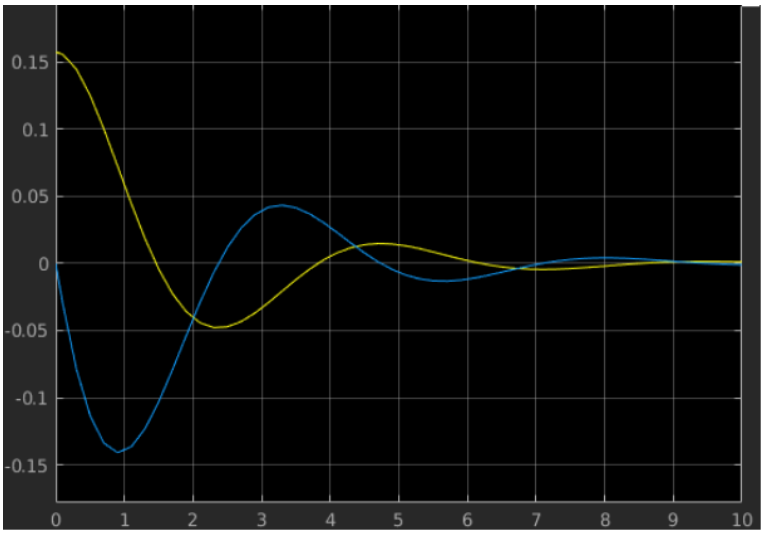
Bontinck Laérian, CM1-F SN 1A Fallot Ysabel, CM1-E SN 1A

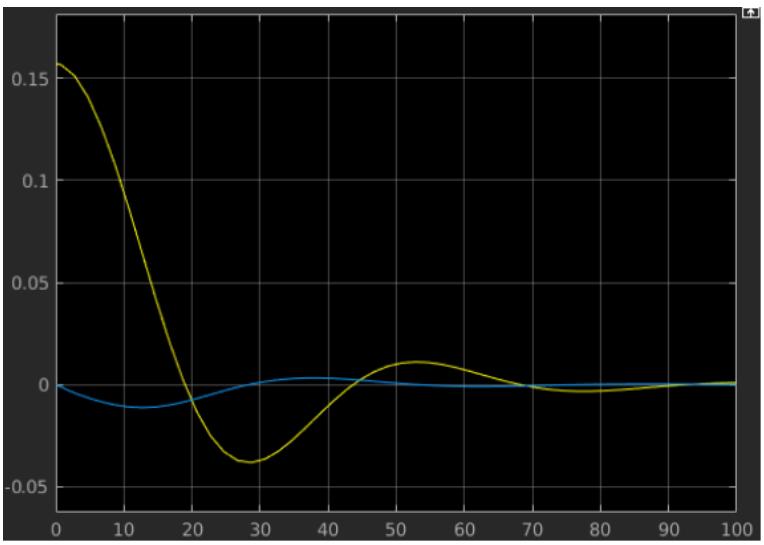
S6 UE AUTOMATIQUE – TP #2

Simulation du pendule inversé contrôlé par retour d'état et de sortie

1 – Contrôle par retour d'état

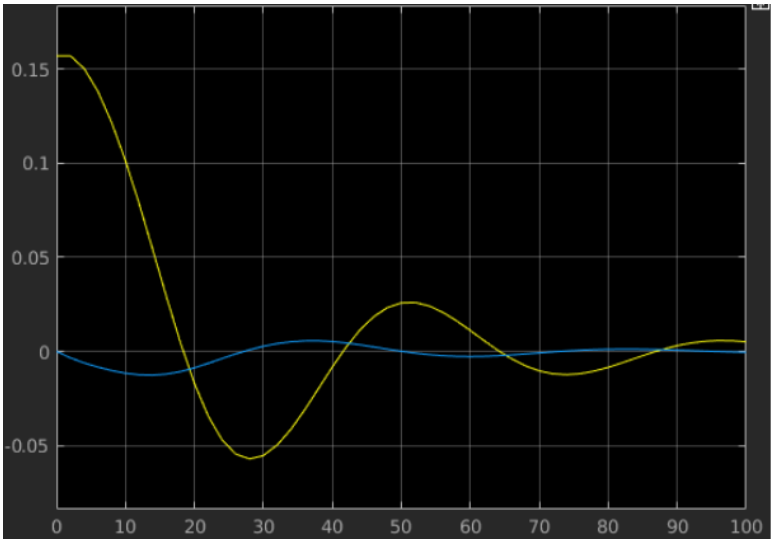


- **Cas 1.1** : cas de référence, pas de commentaire à faire, résultat typique

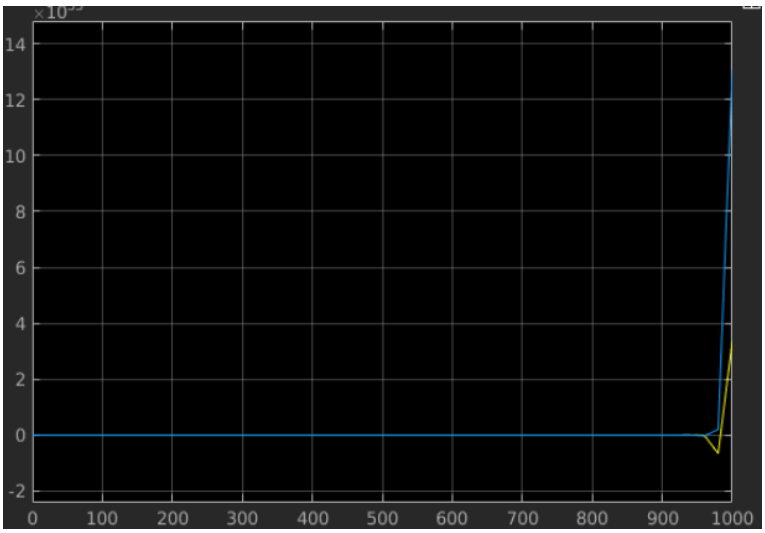


- **Cas 1.2** : on passe le gain de (30,10) à (10,1) et on passe le temps de la simulation de 10 à 100
    - On baisse le gain donc on perd en rapidité, d'où l'intérêt d'augmenter le temps de la simulation

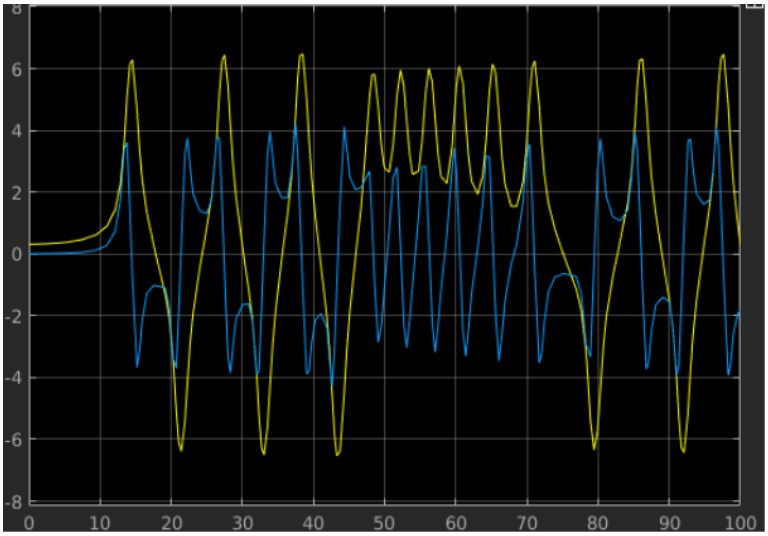
- On a augmenté le ratio du gain en angle sur le gain en longueur, et on observe en conséquence un déplacement réduit du chariot par rapport au cas 1.1



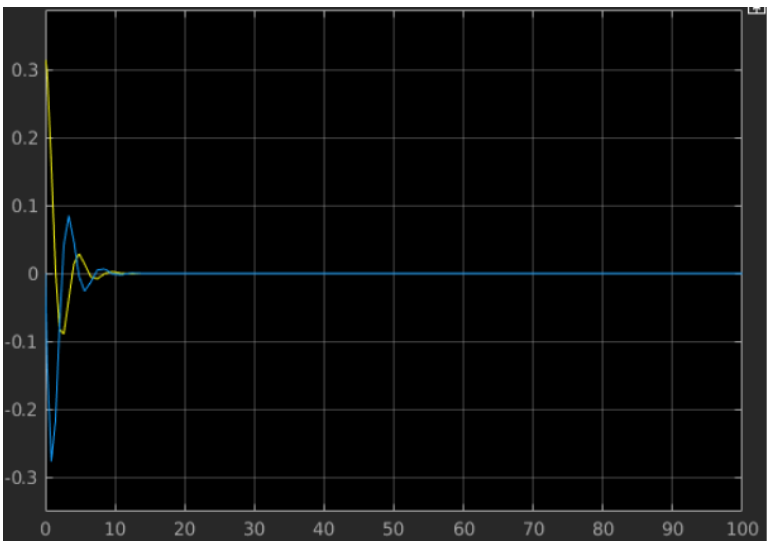
- **Cas 1.3** : on change l'intégrateur de ODE45 à ODE1 (méthode d'Euler),
  - Cet intégrateur est moins précis (généralement la méthode d'Euler n'est que bonne pour résoudre les ODE du premier ordre). Ces imprécisions mènent à une simulation imparfaite du système physique, or notre contrôle va boucler avec ses valeurs. Le système modélisé aura donc un temps de réponse plus long, mais finira tout de même probablement par converger.



- **Cas 1.4** : on passe le temps de simulation de 100 à 1000
  - Avec la méthode ODE1, le pas est constant et par défaut dépend de la durée de simulation. En augmentant cette dernière, le pas devient très grand, ce qui peut mener à des erreurs de calculs. Ce résultat est probablement ce que l'on voit ici.



- **Cas 1.5** : on repasse à 100s de simulation avec l'intégrateur ODE45 On initialise ici le pendule avec un angle de pi/10 au lieu de pi/20
- On observe une claire divergence du système. Le système ne réagit pas assez fort (gain trop faible) aux variations d'états. Ainsi, il n'arrive pas à équilibrer le pendule et diverge.

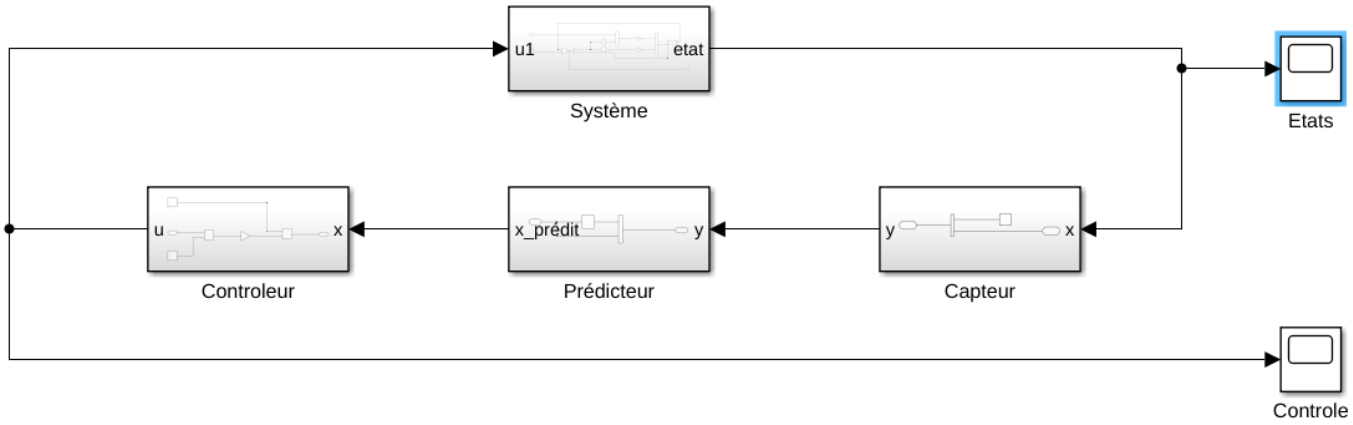


- **Cas 1.6** : on augmente le gain de (10,1) à (30,10)
  - En augmentant le gain et en changeant les ratios, le système réagit mieux aux variations d'état, et arrive à équilibrer le pendule.

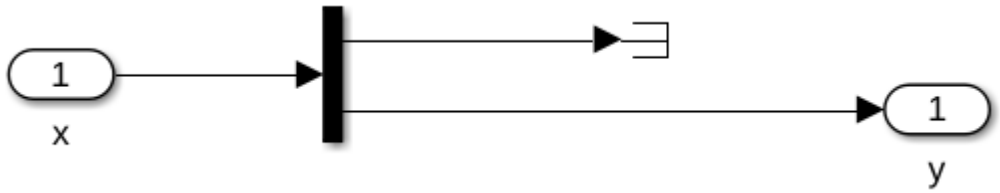
2 - Capteurs

On utilise le schéma Simulink suivant pour schématiser le système de manière plus réaliste : on n'a désormais plus qu'accès aux données que nous donnent les capteurs, et on doit prédire/estimer les autres.

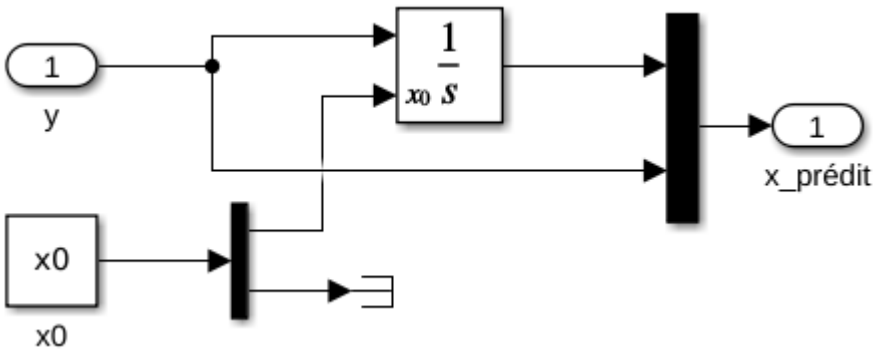
On ajoute alors deux blocs, un bloc Capteur qui réduit les données, et un bloc Prédicteur qui les reconstruit :



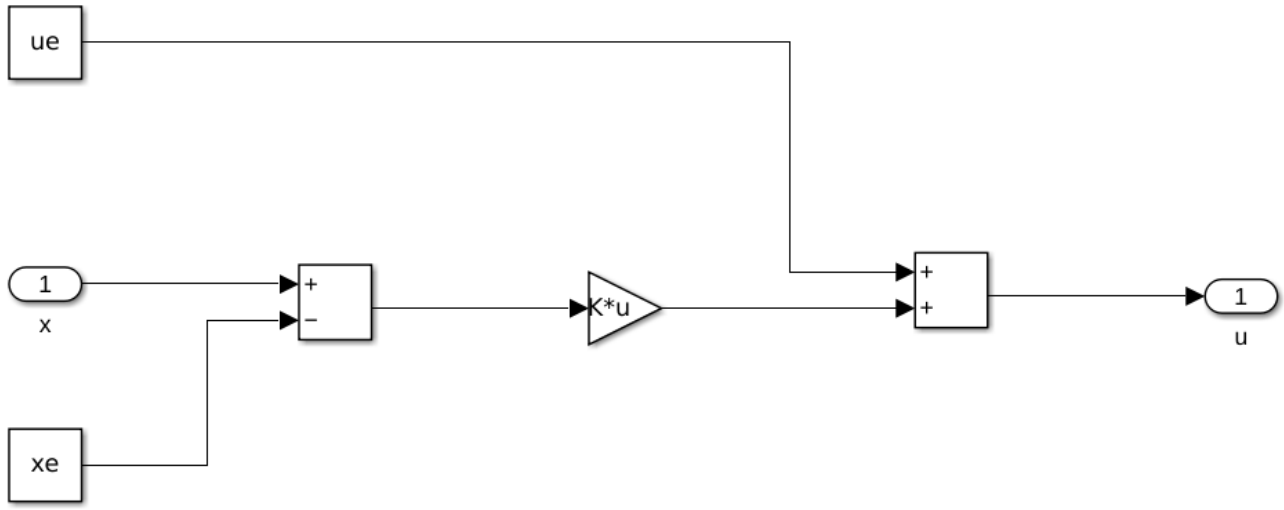
Le sous-système Capteur est ainsi :



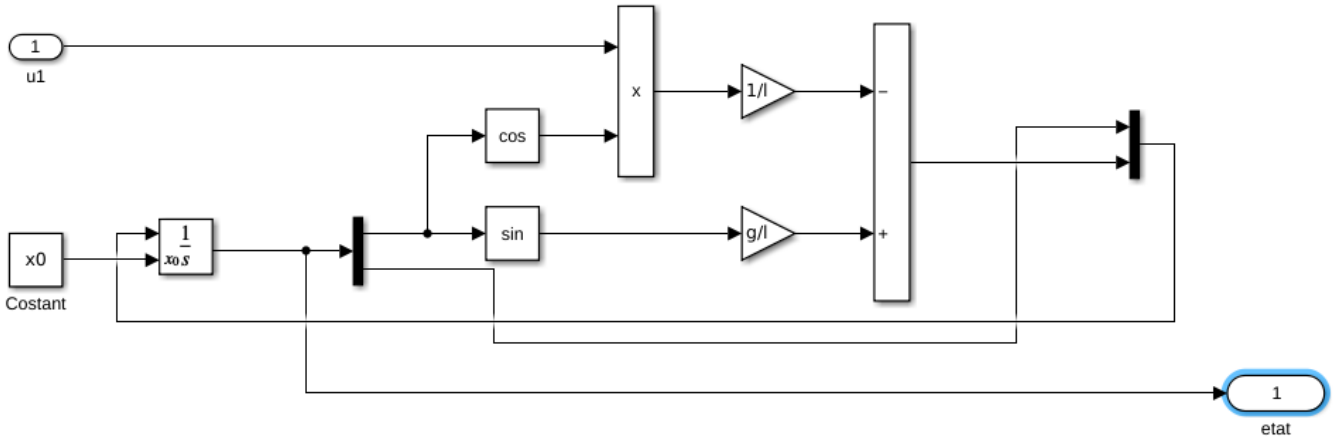
Le sous-système Prédicteur est ainsi :



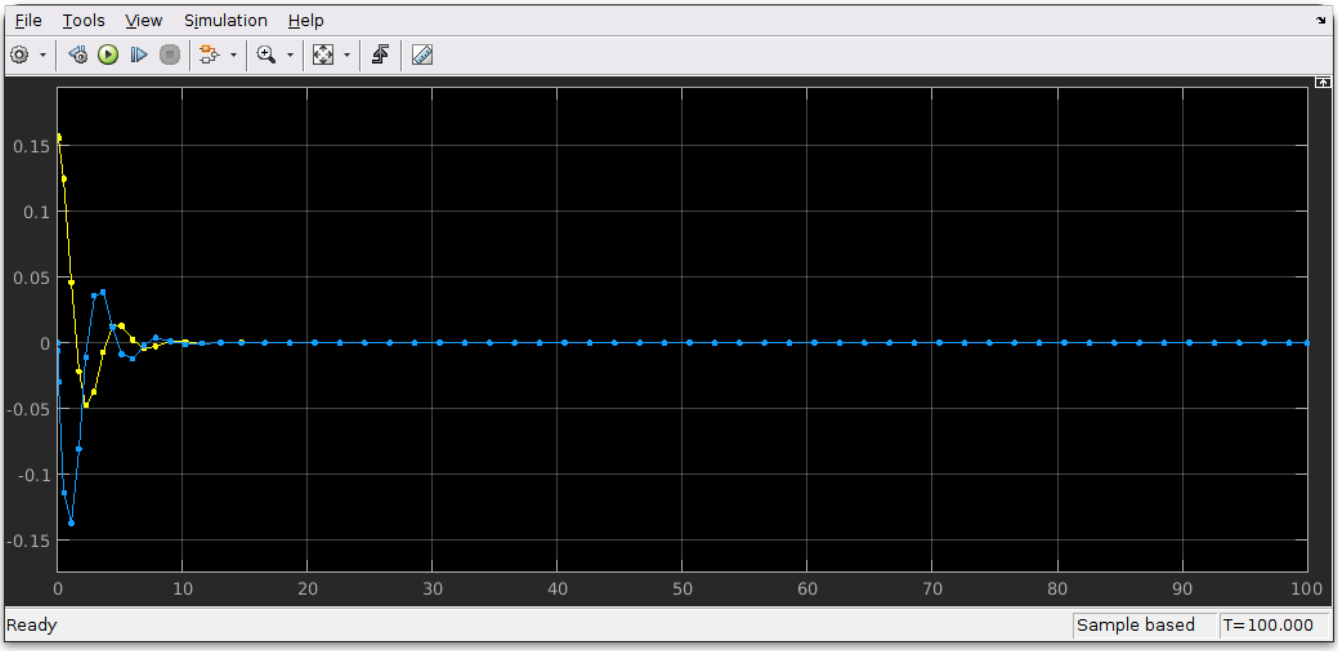
Le module de contrôle reste le même :



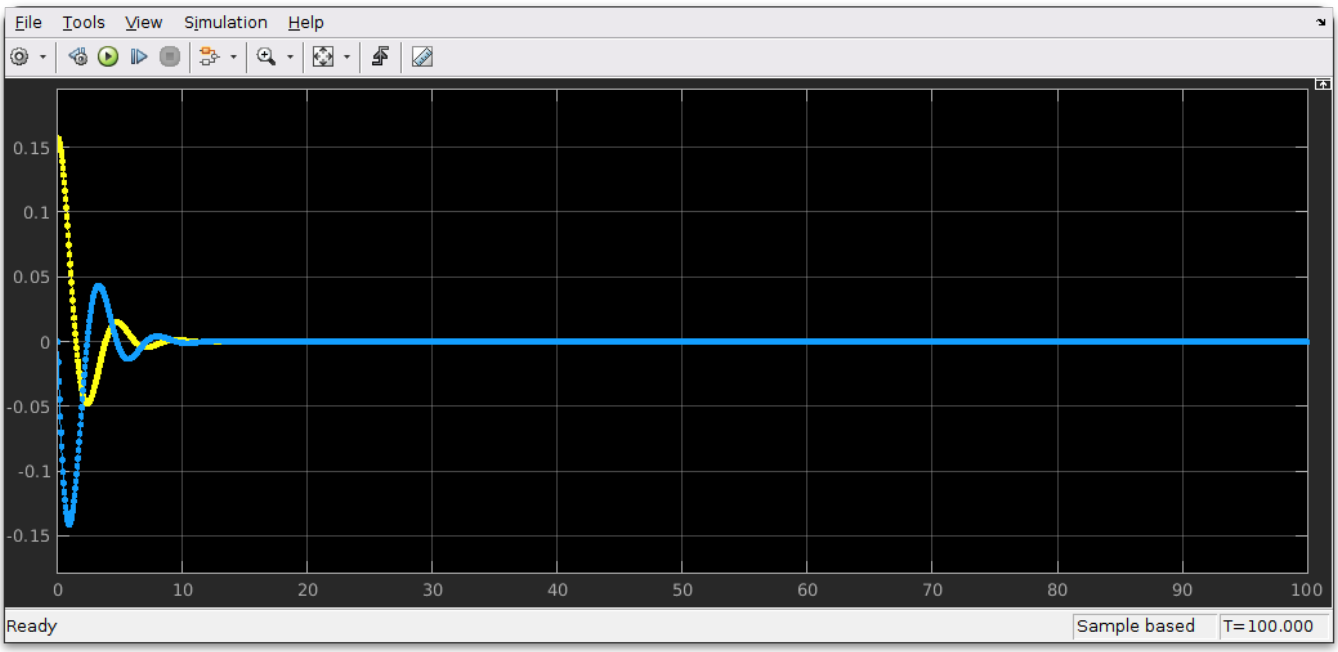
Le module Système aussi :



Étudions maintenant les courbes obtenues après avoir simulé le système dans plusieurs cas.

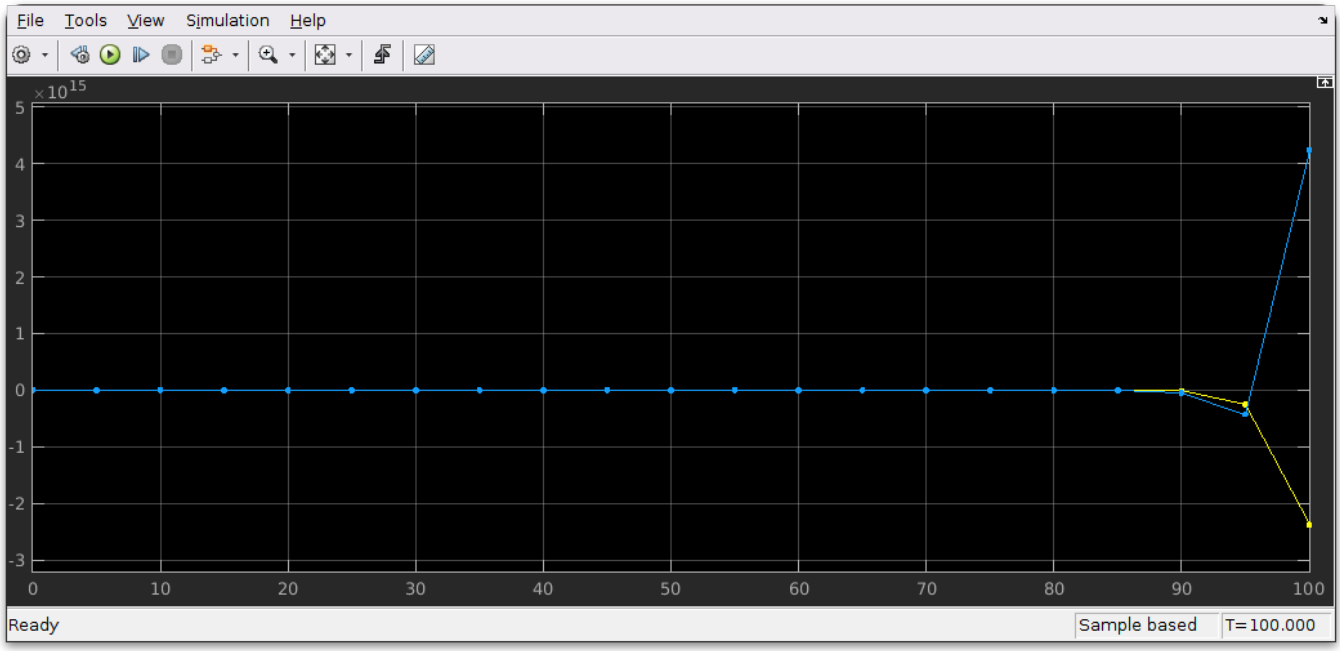


- **Cas 2.1** : cas d'initialisation $x_0 = (\pi/20, 0)$, $tf = 100$, $K = (10, 1)$, intégration par méthode ODE45 (Dormand-Prince) et pas par défaut (variable)
    - On remarque que le système converge relativement normalement.



- **Cas 2.2** : Intégration par méthode ODE1 (Euler), $pas = 0.001$
    - On remarque le même résultat qu'au cas d'avant : c'est normal. Même si la méthode d'Euler a un pas fixe, ce qui a tendance à la rendre moins précise et moins efficace, comme on a fixé le pas très petit, on obtient quand même des résultats proche de ce que la méthode ODE45 (Dormand-Prince) nous donne. Cependant, l'utilisation de la

méthode ODE1 (Euler) est certainement beaucoup moins efficace qu'utiliser la méthode ODE45 (Dormand-Prince).



- **Cas 2.3** : Intégration par méthode ODE1 (Euler), $pas = 5$
    - Contrairement au cas d'avant, on fixe ici un pas très grand, ce qui apporte une grande imprécision sur les calculs, menant à des résultats aberrants. On remarque des résultats similaires à ce que l'on avait obtenu au cas 1.4 : il s'agit du même problème, c'est-à-dire un pas trop élevé.