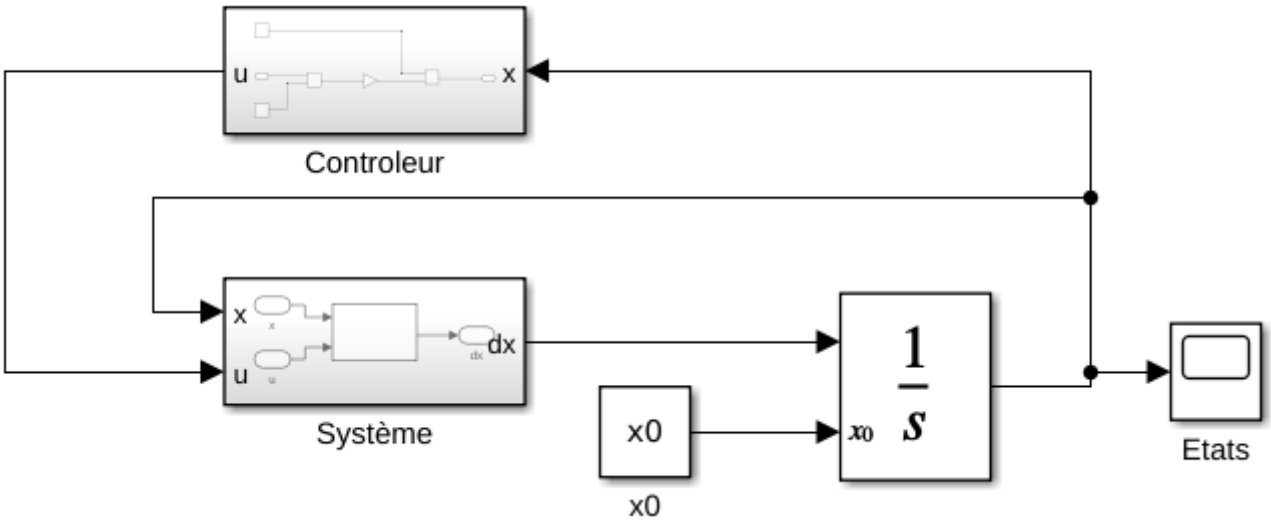
Automatique -- TP n°3

Simulation du robot LEGO pendule inversé

1 - Modèle continu

1. Utilisation de Matlab pour représenter le modèle du robot

En modifiant un peu le modèle utilisé au TP précédent, on obtient ce modèle :



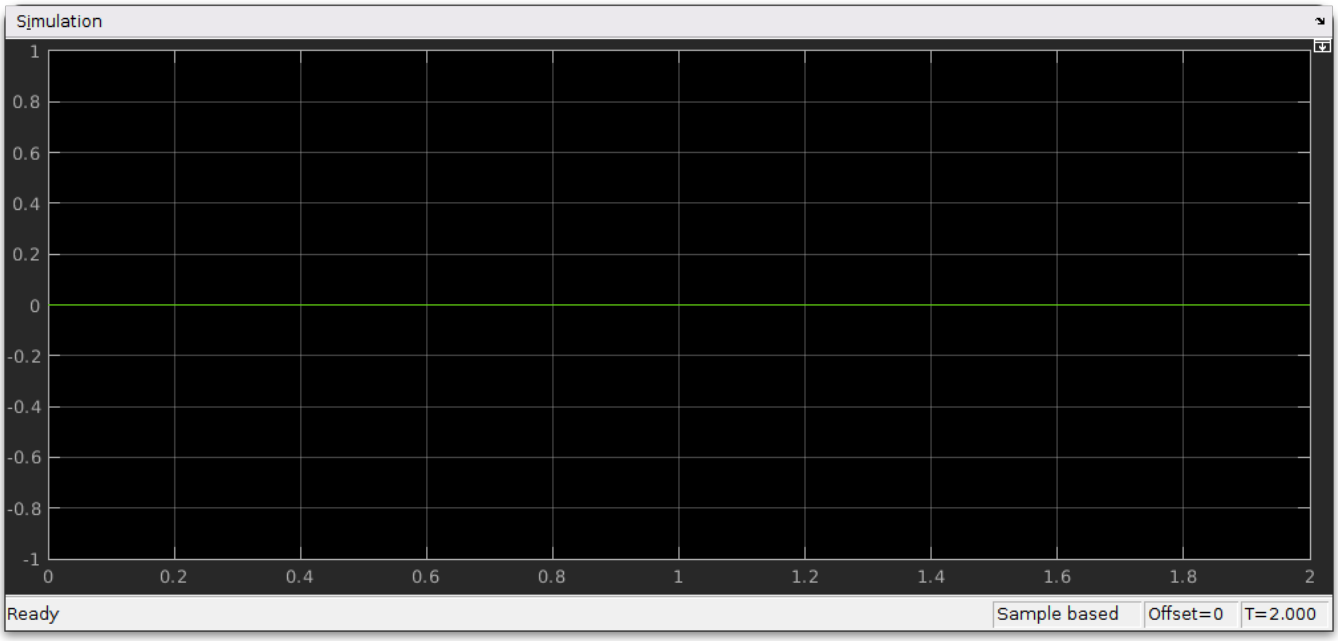
Le modèle a du mal à simuler, dû au manque de bons coefficients pour K. C'est justement l'objectif de la sous-partie suivante.

2. Synthèse du contrôleur pour le modèle du robot Lego pendule inversé

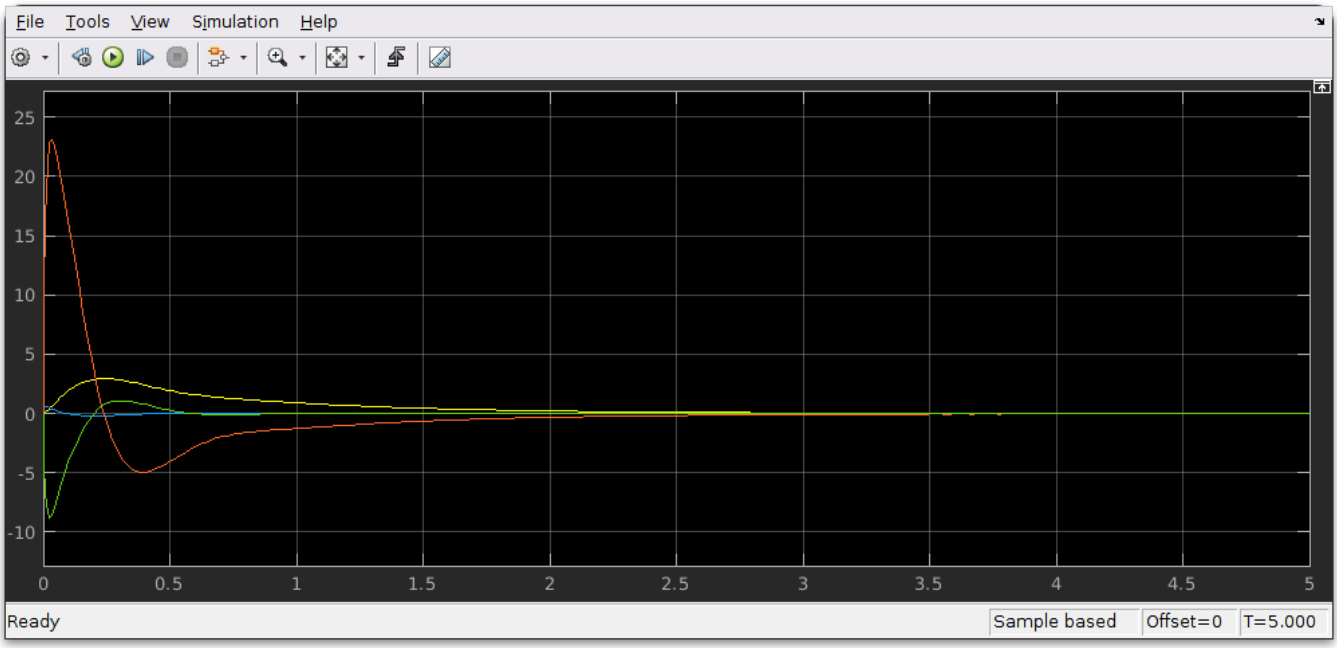
On ajoute les simples lignes suivantes au fichier `matrices.m` :

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

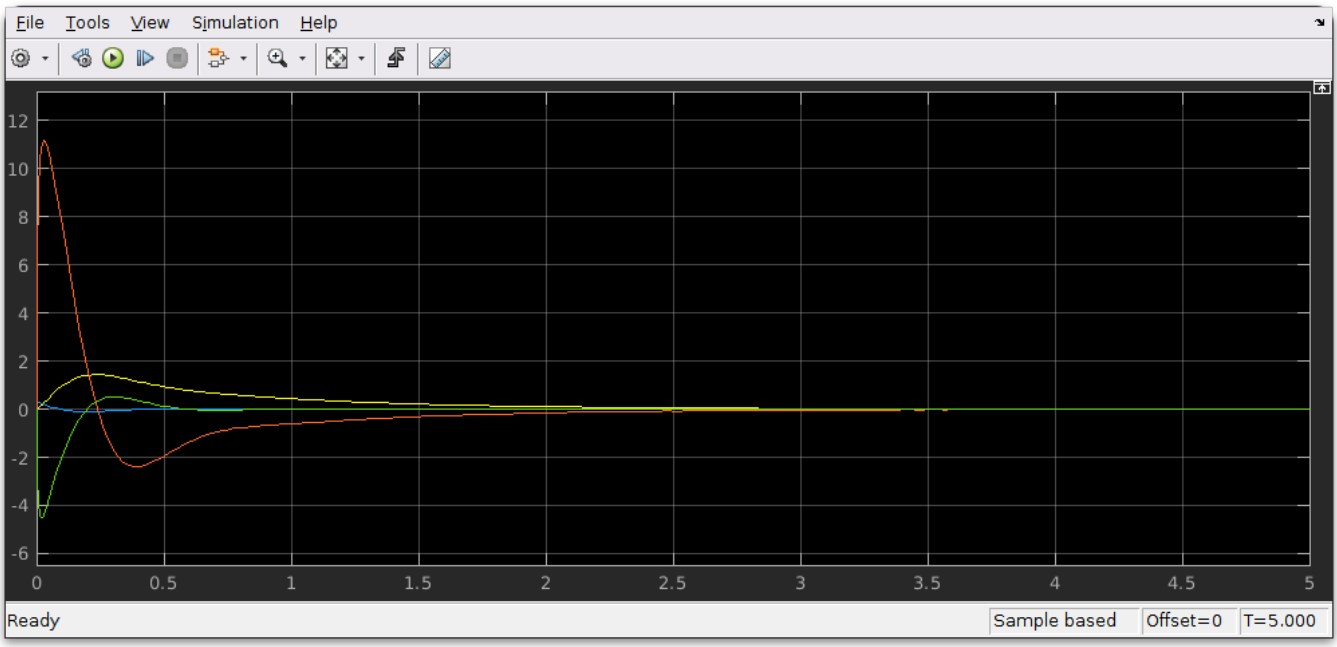
Maintenant que l'on a calculé le vecteur des gains K, on peut simuler le modèle.



- **Cas 1.1** : $\begin{cases} K = (0.67, 19.90, 1.07, 1.96) \\ x_0 = (0,0,0,0) \\ u_e = 0 \\ x_e = (0,0,0,0) \\ t_f = 10 \end{cases}$, méthode ODE45 (Dormand-Prince)
  - Ce cas ne nous donne pas beaucoup de résultats intéressants : en commençant au point d'équilibre, la seule qualité du système que l'on évalue est sa capacité à rester en ce même point, et par définition d'un point d'équilibre ce n'est pas bien compliqué. Au moins le système ne dégénère pas en son point d'équilibre.



- **Cas 1.2** : $x_0 = (0, \frac{\pi}{5}, 0, 0)$ et $t_f = 5$
    - On constate dans ce cas que le système converge correctement.

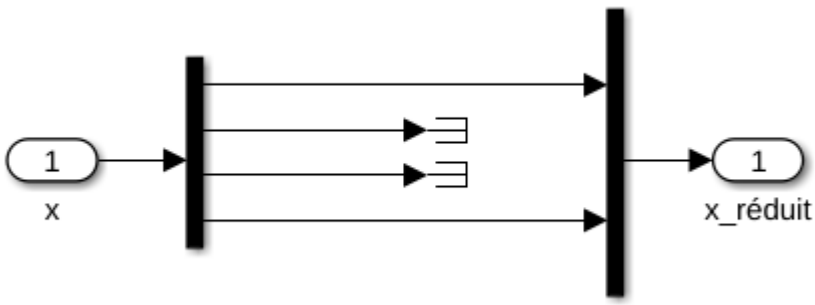


- **Cas 1.3** : $x_0 = (0, \frac{\pi}{10}, 0, 0)$
    - On constate que le système réagit exactement de la même manière à amplitude différente lorsque qu'on change l'angle initial θ_0

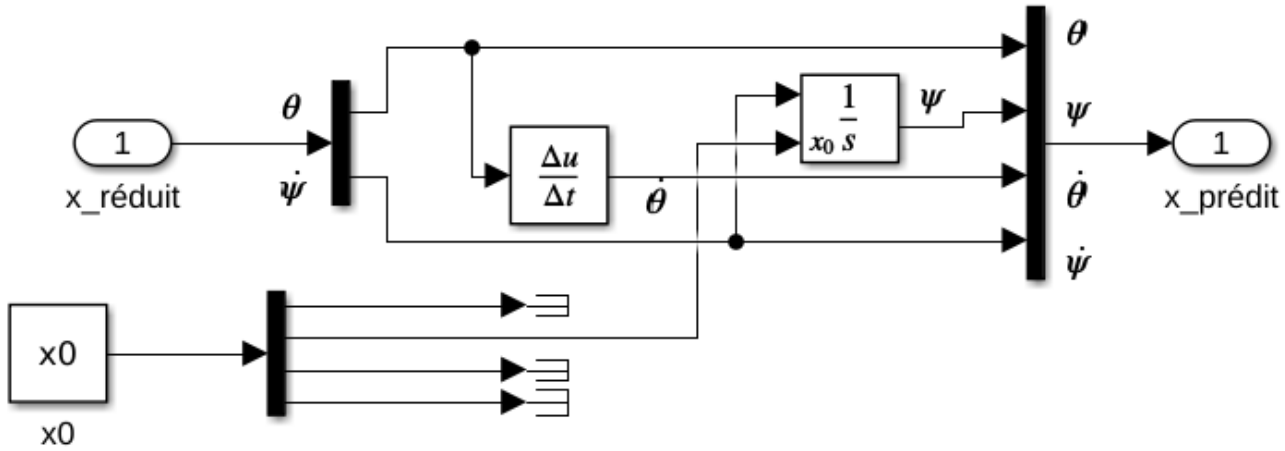
2 - Introduction des capteurs et actionneurs

On modélise deux nouveaux blocs :

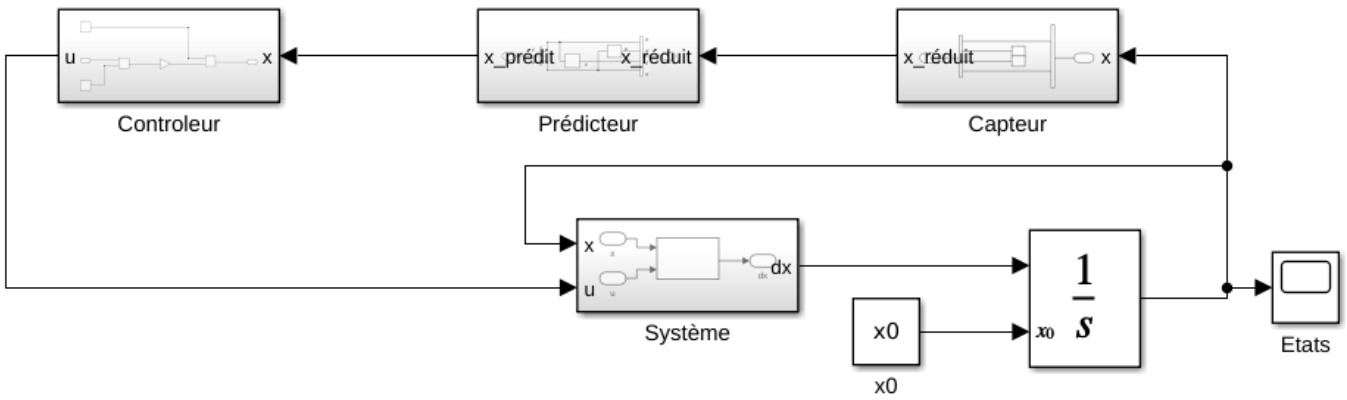
- un bloc **Capteur**, qui réduit les données.



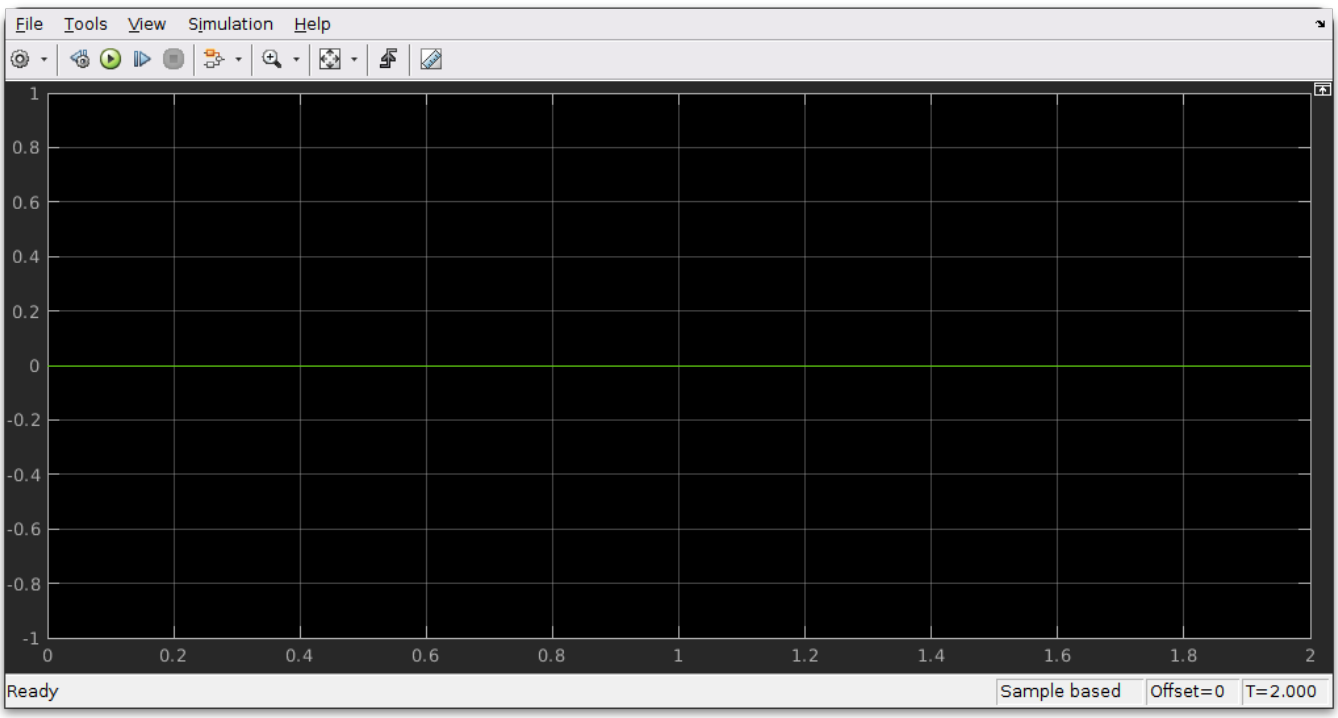
- un bloc **Prédicteur**, qui calcule les données perdues avec celles que l'on a encore



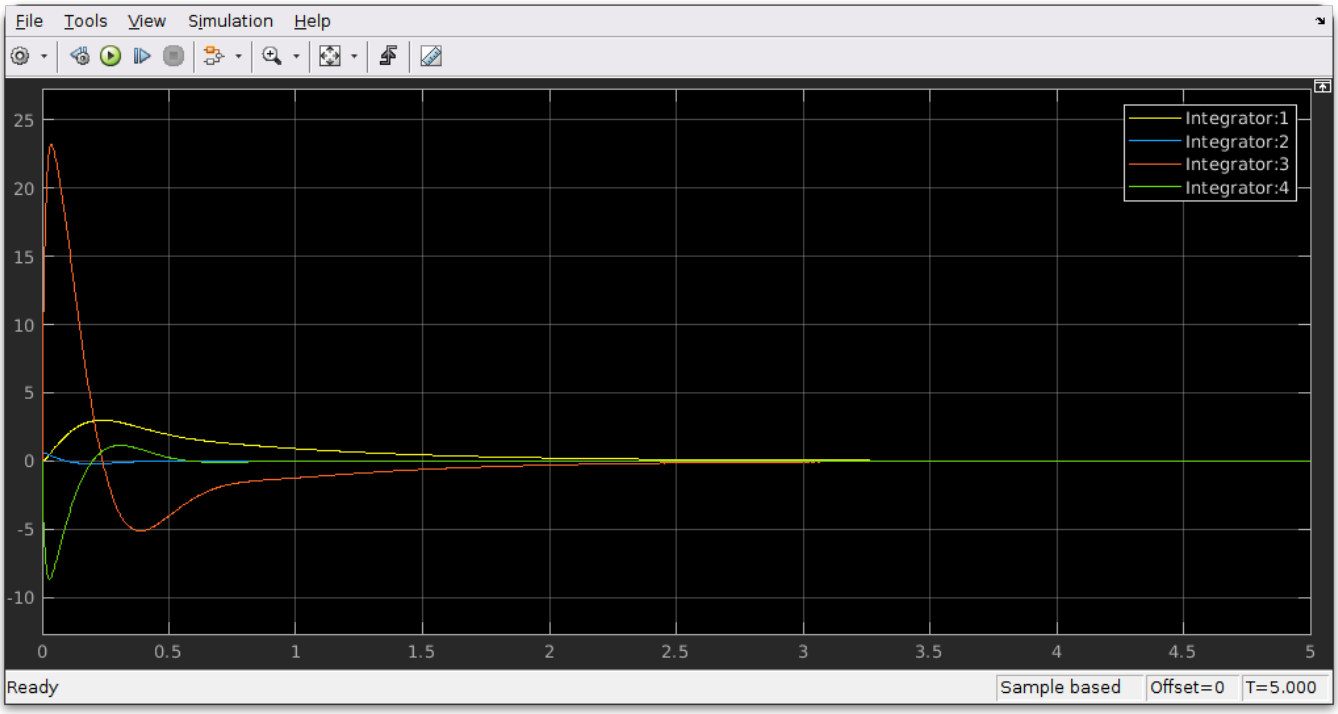
Le nouveau modèle devient alors ceci :



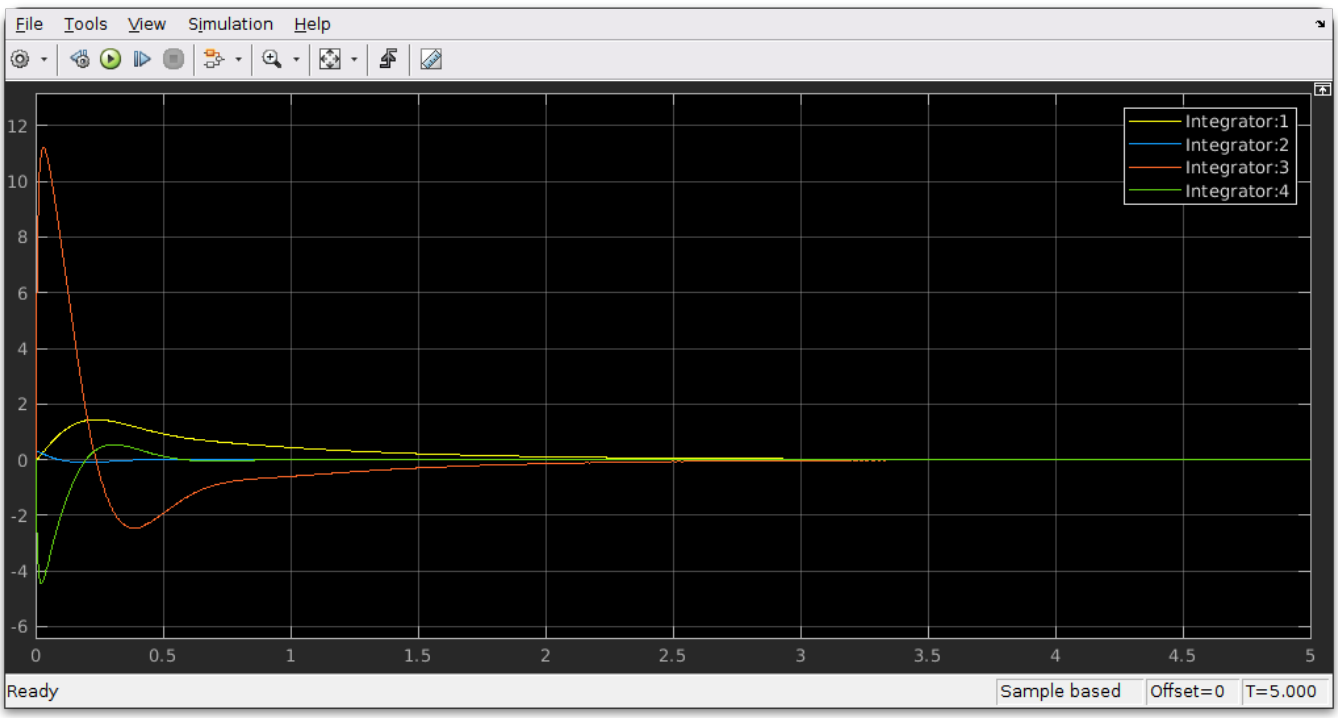
On simule avec ce nouveau modèle :



- **Cas 2.1** : cas analogue au cas 1.1
    - On constate les mêmes résultats qu'au cas analogue. Le système reste stable en son point d'équilibre.



- **Cas 2.2** : cas analogue au cas 1.2
    - On constate les mêmes résultats (ou presque) : le système converge correctement !

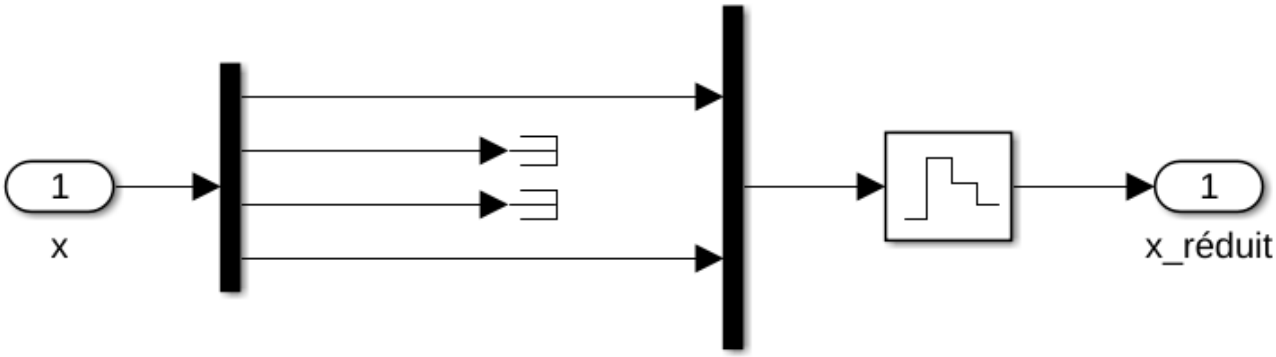


- **Cas 2.3** : cas analogue au cas 1.3
    - Encore une fois, on constate quasiment les mêmes résultats.

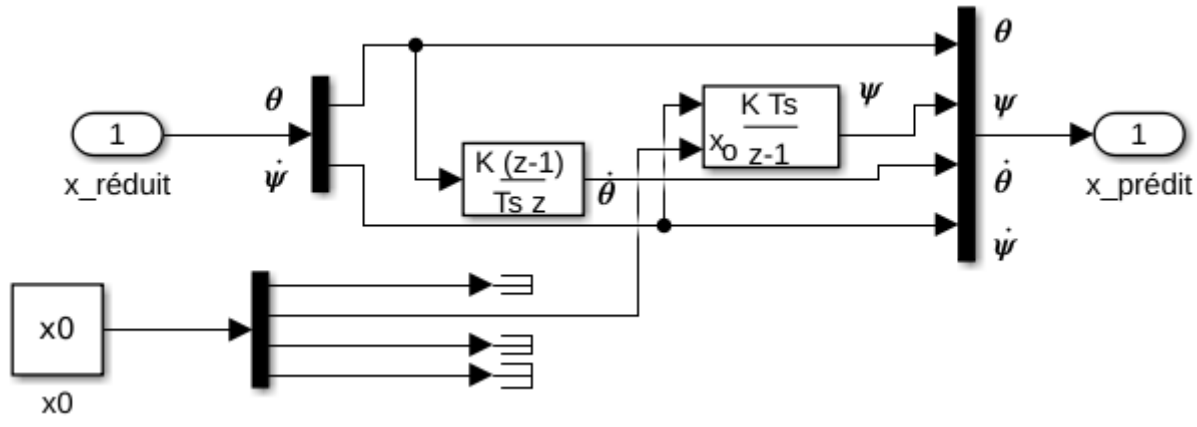
3 - Construction du modèle hybride

On modifie les sous-systèmes Capteur et Prédicteur :

- On introduit un bloc **Zero-Order Hold** dans le sous-sytème Capteur pour obtenir des données discrètes

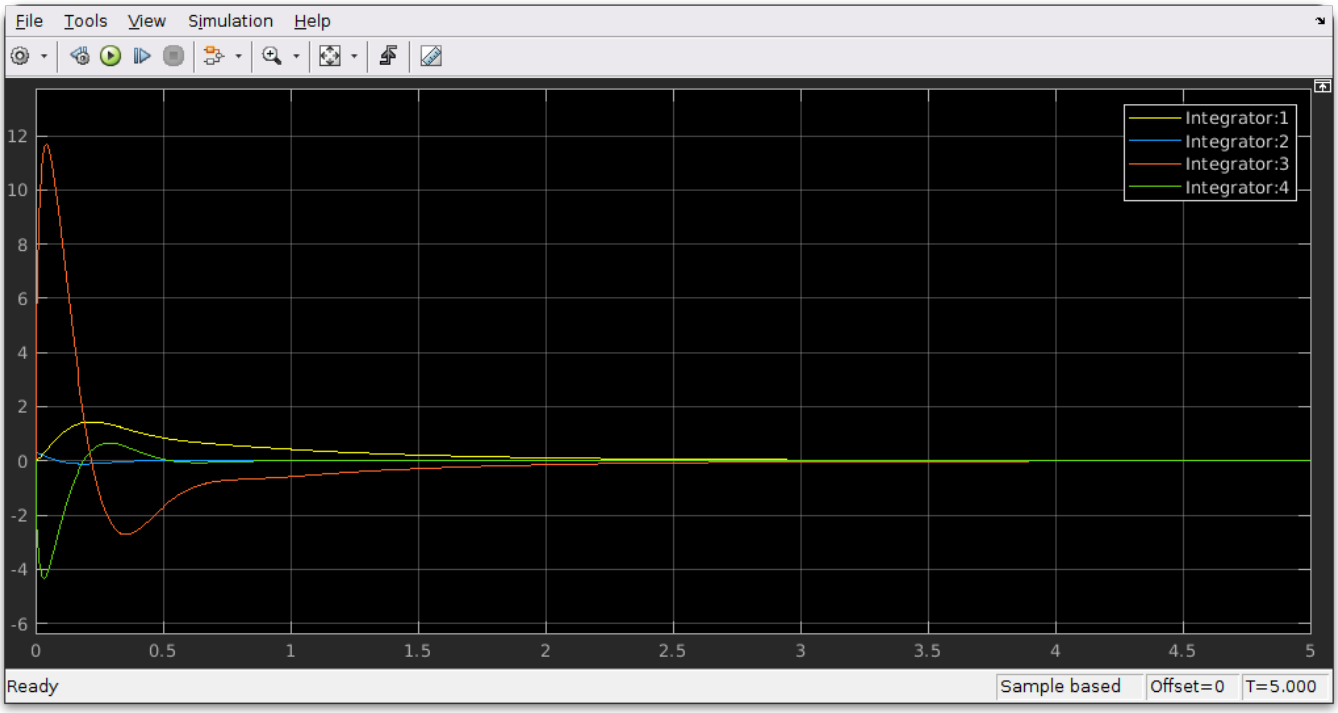


- On modifie le bloc Prédicteur, en utilisant les blocs **Discrete-Time Integrator** et **Discrete-Time Derivative**



Note : il faut bien faire attention à manuellement mettre un pas bas (ici, on utilise $pas = 0.005$), sans quoi les résultats sont aberrants et divergent.

On teste rapidement avec les mêmes paramètres que le cas 2.3 :



On voit bien que l'on obtient les mêmes courbes à vue d'œil.